Western Australian School of Mines
Department of Exploration Geophysics

Reservoir Imaging
Using Induced Microseismicity

Abdullah Ali S Al Ramadhan

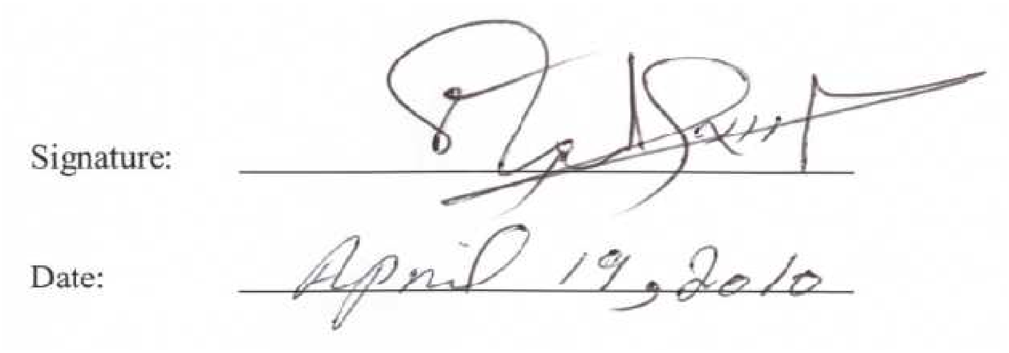
This thesis is presented for the Degree of
Doctor of Philosophy
of
Curtin University of Technology

April 2010

AUTHOR'S DECLARATION

To the best of my knowledge and belief this thesis contains no material previously published by another person except where due acknowledgement has been made.

This thesis contains no material which has been accepted for the award of any other degree or diploma in any university.



Signature:

Date: April 19, 2010

Abstract

Production activities within a hydrocarbon reservoir, such as extracting oil or injecting fluid, result in changes in stress which consequently cause micro-earthquakes. The induced micro-seismic events are small earthquakes producing high frequency waves which can be used to give a hi-resolution image of the hydrocarbon reservoir. However, induced micro-seismic events are usually too small in magnitude to be detected on the surface due to seismic wave attenuation through the overburden rocks. In addition, we lack information about their hypocentres and origin times. Besides, because the ray-path depends on the slowness model, the relationship between the arrival time and the slowness is nonlinear. Therefore, it is important to deploy many sensors well positioned within the hydrocarbon reservoir in order to make use of such induced micro-seismic events for monitoring, characterizing and/or imaging of the hydrocarbon reservoir.

The current practice uses a fixed slowness model to obtain the origin times and hypocentres of induced micro-seismic events within a hydrocarbon reservoir. This, on the one hand, assumes that the velocity model is not changing, which may introduce errors into the hypocentres and origin times. It also ignores the information carried by the waves through the inactive zone. This, on the other hand, cannot replace the conventional 4D seismic time-lapse monitoring method to monitor the dynamic changes within a carbonate hydrocarbon reservoir.

In this thesis, I present an iterative two-stage integrated framework to incorporate arrival times in order to address the problem. First, to estimate the hypocentre and origin time for each micro-seismic event, I have developed and implemented a systematic grid search algorithm to obtain the global minimiser of a nonlinear and multimodal objective function. The algorithm can also be applied to SWD (seismic while drilling) data to locate the drilling bit. Second, to reconstruct an improved velocity model, I have developed and implemented a two-phase algorithm to initially construct an objective function with its gradient for all the micro-seismic events and then apply the variable metric method to optimise the objective function. The algorithm can also be applied to VSP (vertical seismic profiling) data to construct the velocity model. The procedure is iterated until an acceptable match between observed data and computed synthetics is achieved. There are two main reasons for such a choice. First is the fact that both the position coordinates and origin time are unique for each particular micro-seismic event, whereas the slowness is common to all sources. Second is that we start with a good velocity model resulting from the prior information within a hydrocarbon reservoir and this can lead to a very accurate source positions coordinates and origin times. The framework can be used for either P-wave or S-wave.

The methodology could lead to enhanced understanding and hence efficient management of the hydrocarbon reservoir. This in turn would enhance the understanding of fluid movements resulting in improved petroleum recovery from the reservoir.

Acknowledgements

I would like to express my deep love and gratitude to Madinah, my wife, for her loving and loyal support. I also would like to thank my Mum for her continuous praying for me.

I would like to thank Curtin University of Technology. Equally, I would like to express my sincere appreciations to Saudi Arabian Oil Company (Saudi Aramco) for providing me the opportunity and for the moral and financial support they have provided for me and my family members during the course of my study. I also would like to thank the Australian Society of Exploration Geophysicists for providing me the financial support to buy a powerful computer.

I am very grateful to my Thesis Committee and would like to express my profound appreciation and gratitude to the Thesis Advisor, Dr. Bruce Hartley, for his patience, his guidance, his encouragement, and his support and the many thoughtful discussions. Further, I would like to express my deep thanks to Professor Brian Evans, the Thesis Chairperson, for his valuable comments, suggestions, and discussions. Also, I would like to share the same thanks and appreciation to Dr. Milovan Urosevic, Thesis Committee member. I thank Dr. X Wang for his support during the time when he was my Co-supervisor. Further, I would like to thank the post Graduate Coordinator, Professor Boris Gurevich, for his guidance and comments. Special thanks and appreciations are due to Dr. Andrej Bóna for putting the time and effort to review the Thesis and for his helpful comments and considerate discussions. In addition, I would like to thank the faculty and staff of the Department of Exploration Geophysics represented by Dr. Anton Kepic, Department Head, for their endless support and dedication throughout my study at Curtin University of Technology. Special thanks are due to Ms Deirdre Hollingsworth.

I am indebted to many friends and colleagues, past and current PhD students, and many other people who have enriched my experience during my extended stay in the beautiful city of Perth. Specially, I would like to thank my friend, Dr. Dariush Nadri, for the many helpful discussions in optimisation and programming.

Dedication

This Thesis is dedicated to the memory of my father, Ali Salman AlRamadhan (1925-1998).

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Research Motivation

Located in the eastern Saudi Arabia, the Ghawar Field shown in Figure (1.1) is by far the largest known oil field in the world in terms of both production and reserve. It is more than 250 Kilometres long and 30 Kilometres wide and has been producing oil from the Arab-D reservoir since 1951. The Arab-D reservoir has an approximately gross thickness of 100 meters with a recovery factor of 60 % of the original oil in place (Dasgupta, 2005). In other words, approximately 40 % of the original oil would eventually remain within the reservoir. In addition, due to the negligible contrast between the elastic properties of extracted oil and injected fluid, 4D seismic is considered a poor option for managing such a reservoir (Lumley, 2001). Therefore, Saudi Aramco is considering other alternatives to better understand its enormous hydrocarbon reservoirs in order to optimize production and thus enhances hydrocarbon recovery. One technology is being considered is the use of passive seismic technique.

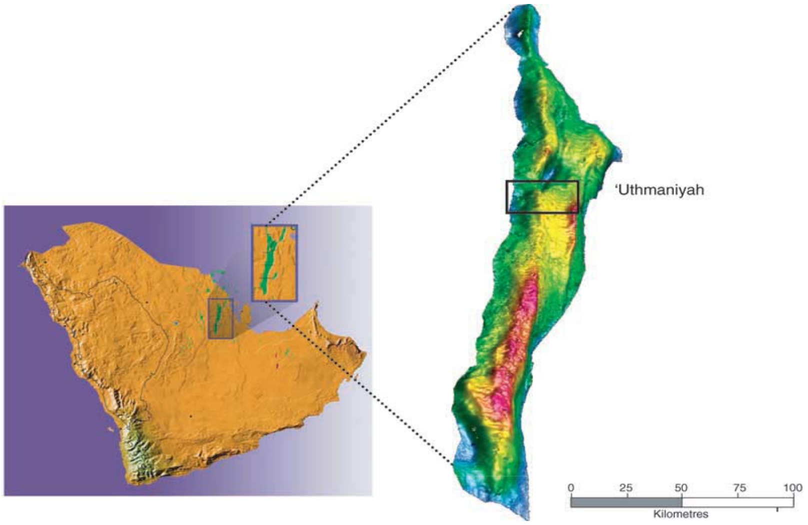


Figure (1.1): The Ghawar Field in Saudi Arabia.

1.2 Background

The use of passive seismic techniques in oil and gas industry has recently attracted an increasing attention as an emerging technology for hydrocarbon reservoir monitoring, characterizing, and/or imaging. For example, the EAGE (European Association of Geoscientists and Engineers) organized a three-day workshop on “Passive Seismic: Exploration and Monitoring Applications”. The event was held in Dubai during the period 10-13 December 2006 and attracted approximately 120 participants from both oil industry and academia.

Active seismic methods – for instance reflection surface seismic – use active energy sources such as vibrators, airguns, or dynamite. Passive seismic techniques, on the other hand, use either naturally produced small-sized earthquakes or induced micro-earthquakes as their energy sources and hence the name passive. Such passive sources lack information about both their positions and the timings. There are two distinct approaches for capturing (monitoring) passive sources.

The first approach is to monitor the seismic events generated as a result of the locally small-sized earthquakes beneath the targeted zone as shown in Figure (1.2). Such triggered events are used for transmission tomography (Duncan, 2005). Local earthquakes produce seismic events propagating within the earth body and passing through the target zone. Generated seismic signals would then be recorded by tri-axial geophones buried just beneath the earth surface to avoid the noisy surface environment, the surface variability and the effects of the strongly attenuating near surface layers. This approach is used to substitute for the 3D surface seismic approach in inaccessible areas such as mountainous terrain and reserved areas, and to reduce cost. The result of such an approach would be a 3D image of the targeted subsurface volume consisting of both compressional and shear velocity structures. However, it would be applicable only for areas where there is enough seismic activity and would require more time to reach a suitable resolution. Furthermore, the observable signals from such passive sources have a limited resolution with a maximum frequency of 60Hz. Therefore, this approach is incapable of generating detailed information on hydrocarbon reservoirs.

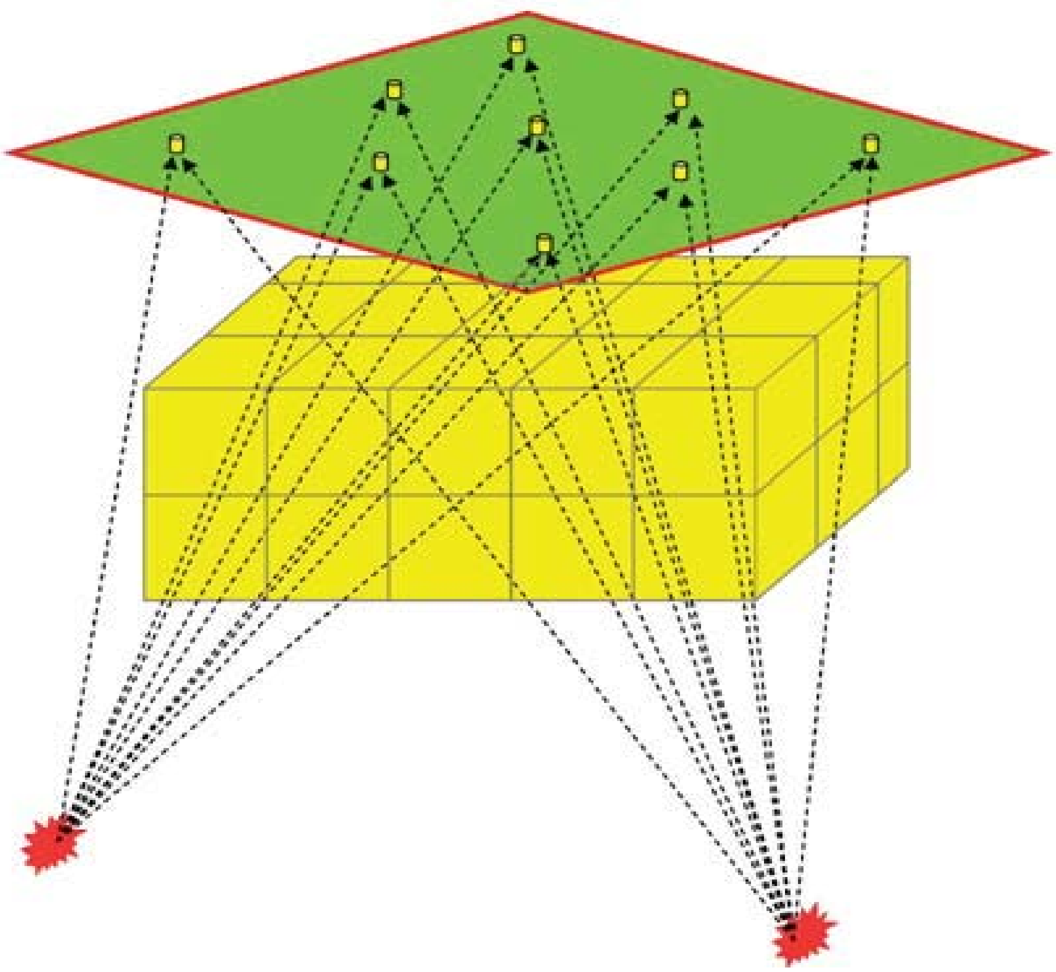


Figure (1.2): A cartoon showing passive seismic transmission tomography (Duncan, 2005).

The second approach is to monitor the micro-seismic events produced as a result of a hydrocarbon reservoir production activities such as hydrocarbon extraction, fluid injection and/or rock fracturing caused by alterations of local stress or pore pressure (Albright et al., 1994; Zang et al., 1998). Figure (1.3) shows how micro-seismic events might start in a hydrocarbon reservoir (Sminchak et al., 2001). Induced micro-seismic events are generally very small in magnitude ranging from -4 to +1 on the Richter scale, a logarithmic measure of energy released by a seismic event. However, the majority of the induced micro-seismic events within a reservoir have magnitude ranging between -4 and -2 (Albright et al., 1994; Phillips et al., 2002). The energy generated as a result of the induced micro-earthquakes propagates as a compressional wave at the P-wave speed followed by a shear wave at the slower S-wave speed. Such waves can be conveniently recorded by tri-axial geophones located close enough to injection wells. The tri-axial sensors are suitably located within monitoring wells to minimise the influence of seismic wave attenuation (Warpinski et al., 1997).

Induced micro-seismicity has been successfully applied in a number of reservoir monitoring operations in order to map the origin of the induced micro-seismic events (Albright and Pearson, 1982; Asanuma et al., 2004; Jones et al., 2004; Phillips et al., 1998; Phillips et al., 2002; Rutledge et al., 1998a). The induced micro-seismic events form cloud-like patterns reduced into simple geometrical shapes such as point clusters, line segments, planar patches or combinations (Phillips et al., 2002). The clustered patterns are thereafter interpreted to characterize the reservoir properties. Some such examples are mapping the hydraulic fracture system within the reservoir (Albright and Pearson, 1982; Block et al., 1994), tracking injected fluid movements within the reservoir (Maxwell et al., 2002; Maxwell et al., 2004), giving a description of the permeability of the reservoir (Rutledge et al., 1998b; Shapiro et al., 1999), or identifying the mechanism of subsidence as hydrocarbon is extracted from a reservoir, and/or replacing 4D seismic method (Dasgupta, 2005).

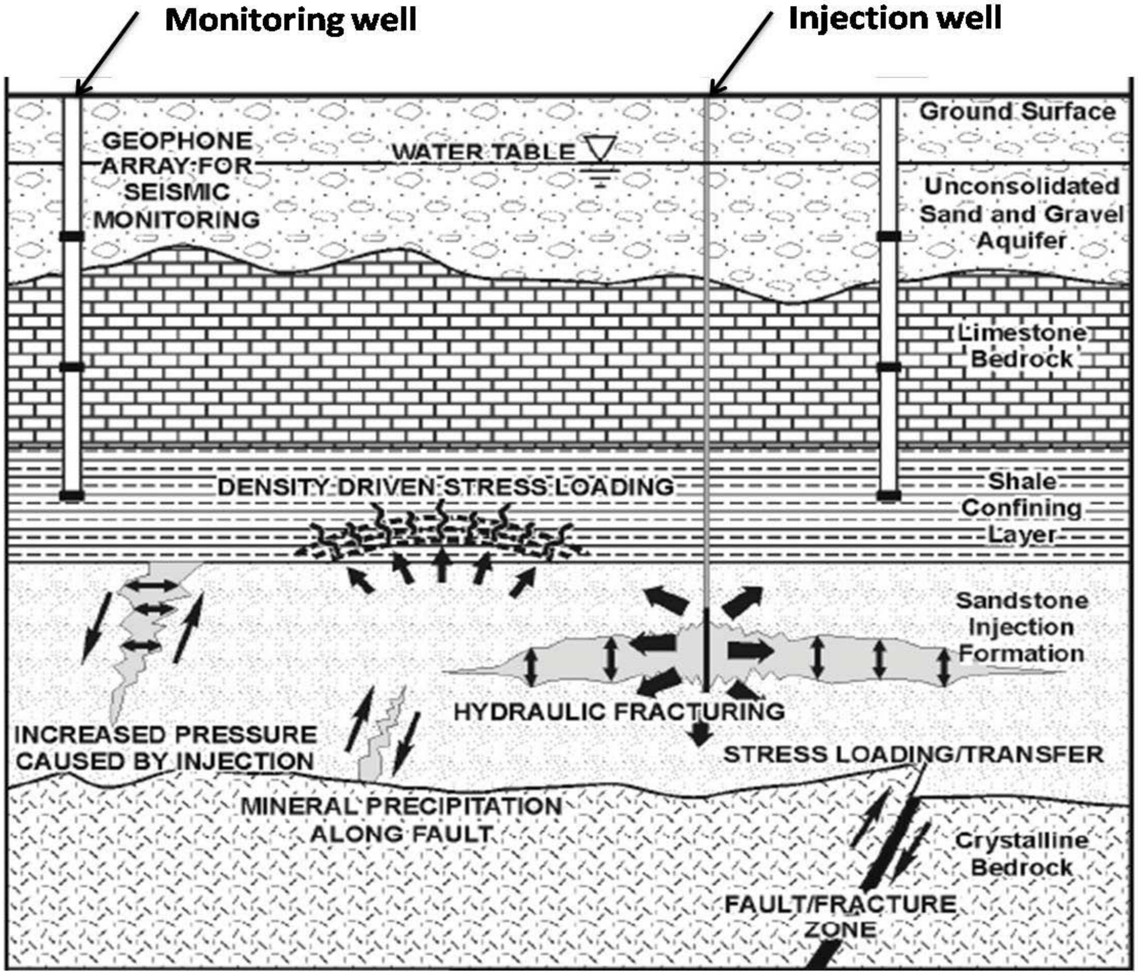

Monitoring well
Injection well
Ground Surface
GEOPHONE ARRAY FOR SEISMIC MONITORING
WATER TABLE
Unconsolidated Sand and Gravel Aquifer
Limestone Bedrock
DENSITY DRIVEN STRESS LOADING
Shale Confining Layer
Sandstone Injection Formation
HYDRAULIC FRACTURING
INCREASED PRESSURE CAUSED BY INJECTION
STRESS LOADING/TRANSFER
MINERAL PRECIPITATION ALONG FAULT
Crystalline Bedrock
FAULT/FRACTURE ZONE

Figure (1.3): Possible mechanisms that generate micro-seismic events as a result of fluid injection (after Sminchak et al., 2001).

The most common mechanism of induced micro-seismicity is tensile failure in hydraulic fracturing. However, the majority of recorded induced micro-earthquakes are thought to be caused by shear slippages (failure) along planes of weakness within the reservoir as a result of alterations in local stress or pore-pressure (Albright et al., 1994). Thus, the presence of micro-seismicity at a given location is taken to indicate a pressure relationship, but not necessarily a high-permeability connection, between that location and the injection well (Phillips et al., 2002). Furthermore, the above described model based on the origin of the sources predicts little shear-slip seismicity along fractures that open in tensile mode, potentially the most conductive flow paths (Phillips et al., 2002). In addition, it does not reveal the complexity of the reservoir in the inactive zones within the reservoir.

1.3 Objective

In this work, I develop a framework to incorporate arrival times of micro-seismic events resulting from the production activities within a hydrocarbon reservoir to determine the hypocentres and origin times of such events and to generate a detailed velocity model. This will lead to an image of heterogeneity in the reservoir.

1.4 Significance

The demand for hydrocarbon is continually increasing and so are the prices. This in turn has lead to directing more resources into new technologies to better manage the existing hydrocarbon reservoirs in order to increase the hydrocarbon recovery, which has been made difficult by the complexity of the hydrocarbon reservoirs. One new technique that has been receiving an increase attention is the induced passive seismic monitoring technique. The reason for this increase is the attractive features that could lead to better understand the complex structure of the hydrocarbon reservoir (such as velocity structure and fracture network), the production properties (such as porosity and permeability) and composition (such as mineralogy and fluid properties). Current practice of hydrocarbon reservoir monitoring and characterization is based on delineating the active zone of the reservoir by locating the positions and origin times of the induced micro-seismic events. These mechanically-based methods are important methods for providing comprehensive information including fault delineation, fluid pressure front movement and flow path anisotropy of the active zone within the hydrocarbon reservoir. However, it is clear that the success of hydrocarbon reservoir monitoring requires a detailed understanding of the hydrocarbon reservoir complexity.

Hydrocarbon reservoir complexity is a function of both anisotropy and heterogeneities in the reservoir properties. Therefore, the degree of anisotropy and heterogeneities is of direct consequence to the hydrocarbon recovery. This spatial variability within the hydrocarbon reservoir cannot be identified with sufficient details by using current practice of 3D surface seismic and well testing data. Therefore, it may be best achieved using recorded induced micro-seismic signals resulting from hydrocarbon reservoir production activity, to thoroughly image such anisotropy and heterogeneities. Unlike the current practice, the technique that I develop is designed for reconstructing the velocity within the hydrocarbon reservoir. The direct transmitted seismic waves resulting from the micro-seismic events are used to derive the velocity model. Therefore, the developed methodology of using micro-seismic events within hydrocarbon reservoirs can give a thorough image of the complexity of the reservoir. A detailed knowledge of the reservoir anisotropy and heterogeneity complexities as well as locations, orientation, spatial density, and connectivity of fractures within the reservoir can lead to better understanding and management of hydrocarbon reservoirs; therefore, greatly enhance the production and petroleum recovery.

1.5 Theory

The determination of a detailed 3D velocity of a hydrocarbon reservoir is a critical problem as the 3D surface seismic method has a limited resolution and thus unable to produce high resolution image. To be able to construct a detailed velocity image of the reservoir, however, one needs to have a hi-frequency source. Induced micro-seismic events produced within the hydrocarbon reservoir as a result of production activities can serve this purpose. Micro-seismic events are characterized by high frequency which can be used to give a hi-resolution image of the reservoir. However, they are too small in magnitude to be detected on the surface due to seismic wave attenuation through the over-burden. Therefore, one must use buried sensors within monitoring wells to be able to use such events for monitoring, characterization and/or imaging of the hydrocarbon reservoir. In this research, I study the use of arrival times of such passive energy sources to locate the micro-seismic events positions and origin time, and to obtain a detailed velocity model to image the heterogeneities within the vicinity of the reservoir.

The arrival time recorded at a particular buried sensor of a passive micro-seismic body-wave resulting from the production activities within a hydrocarbon reservoir is expressed using the ray theory as a path integral,

$$T_{ij} = \tau_i + \int_{s_i}^{r_j} p\left(x, y, z\right) dl. \tag{1.1}$$

T$_{ij}$ represents the arrival time of a micro-seismic event initiated at the source s_i and recorded at the receiver r_j. The term τ_i stands for the time at which the micro-seismic event s_i occurred, p is the slowness function of the medium, and dl is an infinitesimal segment of the ray-path length. The unknowns in equation (1.1) that one needs to determine are the event origin time, the source position coordinates, the velocity field (slowness field), and the ray-path using the recorded traveltimes. Furthermore, the relationship between the recorded arrival traveltime and slowness is nonlinear due to the fact that the path depends on the slowness function.

1.6 Procedure

In geophysics, tomography is used to estimate model parameters, such as velocity distribution, or reflection coefficients, by fitting the forward modelling response to a finite set of observed data. Because the problem in hand is nonlinear, I use an iterative method of linearized inversion using induced micro-seismic arrival times to locate micro-seismic events and to reconstruct the 3D velocity model of the hydrocarbon reservoir. The optimization method is based on the quasi-Newton methods.

The procedure that I am proposing to solve the above problem is a two-stage methodology. First, I solve for the source position coordinates and origin time for each event. Second I solve for the slowness parameters. The procedure is iterated till I achieve an acceptable match between data and synthetics. There are two main reasons for such a choice. First is the fact that both the position coordinates and origin time are unique for each particular source, whereas the slowness field is common to all sources. Second is that one starts with a good slowness model parameters resulting from the prior information within a hydrocarbon reservoir and this can lead to very accurate source positions coordinates and origin times.

1.7 Applications

Saudi Aramco has recently conducted a field pilot study over part of a carbonate oil field (Jervis and Dasgupta, 2006). The study comprises of both downhole and surface 3-components sensors as shown in Figure (1.4) and aims to record microseismic events induced by production and injection activity. The primary objective of the study is to map the flow anisotropy to optimise production. The secondary objective is to use such induced events to monitor the dynamic changes within the carbonate hydrocarbon reservoir as a replacement for the conventional 4D seismic time-lapse monitoring method. The attempt to monitor the dynamic changes in the oil field is an indication that there is a growing interest for the method within the oil industry. Until now the major application of induced passive seismic monitoring technique has been to map the positions of such events based on a fixed velocity model. The new role of the technique is to optimise for both the velocity model and the sources positions in order to account for such changes.

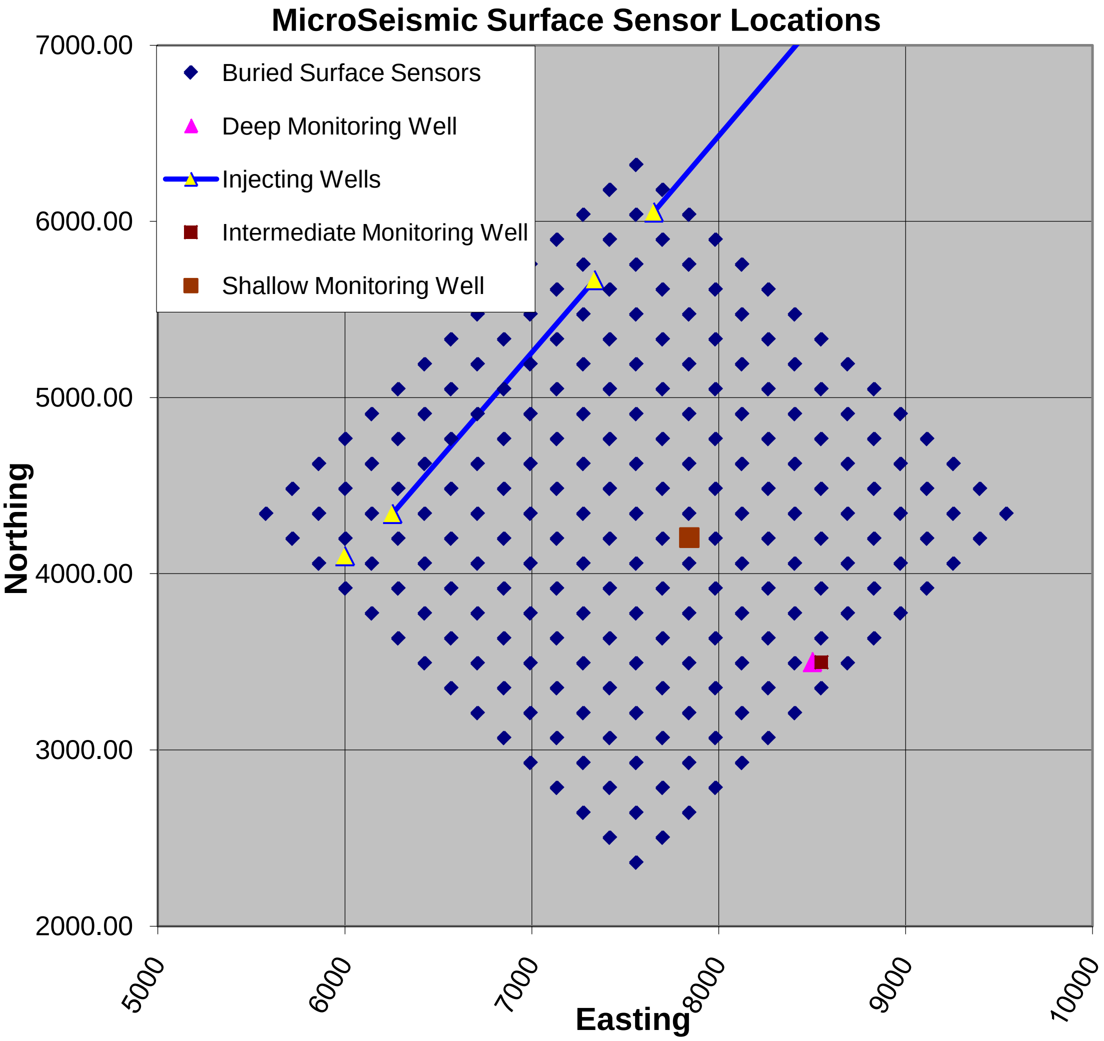


Figure (1.4): Plan view showing the location of the surface sensors network, injecting wells and boreholes in the Saudi Aramco pilot study (after Jervis and Dasgupta, 2006).

1.8 Thesis organization

This thesis is organized into seven chapters. In Chapter 2, I will give a general review of parameter estimation related to micro-seismic events within a 3-dimensional heterogeneous medium. In Chapter 3, I will present the forward modelling of induced micro-seismic events within a heterogeneous 3D medium to generate traveltime function. I will also derive algebraic formulas for tracing a ray-path to compute the ray-path segments through all the traversed cells. In Chapter 4, I will describe the methodology I use to obtain micro-seismic event position and origin time. In Chapter 5, I will detail the procedure I use for reconstructing the slowness model parameters. In Chapter 6, I will present the iterative framework for incorporating all the methods developed in this thesis to determine the events' positions coordinates, the origin times, the ray-path and the slowness model. Chapter 7 is reserved for the conclusions and recommendations.

References

Albright, J.N., B. Cassell, J. Dangerfield, S. Johnstand, and R. Withers, 1994, Seismic Surveillance for Monitoring Reservoir Changes: Oilfield Review, 4-13.

Albright, J.N., and C.F. Pearson, 1982, Acoustic emissions as a tool for hydraulic fracture location: experience at Fenton Hill Hot Dry Rock site: SPE Journal, **22**, no. 4, 523-530.

Asanuma, H., Y. Kumano, T. Izumi, N. Soma, H. Kaieda, K. Tezuka, D. Wyborn, and H. Niitsuma, 2004, Passive seismic monitoring of a stimulation of HDR geothermal reservoir at Cooper Basin, Australia: Applied Geophysics, **159**, 517-541.

Block, L.V., C.H. Cheng, M.C. Fehler, and W.S. Phillips, 1994, Seismic imaging using microearthquakes induced by hydraulic fracturing: Geophysics, **59**, no. 1, 102-112.

Dasgupta, S.N., 2005, When 4D seismic is not applicable: Alternative monitoring scenarios for the Arab-D reservoir in the Ghawar Field: Geophysical Prospecting, **53**, no. 2, 215-227.

Duncan, P.M., 2005, Passive Seismic: what, where and why: Preview, **114**, 26-29.

Jervis, M., and S.N. Dasgupta, 2006, Field monitoring using a large passive seismic array in Saudi Arabia - field pilot study: EAGE Workshop 'Passive Seismic: Exploration and Monitoring Applications', SEG Expanded.

Jones, R.H., D. Raymer, G. Mueller, H. Rynja, K. Maron, and M. Hartung, 2004, Microseismic Monitoring of the Yibal Oilfield, Oman: EAGE 66th Conference & Exhibition, SEG Expanded.

Lumley, D.E., 2001, The next wave in reservoir monitoring The instrumented oil field: The Leading Edge, **20**, no. 6, 640-648.

Maxwell, S.C., T.I. Urbancic, and M. Prince, 2002, Passive Seismic Imaging of Hydraulic Fracture: CSEG 2002, SEG Expanded.

Maxwell, S.C., D.J. White, and H. Fabriol, 2004, Passive Seismic Imaging of CO2 Sequestration at Weyburn: SEG Int'l Exposition and 74th Annual Meeting, SEG Expanded.

Phillips, W.S., T.D. Fairbanks, J.T. Rutledge, and D.W. Anderson, 1998, Induced Microearthquake Patterns and Oil-Producing Fracture Systems in the Austin Chalk: Tectonophysics, **289**, 153-169.

Phillips, W.S., J.T. Rutledge, L.S. House, and M.C. Fehler, 2002, Induced microearthquake patterns in Hydrocarbon and Geothermal Reservoirs: Six case Studies Pure and Applied Geophysics **159**, 345-369.

Rutledge, J.T., W.S. Phillips, L.S. House, and R. Zinno, 1998a, Microseismic Mapping of a Cotton Valley Hydraulic Fracture Using Decimated Downhole Arrays: Presented at The Soc. Explor. Geophys. 68th Ann. Internat. Mtg.

Rutledge, J.T., W.S. Phillips, and B.K. Schuessler, 1998b, Reservoir characterization using oil-production-induced microseismicity, Clinton County, Kentucky: Tectonophysics, **289**, 129-152.

Shapiro, S.A., P. Audigane, and J.J. Royer, 1999, Large-scale in situ permeability tensor of rocks from induced microseismicity: Geophys. J. Int., **137**, 207-213.

Sminchak, J., N. Gupta, C. Byrer, and P. Bergman, 2001, Issues related to seismic activity induced by the injection of CO2 in deep saline aquifers: NETL Proceedings of First National Conference on “Carbon Sequestration,” SEG Expanded.

Warpinski, N.R., J.E. Uhl, and B.P. Engler, 1997, Review of hydraulic fracture mapping using advanced accelerometer-based receiver systems: Natural gas.

Zang, A., F.C. Wagner, S. Stanchits, G. Dresen, R. Andersen, and M.A. Haidekker, 1998, Source analysis of acoustic emission in Aue granite cores under symmetric and asymmetric compressive loads: Geophys. J. Int., **135**, 1113-1130.

Every reasonable effort has been made to acknowledge the owners of copyright material. I would be pleased to hear from any copyright owner who has been omitted or incorrectly acknowledged.

Chapter 2

Parameter Estimation

2.1 Introduction

In this chapter, I give a general review of parameter estimation related to micro-seismic events within a 3-dimensional heterogeneous medium. First, I describe the inverse problem and its operator. Classifications and solutions of the inverse problem will be discussed. Second, I describe the objective function and its derivations for different situations. Third, I give a detailed description of the gradient based optimisations, specifically the quasi-Newton methods because I will use the variable metric method with the BFGS updating formula for constructing the velocity model.

2.2 The Inverse Problem

Inverse problems in geophysics aim to estimate subsurface geophysical parameters, such as velocity or density, and involve three parts. The first is the forward modelling to generate synthetic data for a given set of model parameters. This part is the hardest as it contains all the physics and mathematics governing the response of the system under consideration. Further, it accounts for most of the computations. The second part is the observed data representing a finite set of remotely sensed measurements of the interior of the Earth. Such measurements include error commonly referred to as noise. The third part is the model parameters that need to be estimated. Such parameters represent the unknowns in the inverse problem. Frequently, the geophysical inverse problem is under-determined with fewer independent measurements than unknowns, because geophysical parameters are continuous functions of space coordinates. Thus, we need to constrain the inverse problem in a way to make it well-defined.

2.2.1 Background

Suppose one wants to construct the three-dimensional detailed velocity field of a hydrocarbon reservoir by using arrival times of induced micro-seismic events recorded by sensors placed within monitoring wells. Such arrival times are related to either P or S body waves. I will denote the measured arrival times as $\boldsymbol{d}$ (observed data) while $\boldsymbol{p}$ (unknown parameter) represents the slowness model to be constructed. Assuming the data is error free, one can construct the relationship,

$$\boldsymbol{d} = L(\boldsymbol{p}), \tag{2.2.1}$$

where L represents a mathematical operator, commonly referred to as forward modelling operator. In fact, one can treat an operator as a mapping that takes one function in a topological space (Euclidean Space with Euclidean metric) into another function. Such an operator can be either linear or nonlinear according to the superposition principle. Although the forward modelling operator can have different forms, the two most encountered ones are: differential operator and integral operator. In addition, the operator contains all the laws of physics governing the propagation of seismic waves in an elastic medium.

Equation (2.2.1) represents the forward modelling problem. In chapter 3, I will derive the forward modelling operator for the problem and describe how I use such an operator to generate synthetic data. The inverse of such an equation is written as,

$$\boldsymbol{p}=L^{-1}(\boldsymbol{d}), \tag{2.2.2}$$

Equation (2.2.2) is the direct inversion formula. This formula has limited use in real life because one has a finite set of measurements, contaminated with noise, to reconstruct a model with infinitely many parameters (Snieder, 1998). Instead, the method that is commonly used in geophysical inverse problems is the optimisation technique, known as the model based inversion method.

The problem we are considering is an inverse one which can be described as either linear or nonlinear according to the forward modelling operator. It can also be labelled as a well-posed or ill-posed problem. Well-posedness requires that the problem has a unique solution (existence and uniqueness condition) and that small error in the observed data does not unduly affect the estimated solution (stability condition). One can use a regularisation technique to convert ill-posed problems into well-posed ones or to reduce the condition number in poorly conditioned problems (Berryman, 2000a). When the estimated solution is not sensitive to small numbers of outliers, according to a metric function, the metric function is robust. When the problem is linear, then the forward operator is a matrix. Such a problem would have a unique solution when the inverse of the matrix exists and it is regular; otherwise, it is singular. The stability of such an operator depends on the condition number, the ratio of the largest eigenvalue to the smallest nonzero eigenvalue: the larger the condition number, the less stable is the solution. The matrix becomes singular when the condition number approaches infinity.

The problem we have is characterized by irregular distributions of sources and receivers and so results in an uneven sampling of the medium by induced elastic waves. This condition results in some parts of the medium being sufficiently sampled while other parts remain severely under sampled. Therefore, the problem in hand is ill-posed in the sense that one can generate many solutions agreeing with the observed data (Snieder and Trampert, 1998). Further, the recorded data is inconsistent due to the presence of noise.

2.2.2 Data Space and Model Space

Equation (2.2.1) represents the continuous form which is impossible to obtain as we would require infinitely many measurements. In reality, we deal with discrete data (real numbers), which belong to a data manifold, usually referred to as data space. Such data can be conveniently represented as a column vector,

$$\boldsymbol{d}^o = [d_1^o, d_2^o, d_3^o, ..., d_m^o]^t, \tag{2.2.3}$$

such that t represents the transpose operator and we have a set of m observed data. In other words, the data space is a subset of $\mathbb{R}^m$, the *m*-dimensional Euclidean space, and it is defined over the field of real numbers.

The slowness model is a piecewise continuous function of location and thus has infinitely many model parameters. In chapter 3, I will describe a discretization scheme, by which a medium is expressed as a finite set of model parameters (real numbers) belonging to a model manifold, known as model space. Likewise, such model parameters are expressed as a column vector,

$$\boldsymbol{p} = [p_1, p_2, p_3, ..., p_n]^t. \tag{2.2.4}$$

Each parameter can be thought of as a coordinate of the model space. Similarly, the model space is a subset of $\mathbb{R}^n$, the *n*-dimensional Euclidean space.

The discrete form of equation (2.2.1) is then given as,

$$d_i^c = L_i\,(p_1^0, p_2^0, p_3^0, ..., p_n^0). \tag{2.2.5}$$

The datum d_i^c represents the computed datum such that the initial model parameters $\boldsymbol{p}^0$ are given. Different model discretization leads to different synthetic data. Still, the inaccuracy in synthetic data can be made to be small enough as I will show in chapter 3.

2.2.3 Linear Operator

When the forward modelling operator is a linear operator from the finite-dimensional model space into the finite-dimensional data space over the field of real numbers, one can rewrite equation (2.2.1) as a system of m linear equations,

$$A[p_1, p_2, p_3, ..., p_n]^t = [d_1 + e_1^o, d_2 + e_2^o, d_3 + e_3^o, ..., d_m + e_m^o]^t = \boldsymbol{d} + \boldsymbol{e}, \tag{2.2.6}$$

The linear operator A is then expressed as a matrix with m rows and n columns and referred to mathematically as a linear transformation. It is worth mentioning that the linear operator is not a function, but rather a functional and the individual elements of the operator can be either linear or nonlinear functions. The term e_i^o is the error attached to the d_i^o observed datum. Equation (2.2.6) represents a linear relationship between the observed data (true time with errors) and the model parameters (slowness field). If the synthetic data generated by the forward modelling operator is error free, the calculated data is expressed mathematically as,

$$\boldsymbol{d} = A\boldsymbol{p}. \tag{2.2.7}$$

If the number of model parameters n and the number of observed data points m are equal, the problem is labelled as well-determined and has no solution unless A is regular,

$$\tilde{\boldsymbol{p}} = A^{-1}\boldsymbol{d} + A^{-1}\boldsymbol{e}. \tag{2.2.8}$$

The term $A^{-1}\boldsymbol{e}$ suggests that the error in the data space has propagated to the model space. And the solution $\tilde{\boldsymbol{p}}$ is only an approximation of the true model unless the observed data is error free. The true model here reflects the discretized true model rather than the continuous true model.

If the number of model parameters (unknowns) is more than the number of observed data points (equations), the problem is labelled under-determined and has no unique solution. Consequently, one needs to constrain the problem to compensate for the missing data. One useful solution is obtained by using the minimum norm (Menke, 1984),

$$\tilde{\boldsymbol{p}} = A^t(AA^t)^{-1}\boldsymbol{d} + A^t(AA^t)^{-1}\boldsymbol{e}. \tag{2.2.9}$$

Similarly, the term $A^t(AA^t)^{-1}\boldsymbol{e}$ is the error that propagated to the model space. Using equation (2.2.7), it follows,

$$\tilde{\boldsymbol{p}} = A^t(AA^t)^{-1}A\boldsymbol{p} + A^t(AA^t)^{-1}\boldsymbol{e}. \tag{2.2.10}$$

The matrix $A^t(AA^t)^{-1}A$ is the model resolution operator for the under-determined system. When the resolution operator is close to the identity, the resolved model parameters are closer to the true ones.

On the other hand, if the number of measurements is more than the model parameters and A has a rank of n, the problem is said to be over-determined, but it is still difficult to find a solution that satisfies all the measurements, as observed data are contaminated with noise. In this situation, one can search for a solution that makes the difference between the observed data and the computed data as small as possible. The sum of squared differences is expressed as,

$$\psi(\boldsymbol{p}) = (\boldsymbol{d}^o - A\boldsymbol{p})^t(\boldsymbol{d}^o - A\boldsymbol{p}). \tag{2.2.11}$$

Differentiating equation (2.2.11) with respect to p and setting to zero, one obtains,

$$\tilde{\boldsymbol{p}} = (A^tA)^{-1}A^t\boldsymbol{d} + (A^tA)^{-1}A^t\boldsymbol{e}. \tag{2.2.12}$$

Using equation (2.2.7), equation (2.2.12) reads,

$$\tilde{\boldsymbol{p}} = (A^tA)^{-1}A^tA\boldsymbol{p} + (A^tA)^{-1}A^t\boldsymbol{e}, \tag{2.2.13}$$

where $(A^tA)^{-1}A^t\boldsymbol{e}$ is the error propagated from data space into model space and the expression $(A^tA)^{-1}A^tA = I$ is the model resolution operator for the over-determined system. The over-determined problem has a perfect model resolution; however, the approximated model does not reflect the true model due to the errors in observed data.

To solve a system of linear equations, there are several methods that avoid computing the inverse of a linear operator. One way to solve the linear system is to use a direct method, such as Gaussian elimination or matrix factorization, especially when the system is relatively mall. Another way is to use the singular value decomposition (SVD) method, which is impractical for ill-conditioned matrices, especially when the operator is rank deficient. For more details, one can refers to (Burden and Fares, 1989).

2.2.4 Nonlinear Operator

In geophysical inverse problems, it is very common that the problem is nonlinear and hence the forward modelling operator is nonlinear as expressed in equation (2.2.1). Nonetheless, the forward modelling operator can be linearized in a model $\boldsymbol{p}$ when the nonlinearity is weak and the operator is referenced as a weakly nonlinear operator. In this case, the operator can be expanded as a Taylor series around an initial model $\boldsymbol{p}^0$ which is very close to $\boldsymbol{p}$ as,

$$\boldsymbol{d}^o = L(\boldsymbol{p}) = L(\boldsymbol{p}^0 + (\boldsymbol{p} - \boldsymbol{p}^0)) = L(\boldsymbol{p}^0) + \frac{\partial L(\boldsymbol{p})}{\partial \boldsymbol{p}}|_{\boldsymbol{p}=\boldsymbol{p}^0} (\boldsymbol{p} - \boldsymbol{p}^0) + \dots \ . \qquad (2.2.14)$$

If one retains the first two terms from equation (2.2.14), then one obtains a linearized formula for the weakly nonlinear forward modelling operator,

$$\boldsymbol{d}^o \cong L(\boldsymbol{p}^0) + \frac{\partial L(\boldsymbol{p}^0)}{\partial \boldsymbol{p}}|_{\boldsymbol{p}=\boldsymbol{p}^0} (\boldsymbol{p} - \boldsymbol{p}^0) = \boldsymbol{d}^c + \frac{\partial \boldsymbol{d}^c}{\partial \boldsymbol{p}}|_{\boldsymbol{p}=\boldsymbol{p}^0} (\boldsymbol{p} - \boldsymbol{p}^0) = \boldsymbol{d}^c + G(\boldsymbol{p} - \boldsymbol{p}^0), \qquad (2.2.15)$$

where G is the Jacobian matrix with m rows and n columns containing the partial derivatives of the synthetic data $\boldsymbol{d}^c$ generated from the initial model $\boldsymbol{p}^0$ with respect to the model $\boldsymbol{p}$ and $\boldsymbol{d}^o$ is the observed data. Equation (2.2.15) can be expressed as,

$$\boldsymbol{d}^o - \boldsymbol{d}^c = G(\boldsymbol{p} - \boldsymbol{p}^0). \qquad (2.2.16)$$

Equation (2.2.16) represents a linear relationship between the change in data and change in model parameters such that the matrix G acts as a forward modelling operator. That is, the generation of the synthetic data for a given model is accomplished through a linear perturbation of the computed data for a reference model when the forward modelling operator is weakly nonlinear. For the case when the operator is linear or weakly nonlinear, one can obtain an estimate of the solution in a single step.

When the operator cannot be linearized, then it is referred to as either quasi-linear or strongly nonlinear. The quasi-linear operator is important as the problem we are facing is not a strongly nonlinear one. A quasi-linear operator is approached by linearizing the synthetic data rather than the operator. In other words, one needs to iteratively compute the quasi-linear operator as a nonlinear operator; subsequently, generate a linearized function for the computed data in terms of model parameters. Mathematically one has,

$$L(\boldsymbol{p}^j) = \boldsymbol{t}_j^c = \sum_{i=1}^{N_p} a_i^j p_i, \tag{2.2.17}$$

where j is the iteration number, N_p is the number of model parameters, and a_i^j is a parameter (which could be constant, linear, quadratic or another function) for the j^th iteration and the i^th parameter. The quasi-linear problem is best approached by optimisation, deploying the gradient and/or the curvature. I will detail this approach in section 2.4.

When the operator is strongly nonlinear, the optimisation approach can be applied, but with many different starting points. Another approach to overcome strongly nonlinear operator is to use global optimisations which demands intensive computations (Sen and Stoffa, 1995).

2.3 Objective Function

It is unlikely that a solution of equations (2.2.6) agrees with all measurements because the observed data contain errors. Also, the matrix A is generally not square and either rank deficient or ill-conditioned or both. In addition, computing the pseudo inverse of a matrix is not efficient, particularly when the matrix is large. Direct methods and SVD method are not feasible for large system, especially when dealing with poorly conditioned matrices. Therefore, the natural approach to solve such a problem is to iteratively optimise a misfit between the synthetic data and the measured data (Berryman, 1990; Berryman, 2000b; Cardiff and Kitanidis, 2008; Lines and Treitel, 1984; Scales and Snieder, 1997; Sen and Stoffa, 1995; Snieder, 1990; Snieder, 1998; Snieder and Trampert, 1998; Tarantola, 2005). In addition, when the operator is nonlinear, it is necessary to use an iterative approach. Generally, an iterative algorithm starts at an initial solution and produces a sequence converging to a local solution that minimises the error function. The misfit function is commonly recognized as the objective function, or the cost function. A nonlinear objective function may have multiple valleys but the shape is nearly quadratic when it is well approximated by a linear operator. A nonlinear objective function is shown in Figure (2.1). Depending on our initial model, we may not be able to locate the desired solution (global minima).

For an optimisation problem, one needs to evaluate all factors and their relationships, requirements, and objective of the problem. This leads to the construction of a mathematical model. Through the construction process, one may need to make some simplifications as long as the mathematical model produces realistic answers that serve the purpose for which they are designed. Such simplifications can be in the form of assumptions, approximations, and/or estimations. Assumptions are made to define the limitations of the mathematical model and lead to structural simplifications. The purpose of such assumptions is to have a simple mathematical model yet retain adequate realism of the actual model. Approximations are mathematical tools used to calculate the value of a complex function by a rather simple one without sacrificing the accuracy of the mathematical model. Estimations, on the other hand, are statistical methods used to assign uncertainties to model parameters when such parameters are random variables or depend on some stochastic process.

In the following, I describe three different settings to build the objective function. Such function can be simple or rather more complex depending on the nature of the problem and the available prior information.

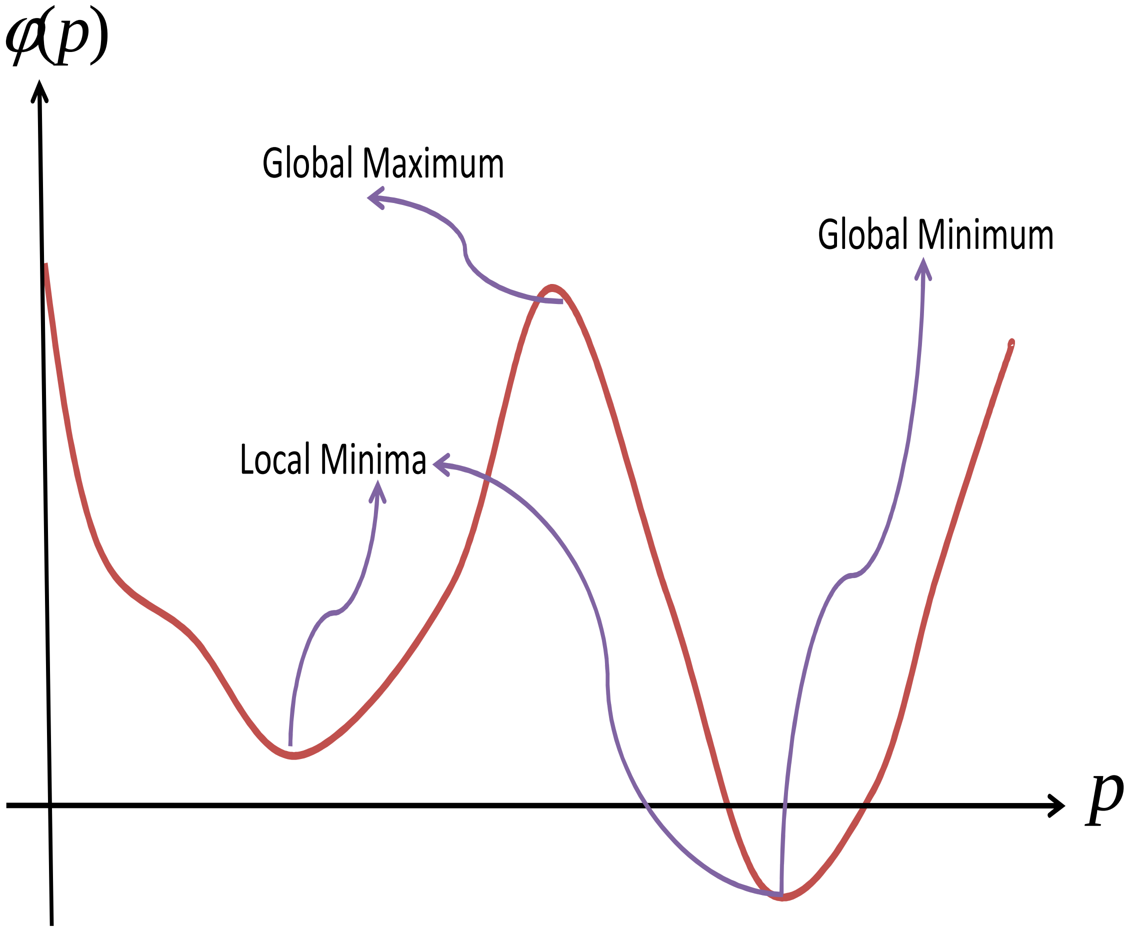


Figure (2.1): Diagram of a possible objective function for a nonlinear problem.

2.3.1 Algebraic Derivation

Let $\mathbb{R}^n$ represent metric space such that $n \in \mathbb{N}$, then,

$$\mathbb{R}^n = \{\boldsymbol{x} = \{x_1, x_2, x_3, ..., x_n\} : x_i \in \mathbb{R} \text{ for each } i = 1, 2, 3, ..., n\}, \tag{2.3.1}$$

where $\mathbb{R}$ is the set of real numbers and $\mathbb{N}$ is the set of natural numbers. Further, define the function $\varphi : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$ as,

$$\varphi(\boldsymbol{x}, \boldsymbol{y}) = \left(\frac{1}{r}\sum_{i=1}^{n} \left|x_i - y_i\right|^r\right)^{\frac{1}{r}}. \tag{2.3.2}$$

The function φ represents a metric function for $\mathbb{R}^n$ space when $r \geq 1$ and is used to measure length or size. The differences $(x_i - y_i)$ are weighted according to *r* value: the larger the value of *r*, the larger weight is given to the larger differences (errors). When *r* is *1*, the errors of different sizes are treated equally.

If *r* equals *2*, we have the Euclidean metric, which is commonly known as the l_2 norm. Unlike the l_1 norm, the Euclidean metric is a smooth functional and thus more efficient for optimisation. It follows that one can use the Euclidean metric, equation (2.2.3) and equation (2.2.5) to construct a misfit function between observed and computed data as,

$$\varphi^2 = \frac{1}{2}\sum_{i=1}^{m} (d_i^o - d_i^c)^2. \tag{2.3.3}$$

Equation (2.3.3) can be expressed in vector form as,

$$\varphi^2 = \frac{1}{2}(\boldsymbol{d}^o - L(\boldsymbol{p}))^t(\boldsymbol{d}^o - L(\boldsymbol{p})). \tag{2.3.4}$$

Then the objective function one aims to minimise for an operator,

$$2\theta(\boldsymbol{p}) = (\boldsymbol{d}^o - L(\boldsymbol{p}))^t(\boldsymbol{d}^o - L(\boldsymbol{p})) \tag{2.3.5}$$

Equation (2.3.5) is the nonlinear least-squares misfit function. When the nonlinear operator is replaced with a linear one, one obtains an objective function for the linear operator. Such a function attains its minimum when the synthetic data is the arithmetic mean of the observed data. This function implies that the observed data has a Gaussian distribution (Tarantola, 2005). When *a priori* information is available, one still can further constrain the problem. *A priori* information is independent of the observed data and is expressed in many forms. For example, slowness parameters are constrained to have positive values within certain ranges.

If one has a reference model, the objective function is expressed as,

$$2\theta(\boldsymbol{p}) = (\boldsymbol{d}^o - A\boldsymbol{p})^t(\boldsymbol{d}^o - A\boldsymbol{p}) + \alpha\left\|\boldsymbol{p}\right\|^2, \tag{2.3.6}$$

where α is a positive weighting constant used to quantify how reliable the reference model is. When the value of α is large, more weight is given to the prior information and vice versa. Further, different model parameters may have different weights.

2.3.2 Statistical Derivation

Recorded data is contaminated with unknown errors and hence a statistical analysis is needed to quantify errors in the estimated model parameters. Each observed datum is consider as a random variable as indicated by equation (2.2.6) with a known mean and variance. Assume that such a random variable has a Gaussian probability density distribution. The joint probability density function in vector form is given as,

$$\rho(\boldsymbol{d}^o) = c\exp(-\frac{1}{2}(\boldsymbol{d}^o - L(\boldsymbol{p}))^t C_o^{-1}(\boldsymbol{d}^o - L(\boldsymbol{p}))). \tag{2.3.7}$$

The matrix C_o represents the covariance matrix for the observed data and $L(\boldsymbol{p})$ is the synthetic data representing the mean data. When the observed data are uncorrelated, the elements of the covariance matrix are zero except those along the main diagonal and they correspond to the variances. Further, if the problem is linear, equation (2.3.7) which is a maximum likelihood function is given as,

$$\rho(\boldsymbol{d}^o) = c\exp(-\frac{1}{2}(\boldsymbol{d}^o - A\boldsymbol{p})^t C_o^{-1}(\boldsymbol{d}^o - A\boldsymbol{p})). \tag{2.3.8}$$

So, instead of using the maximum likelihood method to optimise the model parameters by maximizing the probability of the observed data such that the uncertainties in the data are given, one can minimize,

$$2\theta(\boldsymbol{p}) = (\boldsymbol{d}^o - L\boldsymbol{p}\ )^t C_o^{-1}(\boldsymbol{d}^o - L\boldsymbol{p}\ ). \qquad (2.3.9)$$

It follows that equation (2.3.5) and (2.3.9) are the same when C_o is reduced to the identity matrix. For uncorrelated measured data, the residuals are inversely weighted according to their corresponding variances. When synthetic data are not exact, one can still use the Gaussian distribution to describe the uncertainties (Tarantola, 2005),

$$2\theta(\boldsymbol{p}) = (\boldsymbol{d}^o - A\boldsymbol{p}\ )^t C_d^{-1}(\boldsymbol{d}^o - A\boldsymbol{p}\ ). \qquad (2.3.10)$$

The covariance matrix C_d is the sum of the two covariance matrices for synthetic and observed data. When variances are not available, one can still use weighting matrix and the objective function is defined as,

$$2\theta(\boldsymbol{p}) = (\boldsymbol{d}^o - A\boldsymbol{p}\ )^t W_d(\boldsymbol{d}^o - A\boldsymbol{p}\ ). \qquad (2.3.11)$$

Equation (2.3.11) is commonly known as the weighted least-squares method. One can arrive at the same result by inserting weights in equation (2.3.5). Such a weighting matrix is needed to suppress the contribution of the outliers. Also, one may want to apply smaller weights to less reliable data and larger weights to more reliable data. When the reference model has a Gaussian distribution such that $\boldsymbol{p}_{prior}$ is the mean and C_{prior} is the covariance matrix, one can use the maximum likelihood method to obtain the objective function for such situation (Sen and Stoffa, 1995),

$$2\theta(\boldsymbol{p}) = (\boldsymbol{d}^o - A\boldsymbol{p}\ )^t C_d^{-1}(\boldsymbol{d}^o - A\boldsymbol{p}\ ) + (\boldsymbol{p} - \boldsymbol{p}_{prior})^t C_{prior}^{-1}(\boldsymbol{p} - \boldsymbol{p}_{prior}). \qquad (2.3.12)$$

2.3.3 Bayesian Derivation

The idea of this section is to use the information that one knows before making a measurement with what is known afterwards, by deploying Bayes theorem e.g. (Scales and Snieder, 1997; Sivia, 1998). Let us assume that each observed datum d_i^o and model parameter p_j are random variables with Gaussian distributions. Let $\rho_d(\boldsymbol{d}^o \mid \boldsymbol{p}\ )$ be the conditional probability distribution function for $\boldsymbol{d}^o$ such that $\boldsymbol{p}$ is given and $\rho_m(\boldsymbol{p}\ \mid \boldsymbol{d}^o)$ be the conditional probability distribution function for $\boldsymbol{p}$ such that $\boldsymbol{d}^o$ is given. Then, Bayes theorem states,

$$\rho_m(\boldsymbol{p} \mid \boldsymbol{d}^o) = \frac{\rho_d(\boldsymbol{d}^o \mid \boldsymbol{p}\ )\rho(\boldsymbol{p})}{\rho(\boldsymbol{d}^o)}. \qquad (2.3.13)$$

The probability distribution $\rho(\boldsymbol{p})$ is independent of the observed data and hence it represents the probability distribution of *a priori* model. The probability distribution $\rho(\boldsymbol{d}^o)$ is the distribution of the observed data and it is normally distributed with known mean and variance and it is independent of the model parameter $\boldsymbol{p}$. Further, $\rho_d(\boldsymbol{d}^o \mid \boldsymbol{p}\ )$ is the likelihood function and $\rho(\boldsymbol{d}^o)$ is a normalizing constant (Aster et al., 2005). The distribution $\rho_m(\boldsymbol{p}\ \mid \boldsymbol{d}^o)$ is called the posterior probability density function which describes the model parameters. Substituting the corresponding distributions in equation (2.3.13), one obtains the posterior probability density in the model space (Tarantola, 2005),

$$\rho_m(\boldsymbol{p} \mid \boldsymbol{d}^o) = c.\exp(\frac{1}{2}[(\boldsymbol{d}^o - L(\boldsymbol{p}))^t C_d^{-1}(\boldsymbol{d}^o - L(\boldsymbol{p})) - (\boldsymbol{p} - \boldsymbol{p}_{prior})^t C_{prior}^{-1}(\boldsymbol{p} - \boldsymbol{p}_{prior})]). \qquad (2.3.14)$$

Therefore, the posterior conditional distribution partially depends on the prior distribution of the true model. Further, equation (2.3.14) holds for any number of measurements (Scales et al., 2001). Instead of maximising the posterior probability density function, one needs to minimise the objective function in equation (2.3.12) which is equivalent to the exponent parameter. When the forward modelling operator is linear, then the posterior distribution is Gaussian and expressed as (Tarantola, 2005),

$$\rho(\boldsymbol{p}\ \mid \boldsymbol{d}^o) = c.\exp(-\frac{1}{2}(\boldsymbol{p} - \boldsymbol{p}_{posterior})^t C_{posterior}^{-1}(\boldsymbol{p} - \boldsymbol{p}_{posterior})), \qquad (2.3.15)$$

where $\boldsymbol{p}_{posterior}$ is the estimated model mean and $C_{posterior}$ is the covariance matrix for the probability distribution function and is given as,

$$C_{posterior} = [A^t C_d^{-1} A + C_m^{-1}]^{-1}. \tag{2.3.16}$$

Equation (2.3.16) suggests that even nonlinear problems require locating the point that maximises the posterior probability distribution function or equivalently, minimising the objective function; consequently, equation (2.3.16) is still true even when the problem is weakly nonlinear or quasi-linear (Scales et al., 2001; Sen and Stoffa, 1995; Tarantola, 2005).

2.4 Optimisation

A general form of an optimisation problem can be given as,

$$\begin{aligned} &\text{minimise} && f(x) \\ &\text{subject to} && g_i(x) = 0, i = 1, 2, \ldots, m, \\ &&& h_i(x) \geq 0, i = m+1, \ldots, M. \end{aligned} \tag{2.4.1}$$

The function $f(x)$ is the objective function and $g_i(x)$ & $h_i(x)$ are the equality and inequality constraints, respectively. The feasible region is the set which consists of all x such that the constraints are satisfied. The objective function can be: a function of single variable or several variables; a linear or a nonlinear function; a smooth or a non-smooth function. On the other hand, the constraints can be in the form of bonds, linear and/or nonlinear functions, smooth and/or non-smooth function. When there are no constraints, the problem is referred to as unconstrained optimisation problem; otherwise, it is known as constrained optimisation problem. Here I treat the unconstrained optimisation problem because the constraints are explicitly included within the objective function. Initially, I consider the single variable case as it is crucial in the problem of several variables for obtaining the required step length to proceed from one iteration to the next iteration. In section 2.4.2, I will generalise the treatment for the problem of several variables.

2.4.1 One Dimensional Optimisation Problem

Consider the unconstrained minimisation problem,

$$\text{minimise} \quad f(x), \tag{2.4.2}$$

such that $f(x)$ is a smooth function. It has a local minimiser x^* when $f(x^*) \leq f(x)$ over some open interval including x^* and referred to as a global minimiser when the condition is true for any open interval. In numerical optimisation, it is very hard to obtain the global minimiser for a nonlinear optimisation problem. Thus the goal is to locate a local optimiser rather than the global one. In attempting to optimise a function, an iterative algorithm is used and one needs to have a starting point. When the algorithm converges to some local minimiser regardless of the initial solution, it has a global convergence; otherwise, it has a local convergence.

Depending on how much is known about the function to be optimised, one can decide on the method to use. Indeed there are many line search methods, as described hereafter, to pick from (Burden and Fares, 1989; Nocedal and Wright, 1999; Rao, 1996; Ravindran et al., 2006), but I will focus on three categories. Such categories assume that the function is unimodal. That is, it has a single local minimum on a locally closed interval. The first category is referred to as local search methods including the Fibonacci search method and the golden section search method. Such methods do not locate the optimiser, but rather generate an interval of uncertainty as small as required $(O(\sqrt{\varepsilon})$ where ε is the accuracy of the computer) (Rao, 1996). Both methods require only one function evaluation per step. The ratio of interval reduction at each step is variable for the Fibonacci, but rather constant for the golden section method. The two methods are efficient. Fibonacci method has a faster rate of reducing the interval, but it requires the number of function evaluations be known in advance. Therefore, it is more convenient to use the golden section method and it requires less computation at each individual iteration. Generally, the two methods require a simple comparison of function values at two trial points and then to decide how to proceed. For further reading one can refer to

(Rao, 1996; Ravindran et al., 2006). The algorithm for the golden section search method is summarised in the following paragraph.

The golden section method starts with three points $x_l < x < x_r$ such that,

$$f(x) \leq f(x_l) \text{ and } f(x) \leq f(x_r). \tag{2.4.3}$$

This is the bracketing condition because the function is assumed unimodal on $[x_l, x_r]$. The number $\alpha = \frac{\sqrt{5}-1}{2}$ is the constant reduction factor of the uncertainty interval per iteration. Two points are generated at the first iteration and the function is evaluated at such points,

$$\begin{aligned} x_1 &= x_l + \alpha(x_r - x_l) \\ x_2 &= x_r - \alpha(x_r - x_l). \end{aligned} \tag{2.4.4}$$

Compare $f(x_1)$to $f(x_2)$and decide how to reduce the interval. If $f(x_1) \leq f(x_2)$, then the new four points are given sequentially as,

$$x_l = x_l, x_r = x_2, x_2 = x_1, x_1 = x_r - \alpha(x_r - x_l), \tag{2.4.5}$$

and the function is evaluated at the newly generated point x_1 while the reduced interval is $[x_l, x_r]$. If $f(x_2) < f(x_1)$, then the four points are obtained sequentially as,

$$x_l = x_1, x_r = x_r, x_1 = x_2, x_2 = x_l + \alpha(x_r - x_l), \tag{2.4.6}$$

and the function is evaluated at the newly generated point x_2 while the reduced interval is $[x_l, x_r]$. The process is repeated until the desired accuracy is reached. This is determined by the following criterion,

$$|x_r - x_l| > \varepsilon(|x_1| + |x_2|), \tag{2.4.7}$$

where ε is a predefined tolerance value. When the condition in (2.4.7) is violated or the number of iterations becomes large, the algorithm is stopped. The minimiser is $x_m = x_1$ if $f(x_1) \leq f(x_2)$; otherwise, it is $x_m = x_2$.

The second category uses curve fitting techniques including quadratic and cubic interpolators. Generally, the interpolating methods perform better than the local search methods, but they require that the function be highly smoothed (Ravindran et al., 2006). The quadratic fitting curve method – Powell's quadratic fit – starts with three points x_0, x_1, x_2 such that,

$$f(x_0) > f(x_1) < f(x_2). \tag{2.4.8}$$

This is the bracketing condition. A quadratic curve is given as,

$$y = ax^2 + bx + c. \tag{2.4.9}$$

The quadratic curve (2.4.9) crosses the three points when the following three equations in three unknowns are satisfied,

$$\begin{aligned} ax_0^2 + bx_0 + c &= f(x_0) = f_0 \\ ax_1^2 + bx_1 + c &= f(x_1) = f_1 \\ ax_2^2 + bx_2 + c &= f(x_1) = f_2. \end{aligned} \tag{2.4.10}$$

Using linear algebra, one obtains the values of a, b, c as,

$$\begin{aligned} a &= \frac{(x_1 - x_2)f_0 + (x_2 - x_0)f_1 + (x_0 - x_1)f_2}{(x_1 - x_2)(x_0^2 - x_2^2) - (x_0 - x_2)(x_1^2 - x_2^2)} \\ b &= \frac{(x_1^2 - x_2^2)f_0 + (x_2^2 - x_0^2)f_1 + (x_0^2 - x_1^2)f_2}{(x_1^2 - x_2^2)(x_0 - x_2) - (x_0^2 - x_2^2)(x_1 - x_2)} \\ c &= \frac{x_1^2 x_2^2 (x_2 - x_1)f_0 + x_0 x_2^2 (x_0 - x_2)f_1 + x_0 x_1 x_2 (x_1 - x_0)f_2}{x_1(x_1 - x_2)(x_0^2 - x_2^2) - x_0(x_0 - x_2)(x_1^2 - x_2^2)}, \end{aligned} \tag{2.4.11}$$

and the minimiser of the quadratic fitting curve is given as,

$$x_m = -\frac{b}{2a} = \frac{1}{2} \frac{(x_1^2 - x_2^2) f_0 + (x_2^2 - x_0^2) f_1 + (x_0^2 - x_1^2) f}{(x_1 - x_2) f_0 + (x_2 - x_0) f_1 + (x_0 - x_1) f_2}, \tag{2.4.12}$$

provided that the condition $a>0$ is satisfied. It follows that, when the objective function is quadratic, the method requires only one step to attain the minimum value. The algorithm for the quadratic fitting curve is outlined as follows:

1. Find 3 points x_0, x_1, x_2 that satisfy the conditions in equation (2.4.8).
2. Compute the minimiser of the quadratic curve using equation (2.4.12).
3. Compute $f(x_m)$ and let $x^* \in \{x_0, x_1, x_2, x_m\}$ such that $f(x^*)$ is the minimum among the four values.
4. If x_m is within the required accuracy from x_1, the minimiser is the one that satisfies the condition $\min\{f(x_m), f(x_1)\}$. Else go to step 5 or 6.
5. If $x^* = x_1$, either $x_1 \in [x_0, x_m]$, discard x_2 and replace it by x_m and rename the remaining three points as $x_0 = x_0$, $x_1 = x_1$, $x_2 = x_m$ and return to step 2. Or $x_1 \in [x_m, x_2]$, discard x_0 and replace it by x_m and rename the remaining three points as $x_0 = x_m$, $x_1 = x_1$, $x_2 = x_2$ and return to step 2.
6. If $x^* = x_m$, either $x_m \in [x_0, x_1]$, discard x_2 and rename the remaining three points as $x_0 = x_0$, $x_1 = x_m$, $x_2 = x_1$ and return to step 2. Or $x_m \in [x_1, x_2]$, discard x_0 and rename the remaining three points as $x_0 = x_1$, $x_1 = x_m$, $x_2 = x_2$ and return to step 2.

The cubic interpolating method – Davidon's cubic interpolation method – is generally better than Powell's method, but it requires, in addition to the function values, first derivative and thus more computation. For further reading one may consult (Powell, 1964; Rao, 1996).

There are two optimality conditions used to help obtain the local minimiser rather than using the function valuations. The first condition is necessary while the second is sufficient. A local minimiser has to satisfy the necessary condition, still there may exist some other non-minimising points that satisfy the same condition. The first optimality condition requires that the first derivative is zero,

$$f^{'}(x)=0. \tag{2.4.13}$$

Those points that satisfy the first optimality condition are called stationary points. On the other hand, if a point satisfies the sufficient condition, then it is assured to be a local minimiser yet the sufficient condition is not necessary. The second optimality states that if the function and its first n derivatives are continuous, then the function has a local minimiser if and only if n, the order of the first non-vanishing derivative, is even and the n^{th} derivative is positive,

$$f^{(n)}(x^{*})>0. \tag{2.4.14}$$

When a function satisfies the condition,

$$f\ (\lambda x+(1-\lambda)y\ )\leq \lambda f\ (x\ )+(1-\lambda)f\ (x\ ), \tag{2.4.15}$$

for all $x\ , y\ \in \mathbb{R}$ and $\lambda \in [0,1]$, then the function is convex. If a convex function satisfies the first optimality condition at a point, then such point is a global minimiser.

The third category requires that the objective function is both continuous and differentiable. Two famous methods of this category are Newton-Raphson method and the secant method (Burden and Fares, 1989). A smooth one-dimensional function can be expressed mathematically in a Taylor series as,

$$f(x)=f(x_0)+f^{'}(x_0)(x-x_0)+\frac{1}{2!}f^{''}(x_0)(x-x_0)^2+\frac{1}{3!}f^{'''}(x_0)(x-x_0)^3+\ldots\ . \tag{2.4.16}$$

Truncating the series, one obtains different polynomial approximations to the function, within the neighbourhood of x_0. The quadratic approximation is obtained by retaining the first three terms,

$$q(x) = f(x_0) + f^{'}(x_0)(x - x_0) + \frac{1}{2!} f^{''}(x_0)(x - x_0)^2. \qquad (2.4.17)$$

Applying first optimality condition, one obtains,

$$f^{'}(x_0) + f^{''}(x_0)(x - x_0) = 0. \qquad (2.4.18)$$

Rearranging equation (2.4.18), the Newton method is given as,

$$x_{k+1} = x_k - [f^{''}(x_k)]^{-1} f^{'}(x_k), \quad k = 0,1,2,\ldots \; . \qquad (2.4.19)$$

It follows that the Newton's method for a quadratic approximation requires the first and the second derivatives and it requires a single iteration to converge to the optimiser when the objective function is quadratic. When the second derivative is close to zero, the Newton's method will fail to converge to the minimiser. The formula generates a sequence with a quadratic convergence when the initial guess is close to the minimiser. When approximating the second derivative by a secant line, we obtain,

$$f^{''}(x_k) \cong \frac{f^{'}(x_k) - f^{'}(x_{k-1})}{x_k - x_{k-1}}. \qquad (2.4.20)$$

Substituting in equation (2.4.19), this leads to the secant method,

$$x_{k+1} = x_k - \frac{(x_k - x_{k-1})}{[f^{'}(x_k) - f^{'}(x_{k-1})]} f^{'}(x_k), \quad k = 1,2,3,\ldots \; . \qquad (2.4.21)$$

The secant method uses the first derivative, but requires two initial values. It has a super linear convergence rate and takes a single iteration to converge when dealing with quadratic function. Such methods are useful to improve a result found by another method.

2.4.2 Multivariable Optimisation Methods

Now we are ready to treat the multi-dimensional problem. However, I will only deal with the gradient based methods as they are more efficient than the other methods. I will consider real-valued function $f(x)$ in a multi-dimensional space,

$$f : □^n \to □, \qquad (2.4.22)$$

such that $\boldsymbol{x} = [x_1, ..., x_n] \in \mathbb{R}^n$ and the unconstrained problem extends to multi-dimensional as,

$$\text{minimise} \quad f(\boldsymbol{x}), \tag{2.4.23}$$

such that $f(\boldsymbol{x})$ has continuous second derivatives. Further, the definitions of unimodality, convexity, and local minimum extend to the multi-dimensional seamlessly.

2.4.2.1 Newton's Method

The quadratic approximation for $f(\boldsymbol{x})$ is expressed as,

$$q(\boldsymbol{x}) = f(\boldsymbol{x}_0) + [\nabla f(\boldsymbol{x}_0)]^t \cdot (\boldsymbol{x} - \boldsymbol{x}_0) + \frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}_0)^t \cdot H(\boldsymbol{x} - \boldsymbol{x}_0), \tag{2.4.24}$$

where ∇ is the n-dimensional derivative operator and H is the Hessian matrix of $f(\boldsymbol{x})$ evaluated at $\boldsymbol{x}_0$ and denoted as $\nabla^2 f(\boldsymbol{x}_0)$. The Hessian matrix is symmetric because $f(\boldsymbol{x})$ is smooth. After applying the first optimality condition and simplifying the result, we obtain the Newton method,

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - H_k^{-1} \nabla f(\boldsymbol{x}_k) \quad k = 0, 1, 2, 3, ... \ . \tag{2.4.25}$$

Equation (2.4.25) requires a starting point $\boldsymbol{x}_0$ (initial guess) to iteratively generate a sequence of solutions which may converge to a minimiser, a maximiser, or a saddle point. One iteration is required if the function is quadratic; otherwise, one needs to use equation (2.4.25) iteratively since equation (2.4.24) disregards the higher order terms. When the initial guess is close to the desired solution, the Newton method becomes very robust as it is perfect for local convergence with a quadratic convergence rate. That is the case when the difference between the objective function and the quadratic approximation is very close (Nocedal and Wright, 1999). Another adverse property is the fact that Newton method requires the Hessian matrix to be positive definite (non-singular) to converge. Also, it requires the first and the second derivatives which are comparatively expensive to compute. It is tempting to conclude that Newton method is ineffective despite its convergence rate. Still, one may transform Newton method to other forms to overcome such problems or use it with other method, such as the conjugate gradient method which I will discuss in section 2.4.2.4.

2.4.2.2 Descent Direction

When optimising, we have a current solution and we want to move to a next solution such that the new solution is a better minimiser. Therefore, we need to find the optimal direction that leads to the next point. At this stage, we have the current position and the direction, but we lack the step length. A large or a small step length may cause other problems. Consequently, we need to obtain the optimal step length. This notion will lead to the descent direction. Remember that the direction along the gradient produces the fastest rate of increase and known as steepest ascent. Unfortunately, this property is a local one. That is, the direction along the gradient is changing from one point to another. Let me define the gradient of a function at the point $\boldsymbol{x}_k$ as $\boldsymbol{g}(\boldsymbol{x}_k)$ and $\boldsymbol{s} \in \mathbb{R}^n$ to be a descent direction. This leads to,

$$\boldsymbol{g}(\boldsymbol{x}_k)\cdot \boldsymbol{s} < 0, \tag{2.4.26}$$

One can rewrite equation (2.4.26) in the following form,

$$f(\boldsymbol{x}_k + \alpha \boldsymbol{s}) < f(\boldsymbol{x}_k), \tag{2.4.27}$$

where α is a very small and positive number. Define $x^i = x_k^i + \alpha s^i$, then we have,

$$\frac{df}{d\alpha} = \sum_{i=i}^{n} \frac{\partial f}{\partial x^i}\frac{\partial x^i}{\partial \alpha}. \tag{2.4.28}$$

The function $\varphi(\alpha) = f(\boldsymbol{x}_k + \alpha \boldsymbol{s})$ is a univariate function in terms of α and its derivative with respect to α is given by equation (2.4.28) and the outcome of applying the derivative is given by equation (2.4.26). That is, the derivative $\varphi'(\alpha) < 0$ and $\varphi(\alpha) < \varphi(0)$ where α is sufficiently small and positive.

The condition in (2.4.27) suggests that the descent direction $\boldsymbol{s}$ with step length α produces a new solution that reduces the function. That is,

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha \boldsymbol{s}. \qquad (2.4.29)$$

Equation (2.4.29) gives us the solution we are after. It relates the current position, the direction along which to proceed, and the right step length to the next position. I will derive different types of descent directions. I will also address the step length parameter in a later section.

Comparing equations (2.4.29) and (2.4.25), the direction for Newton's method is given as,

$$\boldsymbol{s} = -[\nabla^2 f(\boldsymbol{x}_k)]^{-1} \nabla f(\boldsymbol{x}_k), \qquad (2.4.30)$$

while the step length is one. Applying the n-dimensional derivative to the last equation, we obtain,

$$\nabla f(\boldsymbol{x}_k) \cdot [\nabla^2 f(\boldsymbol{x}_k)]^{-1} \nabla f(\boldsymbol{x}_k) > 0. \qquad (2.4.31)$$

Condition (2.4.31) is the requirement for $[\nabla^2 f(\boldsymbol{x}_k)]^{-1}$ which is the inverse of the Hessian matrix to be positive definite. But, this is true if and only if the Hessian is positive definite (Burden and Fares, 1989). In other words, equation (2.4.30) is a descent direction when H is positive definite. It is possible to apply a variable step length to the Newton's method to read,

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \alpha_k H_k^{-1} \nabla f(\boldsymbol{x}_k) \quad k = 0,1,2,3,\ldots \ . \qquad (2.4.32)$$

Equation (2.4.32) may stop Newton's method from diverging or heading to a local maximiser or saddle point when the objective function is nonquadratic. However, it does not address the problem of positive definiteness of the Hessian. Further, it demands huge memory to store the Hessian and large computations to calculate the second derivative. It also requires the inversion of the Hessian at each step and the multiplication of the Hessian matrix by the gradient. Therefore, this method is impractical for solving problems with many parameters.

Another way to modify the Newton's method is to use the SVD (single value decomposition) method to obtain all the eigenvalues of the Hessian matrix such that

$\lambda_k^1 < \lambda_k^2 < \ldots < \lambda_k^n$ and choose small ε_k such that $\varepsilon_k + \lambda_k^1 = \upsilon > 0$ where υ is a small positive number. When ε_k is added to all elements in the main diagonal of the Hessian matrix, it is modified into a positive definite one. In the case that λ_k^1 is positive and large enough, there is no need to modify the Hessian as it is positive definite. We can also use Cholesky decomposition known as LDU factorization, where L stands for a lower triangular matrix with ones along the diagonal, D is a diagonal matrix and U is the transpose of L. In this case, the diagonal matrix D is modified by adding ε_k to the diagonal to enforce that the least element is positive enough such that the Hessian is positive definite. Nonetheless, such operations are expensive and are not protected from errors. Concurrently, one can modify the Hessian by adding a large enough positive scalar to the main diagonal of the Hessian to convert such a matrix to a positive definite one at each step. Still, one is faced with deciding such scalars. One way to overcome such a problem is to apply first a large scalar, like the largest element in the matrix. In the case that the scalar does not enforce the Hessian into a positive definite matrix, one might double the scalar iteratively until the desired result is obtained. On the other hand, if the applied scalar produces a positive definite matrix, one can halve the scalar iteratively to reach the minimum scalar that makes the Hessian positive definite. The problem reduces to the steepest descent method, described in the next section, when large scalars are used. In practice, one wants to start with steepest descent as it has slow global convergence and finish with Newton's method because it has a fast local convergence. This method is known as Marquardt method (Marquardt, 1963).

In the next two sections, I present two different descent directions methods yet very related. Such directions lead to rather simple optimisation schemes with global convergence. When the Hessian is replaced with an approximation, the method is referred to as quasi-Newton. I will examine this class of methods in section 2.4.2.5.

Optimising the step length parameter is saved for section 2.4.2.6 and I finish the chapter with a discussion on the stopping criteria for the given algorithms.

2.4.2.3 Steepest Descent Method

Choosing $\boldsymbol{s} = -\nabla f(\boldsymbol{x}_k)$, which is known as the steepest descent direction, one obtains,

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \alpha_k \nabla f(\boldsymbol{x}_k) \quad k = 0, 1, 2, 3, \ldots \ . \tag{2.4.33}$$

This is the well-known steepest descent method where it starts from an initial guess and iteratively moves along the steepest descent directions until reaching the minimiser. In 1874, Cauchy was the first to use steepest descent direction for minimisation. It requires that the gradient is evaluated at every individual step to generate the direction. Also, it requires that the step length parameter α_k is optimised at each iteration to produce the optimal step length required for the move. The steepest descent method is a special type of the quasi-Newton methods where the Hessian matrix is approximated by the identity matrix. This leads to making the steepest descent method independent of the second derivative, the curvature. The method has a global convergence though very slow due to the fact that the steepest ascent (descent) is a local property (Nocedal and Wright, 1999; Rao, 1996). Therefore, this method is relatively ineffective.

2.4.2.4 Conjugate Gradient Method

The slow convergence of steepest descent method can be enhanced by the logic of the conjugate direction method (Powell, 1964). Instead of optimising along the steepest directions, one needs to form an orthogonal basis of descent orthogonal directions derived from the gradients and hence the name conjugate gradient method. This leads to minimising the objective function in n, the number of parameters, iterations along the conjugate directions (Nocedal and Wright, 1999). The conjugate gradient method was developed separately by Steifel and Hestenes for linear problems (Hestenes and Stiefel, 1952), but it was much later that the nonlinear conjugate gradient method was proposed

(Fletcher and Reeves, 1964). It is the most common method used for solving large linear problems (Scales et al., 2001).

We assume that the objective function is approximated by a quadratic function,

$$f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^t A\boldsymbol{x} + \boldsymbol{b}^t\boldsymbol{x} + c. \tag{2.4.34}$$

Consider the set of steepest descent directions (vectors) generated by the steepest descent method. We want to form an orthogonal basis out of the set of steepest directions. Let the first direction s_1 be given by the first steepest descent direction and the new point $\boldsymbol{x}_2$ be given by equation (2.4.29) such that the first direction is orthogonal to the second gradient $\boldsymbol{g}_2$,

$$\begin{aligned} \boldsymbol{s}_1 &= -A\boldsymbol{x}_1 - \boldsymbol{g}_1 \\ \boldsymbol{x}_2 &= \boldsymbol{x}_1 + \alpha\boldsymbol{s}_1 \\ \boldsymbol{s}_1^t\boldsymbol{g}_2 &= 0. \end{aligned} \tag{2.4.35}$$

Substituting $A\boldsymbol{x}_2 + \boldsymbol{b}$ for $\boldsymbol{g}_2$, one obtains,

$$\boldsymbol{s}_1^t[A\boldsymbol{x}_2 + \boldsymbol{b}] = \boldsymbol{s}_1^t[A(\boldsymbol{x}_1 + \alpha_1\boldsymbol{s}_1) + \boldsymbol{b}] = 0, \tag{2.4.36}$$

and $A\boldsymbol{x}_1 + \boldsymbol{b} = \boldsymbol{s}_1$, this leads to,

$$\alpha_1 = -\frac{\boldsymbol{s}_1^t\boldsymbol{g}_1}{\boldsymbol{s}_1^t A\boldsymbol{s}_1}. \tag{2.4.37}$$

Using Gram-Schmidt orthogonalization process, the second search direction is expressed as a linear combination of the first search direction and the second steepest descent,

$$\boldsymbol{s}_2 = -\boldsymbol{g}_2 + \beta_2\boldsymbol{s}_1. \tag{2.4.38}$$

We need to obtain β_2 in order to make $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$ conjugate,

$$\boldsymbol{s}_1^t A\boldsymbol{s}_2 = 0. \tag{2.4.39}$$

Substituting (2.4.38) into (2.4.39), one gets,

$$\boldsymbol{s}_1^t A(-\boldsymbol{g}_2+\beta_2\boldsymbol{s}_1)=\frac{(\boldsymbol{x}_1-\boldsymbol{x}_1)^t}{\alpha_1}A(-\boldsymbol{g}_2+\beta_2\boldsymbol{s}_1)=0. \tag{2.4.40}$$

But, we have $(\boldsymbol{g}_2-\boldsymbol{g}_1)=A(\boldsymbol{x}_2-\boldsymbol{x}_1)$ and $A^t=A$, this leads to,

$$(\boldsymbol{g}_2-\boldsymbol{g}_1)^t(\boldsymbol{g}_2-\beta_2\boldsymbol{s}_1)=0. \tag{2.4.41}$$

Simplifying and making use of $\boldsymbol{g}_1^t\boldsymbol{g}_2=-\boldsymbol{s}_1^t\boldsymbol{g}_2=0$, one obtains,

$$\beta_2=\frac{\boldsymbol{g}_2^t\boldsymbol{g}_2}{\boldsymbol{g}_1^t\boldsymbol{g}_1}. \tag{2.4.42}$$

Continuing in the same manner, I obtain two formulas required for generating the conjugate direction such that $\boldsymbol{s}_k$ and β_k are given by,

$$\begin{aligned}\boldsymbol{s}_k&=-\boldsymbol{g}_k+\beta_k\boldsymbol{s}_{k-1}\quad k=2,3,4,\ldots\\ \beta_k&=\frac{\boldsymbol{g}_k^t\boldsymbol{g}_k}{\boldsymbol{g}_{k-1}^t\boldsymbol{g}_{k-1}}\quad k=2,3,4,\ldots\ .\end{aligned} \tag{2.4.43}$$

Equation (2.4.43) represents the search direction for the Fletcher-Reeves conjugate gradient method. It follows that the current search direction depends only on the previous search direction and all the steepest directions up to the current one. Thus it requires little storage and computation (Nocedal and Wright, 1999). The convergence may go beyond n iterations for ill-conditioned problem due to the cumulative effect of rounding errors when generating the search direction and/or inaccuracies when optimising the step length parameter (Rao, 1996). Therefore, in practice, the method is restarted by setting the new search direction to the steepest descent direction whenever the number of iterations becomes a multiple of the parameter number. Like the steepest descent method, the conjugate gradient method has a global convergence; however, it is less efficient and robust when compared to the quasi-Newton methods (Nocedal and Wright, 1999; Rao, 1996; Ravindran et al., 2006).

Another important variant of the Fletcher-Reeves method is the Polak-Ribere conjugate gradient method and its formula is given as,

$$\begin{aligned} \boldsymbol{s}_k &= -\boldsymbol{g}_k + \beta_k \boldsymbol{s}_{k-1} \quad k = 2,3,4,\dots \\ \beta_k &= \frac{\boldsymbol{g}_k^t(\boldsymbol{g}_k - \boldsymbol{g}_{k-1})}{\boldsymbol{g}_{k-1}^t \boldsymbol{g}_{k-1}} \quad k = 2,3,4,\dots \ . \end{aligned} \tag{2.4.44}$$

The Polak-Ribere method is more robust and efficient (Nocedal and Wright, 1999). The following is a generic algorithm for the conjugate gradient methods:

```
1. Start with initial point $\boldsymbol{x}_0$ and evaluate $f_0 = f(\boldsymbol{x}_0)$ and $\boldsymbol{g}_0 = \nabla f(\boldsymbol{x}_0)$
2. Set the first search direction $\boldsymbol{s}_0 = -\boldsymbol{g}_0$ and set the counter $k \leftarrow 0$
   While ( $\boldsymbol{g}_k \neq 0$ )
           Compute the optimal step length $\alpha_k$ in the search direction $\boldsymbol{g}_k$
           Set $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{s}_k$
           Evaluate $\boldsymbol{g}_{k+1}$
           Set $\beta_{k+1}^{F-R} \leftarrow \frac{\boldsymbol{g}_{k+1}^t \boldsymbol{g}_{k+1}}{\boldsymbol{g}_k^t \boldsymbol{g}_k}$ or $\beta_{k+1}^{P-R} \leftarrow \frac{\boldsymbol{g}_{k+1}^t(\boldsymbol{g}_{k+1} - \boldsymbol{g}_k)}{\boldsymbol{g}_k^t \boldsymbol{g}_k}$
           Set $\boldsymbol{s}_{k+1} \leftarrow -\boldsymbol{g}_{k+1} + \beta_{k+1} \boldsymbol{s}_k$
           Set $k \leftarrow k+1$
   End
```

2.4.2.5 Quasi-Newton Methods

Now it is time to address the problem of the Hessian when it is not positive definite. Quasi-Newton methods are designed to use an approximation of the Hessian matrix or its inverse at each iteration by updating a symmetric positive definite matrix B^k in order to avoid computing the second derivative and to produce a descent direction. I assume that I have estimated the matrix B^k for the kth iteration and it is symmetric positive definite. I also assume that I have the solution $\boldsymbol{x}_k$ and the gradient $\boldsymbol{g}_k$ at the same step. I use equation (2.4.32) to obtain the next solution $\boldsymbol{x}_{k+1}$ and hence the corresponding gradient $\boldsymbol{g}_{k+1}$. I need to approximate B^{k+1} from what I know.

I use the fact that the Hessian is simply the divergence of the gradient. But, before I proceed I want to make some simplifications,

$$\begin{aligned} \boldsymbol{\delta}_k &= \boldsymbol{x}_{k+1} - \boldsymbol{x}_k \\ \boldsymbol{\gamma}_k &= \boldsymbol{g}_{k+1} - \boldsymbol{g}_k . \end{aligned} \qquad (2.4.45)$$

Using the Taylor theorem, one can expand $\boldsymbol{g}_k$ as,

$$\boldsymbol{g}_k = \boldsymbol{g}_{k+1} - A_{k+1}\boldsymbol{\delta}_k + \dots \ . \qquad (2.4.46)$$

When the function is quadratic, the equations are exact. Further, the Hessian details the curvature about which I hope to obtain some information from B_k and B_{k+1} when moving from $\boldsymbol{x}_k$ to $\boldsymbol{x}_{k+1}$. Truncating the above equation and simplifying, I obtain,

$$\boldsymbol{\gamma}_k = A_{k+1}\boldsymbol{\delta}_k \Rightarrow B_{k+1}\boldsymbol{\gamma}_k = \boldsymbol{\delta}_k . \qquad (2.4.47)$$

Equation (2.4.47) is known as the secant method. The matrix B_{k+1} is an $n \times n$ symmetric matrix and thus it has at most $(n^2+n)/2$ entries. But, the secant equation consists of n equations. Requiring that the B_{k+1} be positive definite reduces the ambiguity by n; therefore, the secant equation cannot produce a unique approximation to the Hessian. If one assumes that there exists a symmetric positive definite matrix that satisfies the secant equation, then multiplying the secant equation by $\boldsymbol{\gamma}_k^t$ gives,

$$\boldsymbol{\gamma}_k\boldsymbol{\delta}_k = \boldsymbol{\gamma}_k B_{k+1}\boldsymbol{\gamma}_k > 0 \Rightarrow (\boldsymbol{g}(\boldsymbol{x}_{k+1}) - \boldsymbol{g}(\boldsymbol{x}_k))(\boldsymbol{x}_{k+1} - \boldsymbol{x}_k) > 0. \qquad (2.4.48)$$

This inequality is known as the curvature condition and it holds when the function is strictly convex. In other words, the function has a positive curvature. When the curvature condition is not satisfied, I fix this problem by applying a suitable line search such that it enforces the Wolfe condition, described in section 2.4.3, which in turns enforces the curvature condition explicitly.

To construct B_{k+1} one needs to incorporate the curvature information within B_k such that B_{k+1} is expressed as,

$$B_{k+1} = B_k + \Delta B_k. \tag{2.4.49}$$

Two methods are commonly used for evaluating ΔB_k known as rank *1* and rank *2*. The update ΔB_k matrix can have a rank up to n, but it is impractical to assume that it has a rank three or more. Still rank *1* methods do not guarantee that the generated matrix is positive definite; consequently, the search direction does not necessarily produce a descent direction. Rank *2* methods, on the other hand, always generate a symmetric positive definite matrix. Such methods are also known as variable metric methods and B_{k+1} called the metric. Rank *2* update can be expressed as (Rao, 1996),

$$B_{k+1} = B_k + a\mathbf{u}\mathbf{u}^t + b\mathbf{v}\mathbf{v}^t, \tag{2.4.50}$$

where a and b are constants and $\mathbf{u}$ and $\mathbf{v}$ are n-component vectors that all need to be determined. Demanding the secant and the curvature conditions be satisfied, one obtains,

$$\boldsymbol{\delta}_k = B_k\boldsymbol{\gamma}_k + a\mathbf{u}\mathbf{u}^t\gamma_k + b\mathbf{v}\mathbf{v}^t\boldsymbol{\gamma}_k, \tag{2.4.51}$$

it follows that the two vectors are not determined uniquely. Assume that,

$$\begin{aligned} \mathbf{u} &= \boldsymbol{\delta}_k \\ \mathbf{v} &= B_k\boldsymbol{\gamma}_k. \end{aligned} \tag{2.4.52}$$

Then, we have,

$$\begin{aligned} a\mathbf{u}^t\boldsymbol{\gamma}_k &= 1 \Rightarrow a = \frac{1}{\mathbf{u}^t\boldsymbol{\gamma}_k} \\ b\mathbf{v}^t\boldsymbol{\gamma}_k &= -1 \Rightarrow b = -\frac{1}{\boldsymbol{\gamma}_k^t B_k\boldsymbol{\gamma}_k}. \end{aligned} \tag{2.4.53}$$

Substituting (2.4.53) into equation (2.4.50), one obtains,

$$B_{k+1} = B_k + \frac{\boldsymbol{\delta}_k \boldsymbol{\delta}_k^t}{\boldsymbol{\delta}_k^t \boldsymbol{\gamma}_k} - \frac{B_k \boldsymbol{\gamma}_k \boldsymbol{\gamma}_k^t B_k}{\boldsymbol{\gamma}_k^t B_k \boldsymbol{\gamma}_k}. \tag{2.4.54}$$

This is the Davidon-Fletcher-Powell (DFP) formula for updating the inverse of the Hessian (Davidon, 1959; Fletcher, 1963). An even better updating formula is the BFGS (Broyden-Fletcher-Goldfarb-Shanno) (Broyden, 1970; Fletcher, 1970; Goldfarb, 1977; Shanno, 1970). The main advantage of the BFGS is the fact that it is less dependent on exact line search and the formula is given as (Rao, 1996),

$$B_{k+1} = B_k + (1 + \frac{\boldsymbol{\gamma}_k^t B_k \boldsymbol{\gamma}_k}{\boldsymbol{\delta}_k^t \boldsymbol{\gamma}_k}) \frac{\boldsymbol{\delta}_k \boldsymbol{\delta}_k^t}{\boldsymbol{\delta}_k^t \boldsymbol{\gamma}_k} - (\boldsymbol{\delta}_k \boldsymbol{\gamma}_k^t B_k + B_k \boldsymbol{\gamma}_k \boldsymbol{\delta}_k^t) \frac{1}{\boldsymbol{\delta}_k^t \boldsymbol{\gamma}_k}. \tag{2.4.55}$$

The rank two updating formulae require only the gradient to generate symmetric and positive definite approximation of the Hessian or its inverse. The above two equation are used to update an estimation of the Hessian inverse and thus the quasi-Newton methods do not need to solve a linear system as the case for Newton’s method. Further, the variable metric methods have a globally superlinear convergence which may prove to be more efficient than Newton’s method when it is expensive to calculate the second derivative. Of all methods discussed so far, the variable metric with the BFGS is superior (Carpenter and Smith, 1975; Himmelblau, 1972; Sargent and Sebastian, 1971; Shanno and Phua, 1978a, b).

The quasi-Newton formula for updating solution is given as,

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \alpha_k B_k \boldsymbol{g}_k \quad k = 0, 1, 2, 3, \ldots\ . \tag{2.4.56}$$

One remaining unresolved issue is the initial estimation of the Hessian inverse. Since the steepest descent method has global convergence, though comparatively slow, one can start the variable metric methods as a steepest descent method and hence uses a multiple of the identity matrix as the initial approximation for the Hessian inverse. This choice satisfies the requirements. Hereafter, I describe a generic algorithm for the variable metric methods:

1. Start with initial solution $\boldsymbol{x}_0$ and evaluate the initial gradient $\boldsymbol{g}_0$.

```
2. Set the first search direction $B_0 = -\omega I$
3. set the counter $k \leftarrow 0$
   while ( $\boldsymbol{g}_k > \varepsilon$ )
        i.   $\boldsymbol{s}_k = -B_k \boldsymbol{g}_k$
        ii.  $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \alpha_k \boldsymbol{s}_k$ where $\alpha_k$ satisfies the Wolfe strong condition
        iii. $\boldsymbol{\delta}_k = \boldsymbol{x}_{k+1} - \boldsymbol{x}_k$
        iv.  $\boldsymbol{\gamma}_k = \boldsymbol{g}_{k+1} - \boldsymbol{g}_k$
        v.   Compute $B_{k+1}$ using DFP (2.4.54) or BFGS (2.4.55)
        vi.  Set $k \leftarrow k+1$
   end
```

2.4.3 Step Length Optimisation

So far, I have explained different optimisation methods that use descent search direction. Such direction leads to reduction of the function. In this section, I focus on optimising the step length parameter to produce the optimal reduction. Following the discussion in section 2.4.2.2, the step length parameter is expressed as,

$$\varphi(\alpha) = f(\boldsymbol{x}_k + \alpha \boldsymbol{s}) \quad \alpha > 0. \qquad (2.4.57)$$

This function is univariate because the current solution and the descent search direction are known. Therefore, we can obtain the global minimum of such a function by using the methods of section 2.4.1 which will lead to the optimal step length when the function is quadratic or nearly quadratic. Those methods that are used to optimise a univariate function are called line search methods. Section 4.2.1 describes the exact methods which may not produce the global minimiser and are expensive to use even for local minimiser (Nocedal and Wright, 1999; Rao, 1996; Ravindran et al., 2006). Remember that we use an iterative quadratic model to solve for nonlinear problem. The practical alternative is to use inexact methods as long as they provide adequate reduction to the objective function. Such methods require sufficiently less computations, compared to the exact methods; making them more attractive. Still, the inexact methods may face two major problems making them useless.

It is possible for a line search method to continuously induce a reduction to the objective function without causing convergence. The first scenario is when the function has a valley, but the iterative solutions jump from one side of the valley to the other side such that the function reduction is too little compared to the step length and hence no convergence to the minimiser is obtained. To avoid this situation, step length parameter α_k has to satisfy the Armijo condition,

$$f(\boldsymbol{x}_k + \alpha_k \boldsymbol{s}_k) \leq f(\boldsymbol{x}_k) + c_1 \alpha_k \boldsymbol{g}_k \cdot \boldsymbol{s}_k . \tag{2.4.58}$$

The quantity $\alpha_k \boldsymbol{g}_k \cdot \boldsymbol{s}_k$ is negative because the search direction $\boldsymbol{s}_k$ is descent and the step length α_k is positive. It accounts for the reduction in the objective function caused by the slope at $\boldsymbol{x}_k$ in the search direction. It follows from (2.4.26) and (2.4.27) that the inequality holds for α_k only when we choose the constant $c_1 \in (0,1)$ small enough (Nocedal and Wright, 1999). Typical value for c_1 is 10^{-4}. The right hand side of the inequality is a linear function with a negative slope but for α being sufficiently small lies above the graph of $\varphi(\alpha)$. The Armijo condition ensures that the step length does not over-step the minimiser and climbs the other side of the valley. Consequently, it ensures a sufficient reduction to the objective function and hence the name ‘sufficient decrease condition’. The sufficient decrease condition is shown in Figure (2.2) (Nocedal and Wright, 1999). The green dashed arrows show the values for α that satisfy the sufficient decrease condition.

The second scenario is when the step lengths are too small such that the function reduction is too little and thus no convergence to the minimiser is attained. This situation clearly satisfies the Armijo condition when c_1 is sufficiently small. In other words, the Armijo condition is not immune against steps being too short. To avoid this situation, the step length parameter must satisfy the curvature condition,

$$\boldsymbol{g}(\boldsymbol{x}_k+\alpha_k\boldsymbol{s}_k)\cdot\boldsymbol{s}_k \geq c_2\boldsymbol{g}(\boldsymbol{x}_k)\cdot\boldsymbol{s}_k, \tag{2.4.59}$$

where the constant $c_2 \in (c_1, 1)$ with a usual value of 0.9 for quasi-Newton methods and 0.1 for a conjugate gradient methods. The left hand side is the derivative of (2.4.57) at $\alpha = \alpha_k$ while $\boldsymbol{g}(\boldsymbol{x}_k)\cdot\boldsymbol{s}_k$ is the derivative of (2.4.57) at $\alpha = 0$. That is, for large enough α_k the slope of φ has increased over the slope at $\alpha = 0$ by some fixed amount. This leads to the fact that when $\varphi^{'}(\alpha)$ has a relatively large negative value; we can increase the step length to cause huge reduction in the objective function. When the slope approaches zero, on the other hand, indicating that we are close to the local minimiser, one may not be able to cause further reduction and hence it is logical to stop the search. The curvature condition is shown in Figure (2.2). The cyan dashed arrows show the two different intervals for which the step length parameter satisfies the curvature condition.

Adding $-\boldsymbol{g}(\boldsymbol{x}_k)\cdot\boldsymbol{s}_k$ to both sides of (2.4.59) and noting that $\boldsymbol{s}_k = \boldsymbol{x}_{k-1} - \boldsymbol{x}_k$, we obtain,

$$(\boldsymbol{g}(\boldsymbol{x}_k+\alpha_k\boldsymbol{s}_k)-\boldsymbol{g}(\boldsymbol{x}_k))\cdot(\boldsymbol{x}_{k+1}-\boldsymbol{x}_k)_k \geq (c_2-1)\boldsymbol{g}(\boldsymbol{x}_k)\cdot\boldsymbol{s}_k > 0, \tag{2.4.60}$$

This implies that the curvature condition (2.4.59) ensures that BFGS update is well-defined and hence the name. Therefore, to use the BFGS formula, we can use the inexact line search methods that satisfy the sufficient decrease and the curvature conditions as shown in Figure (2.2) by the solid red arrows.

Conditions (2.4.58) and (2.4.59) are known as the Wolfe conditions and sometimes referred to as Armijo-Goldstein conditions. Now consider the following condition,

$$\left|\boldsymbol{g}(\boldsymbol{x}_k+\alpha_k\boldsymbol{s}_k).\boldsymbol{s}_k\right|\le c_2\left|\boldsymbol{g}(\boldsymbol{x}_k)\cdot\boldsymbol{s}_k\right|. \quad (2.4.61)$$

Conditions (2.4.58) and (2.4.61) form the strong Wolfe conditions. The above condition requires the line search to seek for the actual minimiser as we are using the exact line search methods and is, hence, not efficient.

So far, I have described the conditions for the inexact line search to make adequate reduction along the given search direction. Still, it is possible to use only the sufficient decrease condition with the backtracking approach which ensures relatively long steps are taken whenever possible. However, this will not work for the variable metric methods as they require the curvature condition to ensure the update is well-defined. In addition, it is always possible to choose the step length that satisfies the Wolfe conditions such that the variable metric methods have global convergence (Nocedal and Wright, 1999).

The approach I use in this thesis is to satisfy the Wolfe conditions as I use the variable metric method with BFGS updating formula. The initial step length α_1 is chosen such that $\alpha_1 = 1$ for the variable metric methods and $0<c_1<c_2<1$ for Wolfe conditions. The algorithm consists of two stages. The first is designed to bracket the local minimiser of the univariate function. The algorithm starts with α_1 and checks if it satisfies an acceptable step length or it forms with $\alpha_0=0$ an interval that brackets the desired step length. If neither is satisfied, the trial step length is checked against the sufficient decrease condition and the curvature condition and modified. When a desired step length is bracketed in an interval, the interval size is successively decreased and an acceptable step length is obtained by using a quadratic or cubic interpolating curves. The quadratic curve is constructed either using three points with their function evaluations or two points with their valuations in addition to the derivative at either point. We always start with this scenario as we have $\varphi(0)$, $\varphi^{'}(0)$ and $\varphi(\alpha)$ from which a quadratic interpolator is constructed. The new minimiser is given as,

$$\alpha_{new} = \frac{\varphi^{'}(0)\alpha}{2(\varphi(\alpha) - \varphi(0) - \varphi^{'}(0)\alpha)}. \tag{2.4.62}$$

When α is outside the interval $[\sigma_1\alpha, \sigma_k\alpha]$, the new value is assigned to $\sigma_1\alpha$ if $\alpha_{new} < \sigma_1\alpha$; otherwise, it is assigned to $\sigma_k\alpha$. When the quadratic curve fitting method fails to generate a step length satisfying the sufficient decrease condition, the algorithm uses a cubic curve fitting method through three points with their values and the derivative specified at the first value.

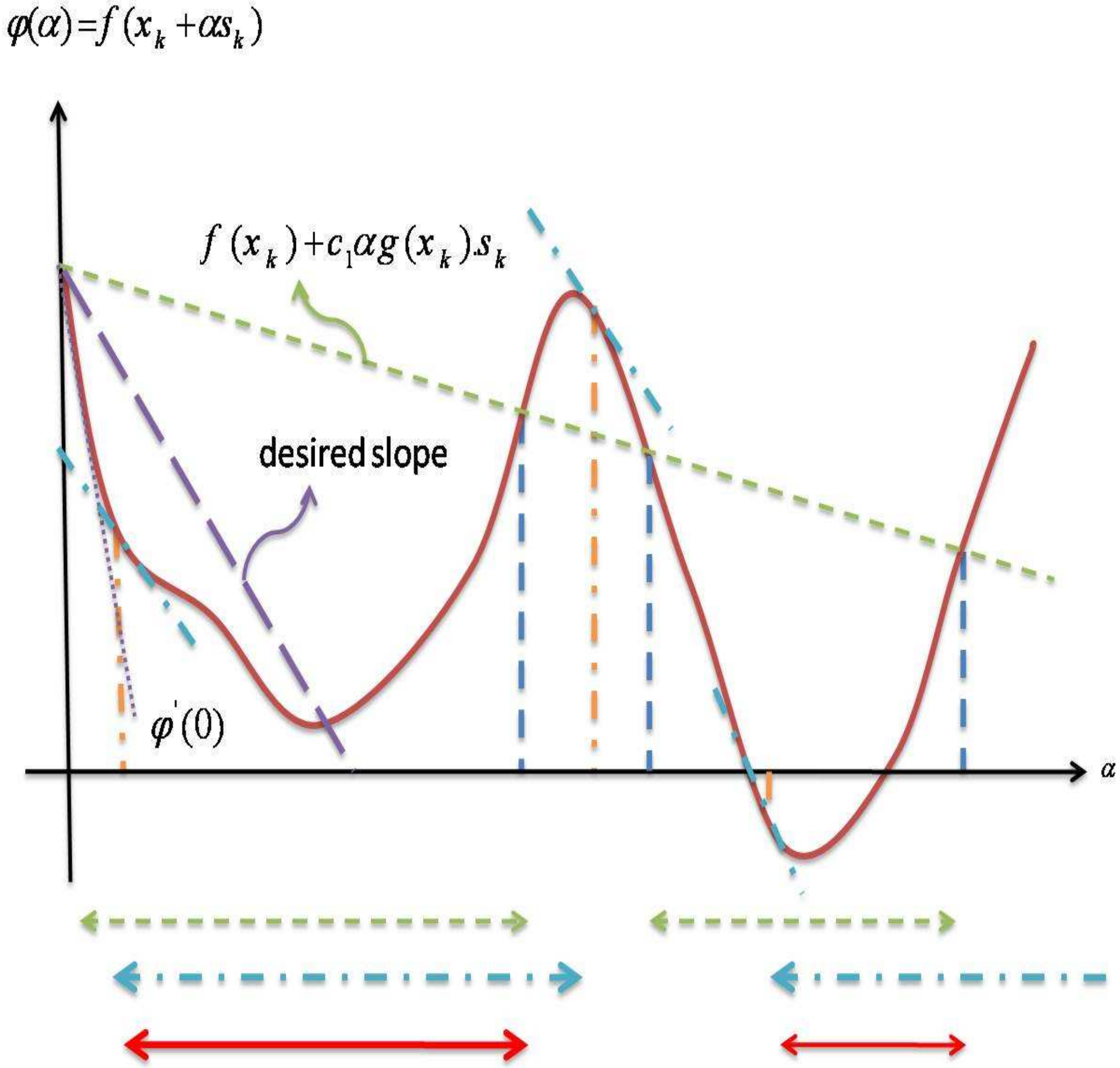

$\varphi(\alpha)=f(x_k+\alpha s_k)$
$f(x_k)+c_1\alpha g(x_k).s_k$
desired slope
$\varphi'(0)$
α

Figure (2.2): Acceptable step lengths satisfying the sufficient decrease (Armijo) condition in green, the curvature condition in cyan, and the Wolfe conditions in red for a nonlinear function (after Nocedal and Wright, 1999).

2.4.4 Stopping Criteria

There are three main criteria that are commonly used to terminate an iterative algorithm. The first one uses the gradient of the objective function. This criterion is given as,

$$\|\boldsymbol{g}(\boldsymbol{x}_k)\| \le \varepsilon_1. \tag{2.4.63}$$

The value ε_1 is known as the gradient tolerance. To address the scaling problem, the above criterion is given as,

$$\max\left\{\frac{g^i(\boldsymbol{x}_k)\max\left\{\left|x_k^i\right|,\lambda_x\right\}}{\max\left\{f(\boldsymbol{x}_k),\lambda_f\right\}}\right\} \le \varepsilon_1, \tag{2.4.64}$$

such that λ_x,λ_f are relatively small values for the independent variables and the objective function, respectively to avoid dividing by zero.

The second criterion is to use the step size. When it is difficult to satisfy the gradient tolerance, then a secondary stopping criterion is employed. This terminating criterion is activated whenever the step length is too small. Similar to the gradient, one has to take into account the scaling of the problem and thus the terminating condition is given as,

$$\max\left\{\frac{\left|x_{k+1}^i - x_k^i\right|}{\max\left\{\|\boldsymbol{x}_k\|,\lambda_x\right\}}\right\} \le \varepsilon_2, \tag{2.4.65}$$

and ε_2 is called the step tolerance.

The third terminating criterion uses the function value. This criterion is given as,

$$\|F(\boldsymbol{x}_k)\| \le \varepsilon_3, \tag{2.4.66}$$

where ε_3 is referred to as the function tolerance. It is always desirable to terminate the algorithm after certain number of iterations as a safeguard. This is required because the convergence of nonlinear problems is difficult to predict.

References

Aster, R.C., B. Borchers, and C.H. Thurber, 2005, Parameter Estimation and Inverse Problems: Elsevier Academic Press.

Berryman, J.G., 1990, Stable iterative reconstruction algorithm for nonlinear traveltime tomography: Inverse Problems, **6**, 2142.

—, 2000a, Analysis of Approximate Inverses in Tomography I. Resolution Analysis of Common Inverses: Optimization and Engineering, **1**, 87–115.

—, 2000b, Analysis of Approximate Inverses in Tomography II. Iterative Inverses: Optimization and Engineering, **1**, 437–473.

Broyden, G.G., 1970, The Convergence of a Class of Double-Rank Minimization Algorithms: J. Inst. Math. Appl., **6**, 76–90, 222–231.

Burden, R.L., and J.D. Fares, 1989, Numerical Analysis: PWS-KENT Publishing Company.

Cardiff, M., and P.K. Kitanidis, 2008, Efficient solution of nonlinear, underdetermined inverse problems with a generalized PDE model: Computers & Geosciences, **34**, 1480– 1491.

Carpenter, W.C., and E.A. Smith, 1975, Computational Efficiency in Structural Optimization: Eng. Optimization, **1**, no. 3, 169–188.

Davidon, W.C., 1959, Variable Metric Method for Minimization: AEC Res. Develop. Rep., ANL-599.

Fletcher, R., 1970, A New Approach to Variable Metric Algorithms: Computer J., **13**, 317–322.

Fletcher, R., and M. J. D. Powell, 1963, A Rapidly Convergent Descent Method for Minimization: Computer J., **6**, 163.

Fletcher, R., and C.M. Reeves, 1964, Function minimization by conjugate gradients: Computer Journal, **7**, no. 2, 149-154.

Goldfarb, D., 1977, Generating Conjugate Directions without Line Searches Using Factorized Variable Metric Updating Formulas: Math Prog., **13**, 94–110.

Hestenes, M., and E. Stiefel, 1952, Methods of conjugate gradients for solving linear systems: NBS J. Research, **49**, 409-436.

Himmelblau, D.M., 1972, Applied Nonlinear Programming: McGraw-Hill.

Lines, L.R., and S. Treitel, 1984, Tutorial: a review of least-squares inversion and its application to geophysical problems: Geophysical Prospecting, **32**, 159-186.

Marquardt, D., 1963, An algorithm for least squares estimation of nonlinear parameters: SIAM Journal of Applied Mathematics, **11**, no. 2, 431-441.

Menke, W., 1984, Geophysical Data Analysis: Discrete Inverse Theory: Academic Press, Inc.

Nocedal, J., and S.J. Wright, 1999, Numerical Optimization: Springer-Verlag.

Powell, M.J.D., 1964, An efficient method for finding the minimum of a function of several variables without calculating derivatives: Computer Journal, **7**, no. 4, 303-307.

Rao, S.S., 1996, Engineering Optimization: Theory and Practice: John Wiley & Sons, Inc.

Ravindran, A., K.M. Ragsdell, and G.V. Reklaitis, 2006, Engineering optimization: methods and applications: John Wiley & Sons, Inc.

Sargent, R.W.H., and D.J. Sebastian, 1971, Numerical Experience with Algorithms for Unconstrained Minimization,’’ in Numerical Methods for Non-Linear Optimization: Academic.

Scales, J.A., M.L. Smith, and S. Treitel, 2001, Introductory Geophysical Inverse Theory: Samizdat Press.

Scales, J.A., and R. Snieder, 1997, To Bayes or not to Bayes?: Geophysics, **62**, no. 4, 1045–1046.

Sen, M.K., and P.L. Stoffa, 1995, Global optimization methods in geophysical Inversion: ELSEVIER.

Shanno, D.F., 1970, Conditioning of Quasi-Newton Methods for Function Minimization: Math. Comp., **24**, 647–657.

Shanno, D.F., and K.H. Phua, 1978a, Matrix Conditioning and Nonlinear Optimization: Math. Prog., **14**, no. 1, 149–160.

—, 1978b, Numerical Comparison of Several Variable-Metric Algorithms: JOTA, **25**, no. 4, 507–518.

Sivia, D.S., 1998, Data Analysis a Bayesian Tutorial: Oxford University Press Inc.

Snieder, R., 1990, A perturbative analysis of non-linear inversion: Geophys. J. Int., **101**, 545-556.

—, 1998, The role of nonlinearity in inverse problems: Inverse Problems, **14**, 387–404.

Snieder, R., and J. Trampert, 1998, Inverse Problems in Geophysics.

Tarantola, A., 2005, Inverse problem Theory and Methods for Model Estimation: Society for Industrial and Applied Mathematics.

Every reasonable effort has been made to acknowledge the owners of copyright material. I would be pleased to hear from any copyright owner who has been omitted or incorrectly acknowledged.

Chapter 3

Forward Modelling of Induced Micro-seismic Traveltime

3.1 Introduction

The main topic of this chapter is the computing of micro-seismic traveltime function in 3-dimensional heterogeneous media. Given a 3D discretized slowness distribution of a particular medium and a starting source position, I compute traveltime function at all the node points by using an efficient and robust eikonal equation solver. Further, I use a trilinear interpolation scheme to calculate the traveltimes between grid points. In chapter 4, I will use the traveltime function to build the objective function for hypocentres and origin times. I then derive algebraic formulas for tracing a ray-path starting at a receiver station to compute the ray-path segments through all the traversed cells. In chapter 5, I will use the ray-path components to construct the objective function and its gradient for the slowness model parameters.

3.2 Traveltime and Slowness

When an induced micro-seismic event occurs within a hydrocarbon reservoir, it produces hi-frequency seismic waves propagating through the 3D medium surrounding the event, before being recorded. The micro-seismic event can be thought of as a point source in both space and time despite the fact that it has a finite size. That is, the event is located in a 3D medium space at point $\{x_0, y_0, z_0\}$and initialized at time t_0. The induced seismic wave travelling through the medium depends on the medium's velocity and density distributions which can be homogeneous or heterogeneous. The heterogeneity can be one-dimensional, two-dimensional, three-dimensional or even four-dimensional, changing in time. When a particular parameter (such as velocity) varies in space and time, it demonstrates a four-dimensional heterogeneity. Such a point source generates a wavefield travelling outward governed by the wave equation.

The time it takes the seismic wave to travel from one space position to another within the medium mainly depends on the medium's velocity distribution. This time is known as traveltime. The frequency of the travelling seismic wave also affects the traveltime when the medium is dispersive: the higher the frequency, the longer the traveltime. Travelling from one position to another, the seismic wave propagates through the medium with a distinct wavespeed, affected by heterogeneity, anisotropy, dispersion and non-linearity (Liner, 2004). Such a wavespeed is known as wave velocity which represents an average speed between the two positions. However, when the speed is averaged over very short distance, it is referred to as a local velocity. The inverse of the local velocity is known as local slowness.

3.3 The Eikonal Equation

The homogeneous elastic wave equation in terms of the scalar potential field φ in a heterogeneous medium can be expressed mathematically as,

$$\nabla^2 \varphi = p^2(x,y,z)\frac{\partial^2 \varphi}{\partial t^2}. \tag{3.3.1}$$

The density is assumed constant in equation (3.3.1) as it has no effect on traveltime function and $p(x,y,z)$ represents the slowness field. Further, it is assumed that the potential field φ can be expressed as a plane wave such that,

$$\varphi(x,y,z,t) = \exp(i\omega(\psi(x,y,z)-t)). \tag{3.3.2}$$

Equation (2.3.2) can be rewritten as,

$$\varphi(x,y,z,t) = \exp(i\omega((\mathrm{Re}\,\psi(x,y,z)+i\,\mathrm{Im}\,\psi(x,y,z))-t)). \tag{3.3.3}$$

$\mathrm{Re}\,\psi(x,y,z)$ and $\mathrm{Im}\,\psi(x,y,z)$ represent the real and imaginary parts of $\psi(x,y,z)$, respectively. Multiplying and simplifying, equation (3.3.3) is expressed as,

$$\varphi(x,y,z,t) = \exp(-\omega((\mathrm{Im}\,\psi(x,y,z))).\exp(i\omega((\mathrm{Re}\,\psi(x,y,z)-t))). \tag{3.3.4}$$

The first expression in equation (3.3.4) represents the wave amplitude whereas $\mathrm{Re}\,\psi(x,y,z)$ is the eikonal, traveltime from an initial position to the point with (x,y,z)

coordinates. It follows that when $\omega \to \infty$, $\exp(-\omega((\mathrm{Im}\,\psi(x, y, z))) \to 0$. I replace $\mathrm{Im}\,\psi(x, y, z)$ by $A(x, y, z)$ and $\mathrm{Re}\,\psi(x, y, z)$ by $T(x, y, z)$ in equation (3.3.4) to obtain,

$$\varphi(x, y, z, t) = A\exp(i\omega(T(x, y, z) - t)). \qquad (3.3.5)$$

Now substituting equation (3.3.5) into equation (3.3.1), one arrives at,

$$[i\omega\nabla^2\varphi - \omega^2\nabla\varphi\cdot\nabla\varphi + \omega^2 p^2]\varphi = 0. \qquad (3.3.6)$$

Dividing equation (3.3.6) by ω^2, then the first term $\frac{i\nabla^2\varphi}{\omega} \to 0$ and the second term $\varphi \to T$ as $\omega \to \infty$. Consequently, I arrive at the eikonal equation,

$$\left(\frac{\partial T}{\partial x}\right)^2 + \left(\frac{\partial T}{\partial y}\right)^2 + \left(\frac{\partial T}{\partial z}\right)^2 = p^2. \qquad (3.3.7)$$

The eikonal equation (Chapman, 2004; Robinson and Clark, 2003) relates the traveltime $T(x, y, z)$ function and the slowness field $p(x, y, z)$ in a 3D medium. The slowness function is positive. Hence, the surface evolution always moves outward from the initial position. A surface where T is constant is called wavefront (level set) and it is perpendicular to the slowness field. Equation (3.3.7) allows for different wave types, direct, reflected, diffracted, head waves. When the slowness function is given, then the eikonal equation is used to compute the traveltimes function within the 3D medium.

3.4 The Ray Equation

Consider a local region within the 3D medium such that the slowness function is constant and define two points s_1 and s_2 as shown in Figure (3.1). The ray that travels from the point s_1 to s_2 is linear. Based on Figure (3.1), the path that the ray travelled is expressed mathematically as,

$$\vec{r} = \vec{r}_0 + s\vec{u}. \qquad (3.4.1)$$

The vector $\vec{u}$ represents a unit vector in the direction of the ray. Differentiating Equation (3.4.1) with respect to s (the path length) and rearranging yields:

$$\vec{u} = \frac{d}{ds}(\vec{r}). \qquad (3.4.2)$$

The eikonal equation can be expressed in vector form as,

$$(\nabla T)\cdot(\nabla T) = (p\vec{u})\cdot(p\vec{u}). \qquad (3.4.3)$$

Thus, the gradient of the eikonal function is in the direction $\vec{u}$ and with a magnitude p. In other words,

$$\nabla T = p\frac{d}{ds}(\vec{r}). \qquad (3.4.4)$$

Taking the gradient of the eikonal equation (3.3.7), one obtains,

$$\nabla\left(\nabla T\right)^2 = 2(\nabla T\cdot\nabla)\nabla T = 2p(\nabla p). \qquad (3.4.5)$$

Using equation (3.4.4), the left side of equation (3.4.5) is written as,

$$(\nabla T\cdot\nabla)\nabla T = (p\frac{d}{ds}(\vec{r})\cdot\nabla)\nabla T. \qquad (3.4.6)$$

Using the chain rule and equation (3.4.4), equation (3.4.6) can be expressed as,

$$(p\frac{d}{ds}(\vec{r})\cdot\nabla)\nabla T = p\frac{d}{ds}\left(\nabla T\right) = p\frac{d}{ds}\left(p\frac{d}{ds}(\vec{r})\right). \qquad (3.4.7)$$

Combining equations (3.4.5) and (3.4.7), one arrives at the ray equation,

$$\frac{d}{ds}\left(p\frac{d}{ds}(\vec{r})\right) = \nabla p. \qquad (3.4.8)$$

The ray equation (Cerveny, 2001; Chapman, 2004) relates the ray-path s and the slowness field $p(x,y,z)$ in a medium. When the slowness function is given, the ray equation is used to calculate traveltimes for a particular wave (direct, reflected, refracted, converted, or multiply reflected) along the ray-path of such wave.

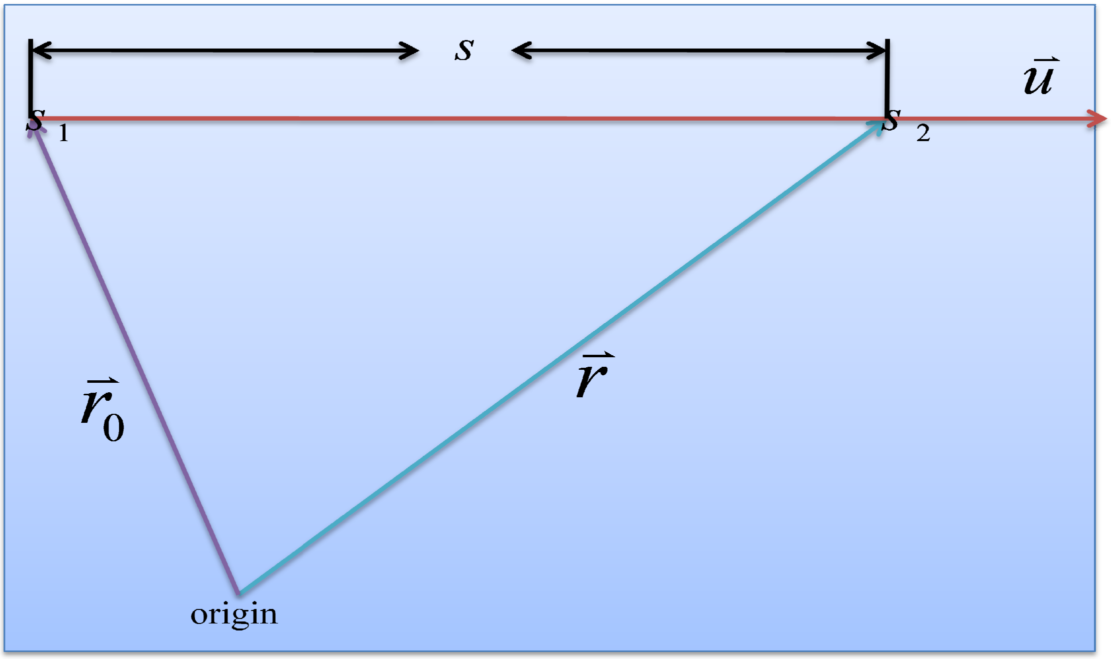


Figure (3.1): Diagram showing the relationship between r and s.

3.5 Ray-path and Fermat's Principle

The traveltime in the above situation between s_1 and s_2, the slowness field, and the ray-path connecting the two points are related through,

$$dt = p(x, y, z)dl, \tag{3.5.1}$$

where dt represents the traveltime and dl is an infinitesimal segment of the ray-path. When the above two points are no longer close to each other, the linearity assumption might not hold. Then, the quantities in equation (3.5.1) are related through an integral of slowness function along a ray-path connecting the two points,

$$T(s_2) - T(s_1) = \int_{\Gamma} p(x, y, z)dl, \tag{3.5.2}$$

such that Γ represents a ray-path from s_1 to s_2. But, there are infinitely many ray-paths starting at s_1 and terminating at s_2 that one can choose from. Furthermore, the traveltime depends on the chosen ray-path. To overcome such a problem, the Fermat's principle – the right ray-path is the one that allows the wavefront to move from s_1 to s_2 in the shortest amount of time – is used. Using equation (3.5.2) and Fermat's principle, the correct traveltime at s_2 is expressed as,

$$T(s_2) = T(s_1) + \int_{\Gamma^*} p(x,y,z)dl. \tag{3.5.3}$$

The ray-path Γ^* denotes the path that goes, in the least time, from s_1 to s_2. Then, there are two approaches to the correct ray-path. The first one is to calculate the minimum traveltime using the eikonal equation and later obtain the correct ray-path. The second approach, on the other hand, is to compute the right ray-path initially by using the ray equation, and then find the traveltime.

3.6 Snell's Law

Now consider the same two points s_1 and s_2 are separated by a plane such that the two formed media are homogenous, but with different isotropic slownesses as shown in Figure (3.2). The first point is in the lower medium while the second one is in the upper medium. It is clear that there are infinitely many straight ray-paths that can be used to connect the two points. The traveltime between the two points is expressed as,

$$T(s_1, s_2) = p_1 l_1 + p_2 l_2, \tag{3.6.1}$$

where l_1 represents ray-path length between s_1 and the point where the ray intersects the plane and l_2 denotes the ray-path length between the point of intersection and s_2. Using trigonometry, equation (3.6.1) is written as,

$$T(d_1) = p_1\sqrt{d_1^2 + h_1^2} + p_2\sqrt{(d - d_1)^2 + h_2^2}. \tag{3.6.2}$$

Applying Fermat's principle to equation (3.6.2) provides a one-dimensional optimization problem,

$$\frac{\partial T(d_1)}{\partial d_1} = 0. \tag{3.6.3}$$

This leads to the well-known Snell's law:

$$p_1 \sin\theta_1 = p_2 \sin\theta_2. \tag{3.6.4}$$

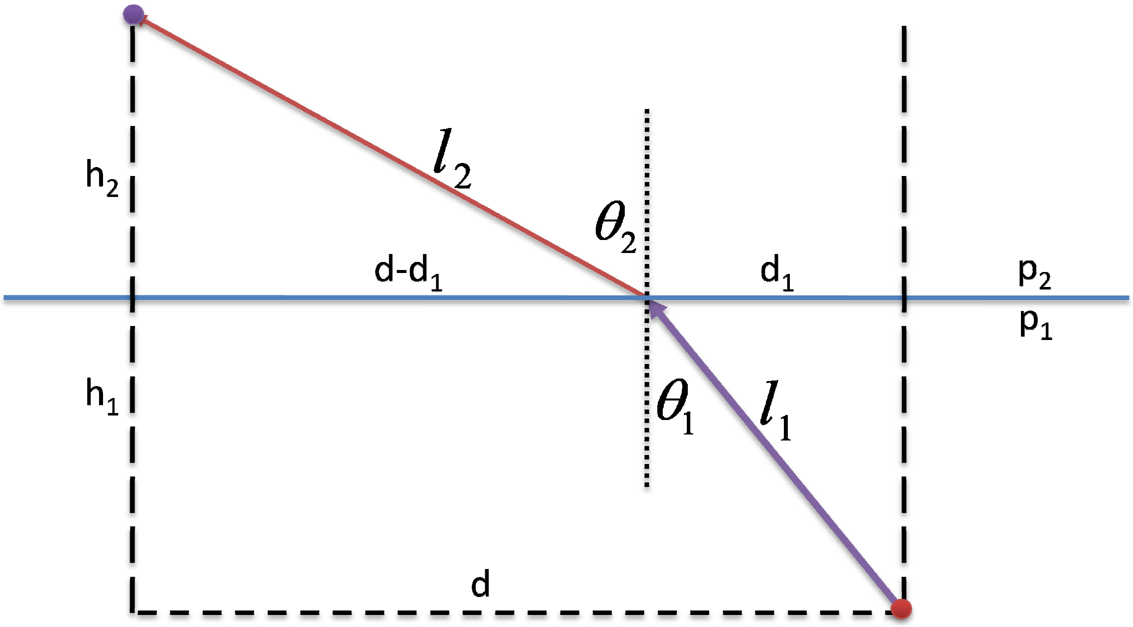


Figure (3.2): A ray starting at the red circle (s_1) and intersecting with the plane separating the two media. The ray refracts in the upper medium according to its slowness before ending at the purple circle (s_2).

3.7 Traveltimes Computation

I use the Eikonal equation as a mathematical model and the given discretized slowness field for either P-wave or S-wave as a set of model parameters to produce traveltimes at all the node points. In this setting, it is obvious that there is a clear distinction between the mathematical model (eikonal equation), the model parameters (discretized slowness field), and the associated model response (traveltimes). While the setting can be used for either 2D medium or 3D medium, I use it for a 3D medium. The method involves solving the eikonal equation starting at either a source or a receiver position. Therefore, the objective is to develop an efficient and robust algorithm to solve the eikonal equation.

3.7.1 Background

In order to be able to use the observed micro-seismic arrival times to determine the event position, the event origin time, the ray-path, and the slowness model parameters, it is essential to have an efficient method to compute synthetic traveltimes of the evolving wavefront as it passes each point in space. The two most frequently used methods to generate traveltimes are ray-tracing and numerical solutions to the Eikonal Equation. On the one hand, the ray-tracing methods are commonly categorized into either ray shooting or ray bending (Julian and Gubbins, 1977). Although the ray-tracing methods can attain very accurate traveltimes (Gazdag, 2000; Lafond and Levander, 1990), they require intensive computations; suffer from convergence problems; and most of the times fail in the shadow zone (Cao and Greenhalgh, 1994; Moser, 1991; Sava and Fomel, 2001; Schneider et al., 1992; Vidale, 1990). On the other hand, Eikonal Equation solvers are classified into either the expanding box strategy or the expanding wavefront scheme (Kim and Cook, 1999). While Eikonal Equation solvers are fast and robust (Kim, 2002; Kim and Cook, 1999; Qin et al., 1992; Sethian and Popovici, 1999; Vidale, 1990), they are limited to computing only the first arrival traveltimes and lack the control of the propagating angle (Sava and Fomel, 2001). Still, there are less common methods such as those based on either dynamic programming (Moser, 1991) or wavefront construction (Sava and Fomel, 2001).

Since the problem on hand requires a fast and robust method that generates only the first arrival traveltimes, it is natural to choose the finite difference Eikonal solvers. Furthermore, the expanding wavefront schemes require more computations than those of the expanding cube, which can be readily modified to attain higher accuracy (Kim and Cook, 1999). In addition, the expanding cube technique requires neither the velocity model parameters to be smooth nor the knowledge of the geologic interfaces (Schneider et al., 1992). Therefore, I decided to adhere to the expanding cube strategy for generating synthetic traveltimes in my settings.

3.7.2 Model Discretization

The 3D medium is discretized into regular grids resulting in regular cubes and assigned with slowness model parameters at the corresponding grid points. So, a node point is located at the corner of a cube. The discretization process is performed in accordance with the Cartesian coordinate system where positive z-direction points downward, positive y-direction points backward, and positive x-direction points to the right. Furthermore, if interpolation is required, a trilinear interpolator is used to specify the slowness parameters between grid points. Although I use an even discretization, it is very straightforward to adjust to other settings.

The origin of the 3D model is specified as $\{x_0, y_0, z_0\}$and can be either positive or negative real numbers. Also one needs to specify the node spacing $h = \Delta x = \Delta y = \Delta z$ and the number of nodes in each direction. N_x , N_y and N_z are the number of nodes for x-direction, y-direction, and z-direction, respectively. Traveltime at node $\{i, j, k\}$ is denoted as $T(i, j, k) = T(i\Delta x, j\Delta y, k\Delta z)$, $i = 0,1,...,N_x$, $j = 0,1,...,N_y$, $\mathrm{k} = 0,1,...,N_z$

whereas the slowness at the same node is denoted as $p(i,j,k) = p(i\,\Delta x, j\,\Delta y, k\,\Delta z)$. Such discretization leads to $(N_x - 1)(N_y - 1)(N_z - 1)$ cubes. Each cube has eight nodes at its corners with homogeneous isotropic slowness, and it is referenced by its upper-front-left corner. This referencing system is very crucial to track the ray-path length spent within each traversed cube.

The slowness field is a continuous function of position. Consequently, the discretization process transforms the set of model parameters from an infinite set (the continuous slowness function) to a finite one (discrete slowness function). This in turn would undoubtedly show in the model response as having far less details than the real one. On the other hand, one can improve on the model response by reducing the cube size and hence increasing the number of grid points. However, this would lead to an increase in the computational cost. Therefore, one needs to balance between the two.

3.7.3 Traveltime Initialization

All the node points are populated with their corresponding slowness parameters. Further, the traveltime function is set to a very large value at all nodes. The source location can be anywhere within the 3D medium either on or off a grid point. When the source point is not on a grid point, the nearest grid point to the source position is determined and the time at that node is computed and updated. Otherwise, the time where the source is located is set to zero. The traveltimes to all the node points within an initial 3x3x3 box surrounding the source grid point at the centre are computed based on linear velocity. Figure (3.3) shows the grid geometry for the initial box. The left column of Figure (3.3) shows the three layers of the cube, where the first layer represents the upper horizontal slice, the second layer stands for the middle horizontal slice, and the third one is for the lower horizontal slice of the cube. The source position is in the centre of the middle layer and labelled as a red circle. The orange circles represent the nodes with known times. The nodes whose times are still not known are marked with purple circles. The remaining six layers in Figure (3.3) represent the different faces of the cube as labelled. To advance to next stage (a 5x5x5 box), only those nodes setting on a face with known times are used.

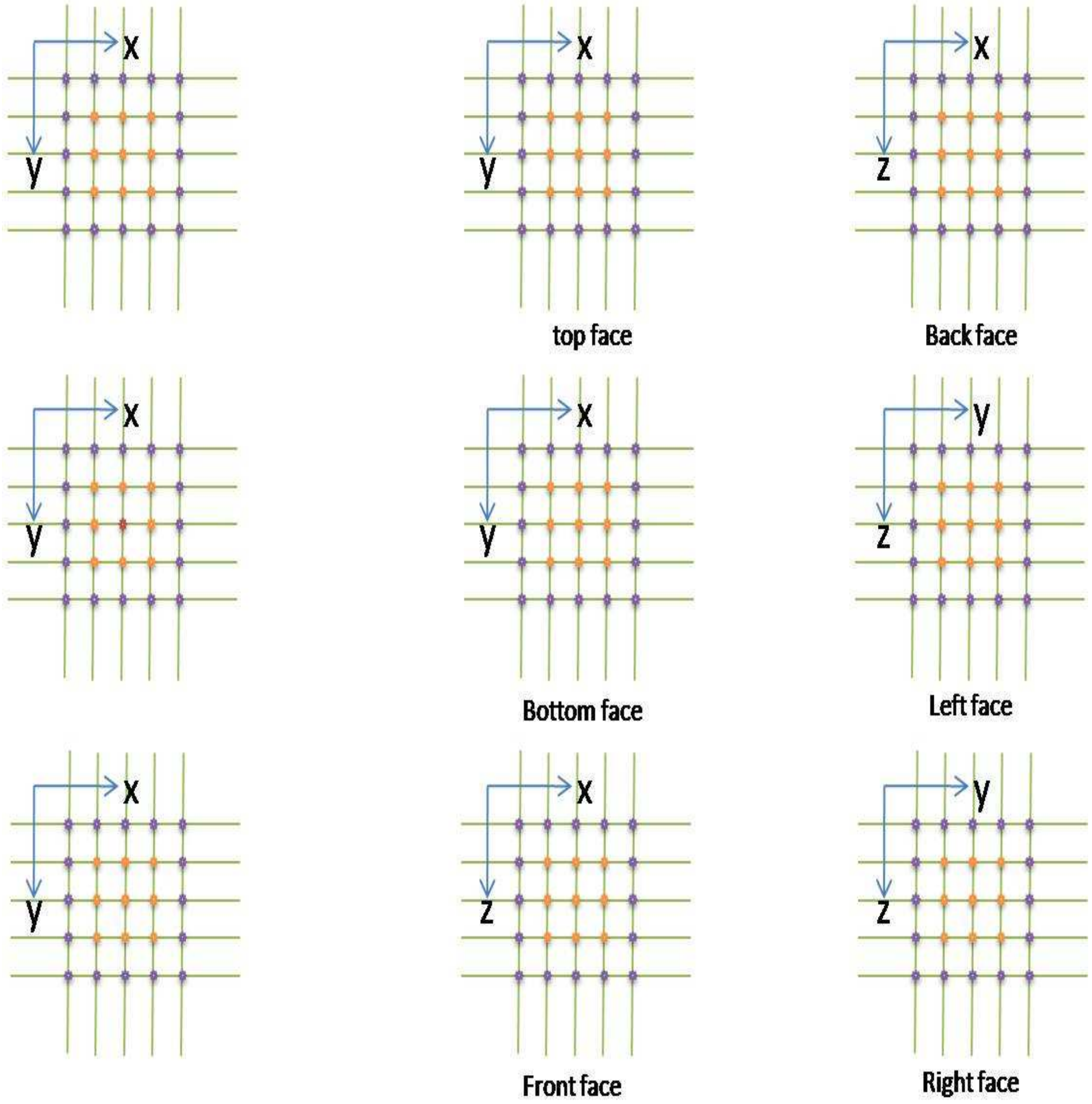


Figure (3.3): Grid geometry for the initial box with the source being labelled as red circle. Orange circles represents timed nodes whereas purple nodes stands for untimed ones. Also, the six faces of the initial timed box are shown with their corresponding timed nodes.

3.7.4 Traveltime Computation Algorithm

To construct the next level traveltimes, the algorithm uses three implicit schemes based on the 3D eikonal equation (Vidale, 1990). The first scheme is used to calculate the traveltime at the eighth corner of a cube having known traveltimes at all the other corners as shown in Figure (3.4). It uses four known traveltimes from the old face and three known traveltimes from the new face,

$$t_7 = t_0 + \frac{1}{\sqrt{2}}\sqrt{6h^2\bar{p}^2 - (t_1 - t_2)^2 - (t_2 - t_4)^2 - (t_4 - t_1)^2 - (t_3 - t_5)^2 - (t_5 - t_6)^2 - (t_6 - t_3)^2}. \tag{3.7.1}$$

The slowness is represented as $\bar{p}$ and it is taken as the average of all the eight slownesses while h denotes the grid spacing size. The second scheme is used to compute the traveltime on a new edge as shown in Figure (3.5). Such a scheme uses four nodes with known traveltimes from the old face and one timed node from the new face,

$$t_5 = t_1 + \sqrt{2h^2\bar{p}^2 - \frac{1}{2}(t_0 - t_3)^2 - (t_2 - t_4)^2}. \tag{3.7.2}$$

The third one is used to construct the traveltime on a new face as shown in Figure (3.6). This scheme uses five known traveltimes from the old face,

$$t_5 = t_2 + \sqrt{h^2\bar{p}^2 - \frac{1}{4}[(t_1 - t_3)^2 + (t_0 - t_4)^2]}. \tag{3.7.3}$$

The above schemes would violate the causality or produce a negative number under the square root when the velocity contrast is more than doubled (Vidale, 1990). A solution to overcome such a problem and other improvements to the original Vidale’s algorithm are presented in (Hole and Zelt, 1995) and used in this algorithm. The remedy is achieved through the use of the following 1-D and 2-D operators to account for head wave travelling along the boundary between the two media with the faster velocity,

$$t_1 = t_0 + \bar{p}h, \tag{3.7.4}$$

$$t_3 = t_0 + \sqrt{2h^2\bar{p}^2 - (t_2 - t_1)^2}. \tag{3.7.5}$$

The algorithm consists of six stages. Each stage is concerned with a specific face of the box as shown in Figure (3.7). Further, each stage has a quick sort and 26 operators (eight 3-D operators, twelve 2-D operators, and six 1-D operators) (Podvin and Lecomte, 1991). Such operators are used to check for all possible wave propagation to the attempted node and hence obtain the least traveltime. Furthermore, they are used to detect head wave at the attempted node.

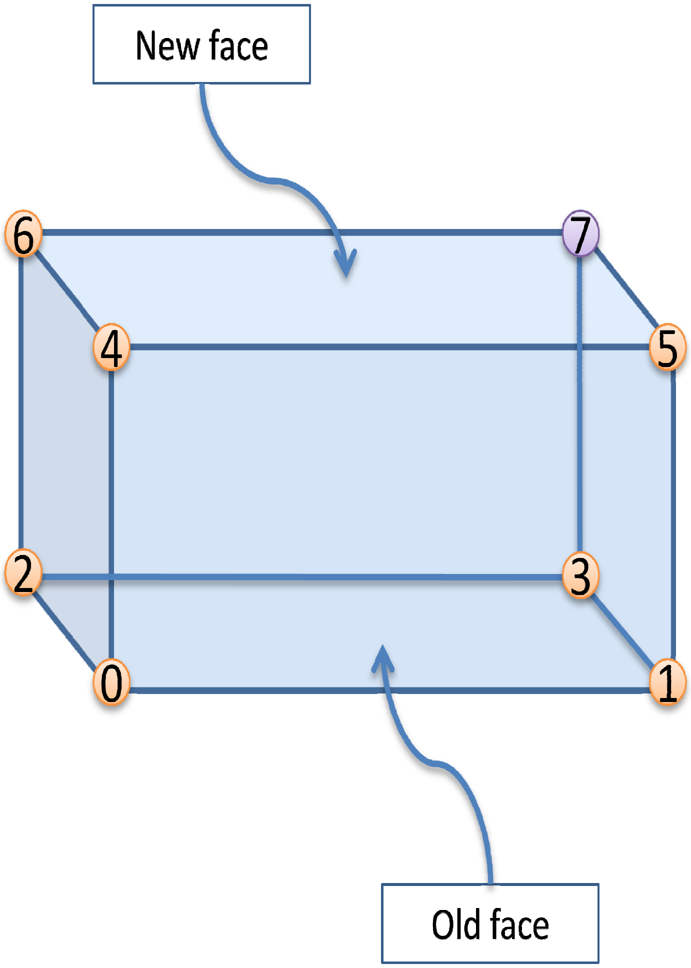


Figure (3.4): 3D transmission stencil using four timed nodes on the old face (marked as orange sphere 0,1, 2, and 3) and three timed nodes on the new face (labelled as orange sphere 4, 5, and 6) to compute the traveltime at node 7 marked with the purple sphere. The stencil uses an average of the slownesses at the eight nodes.

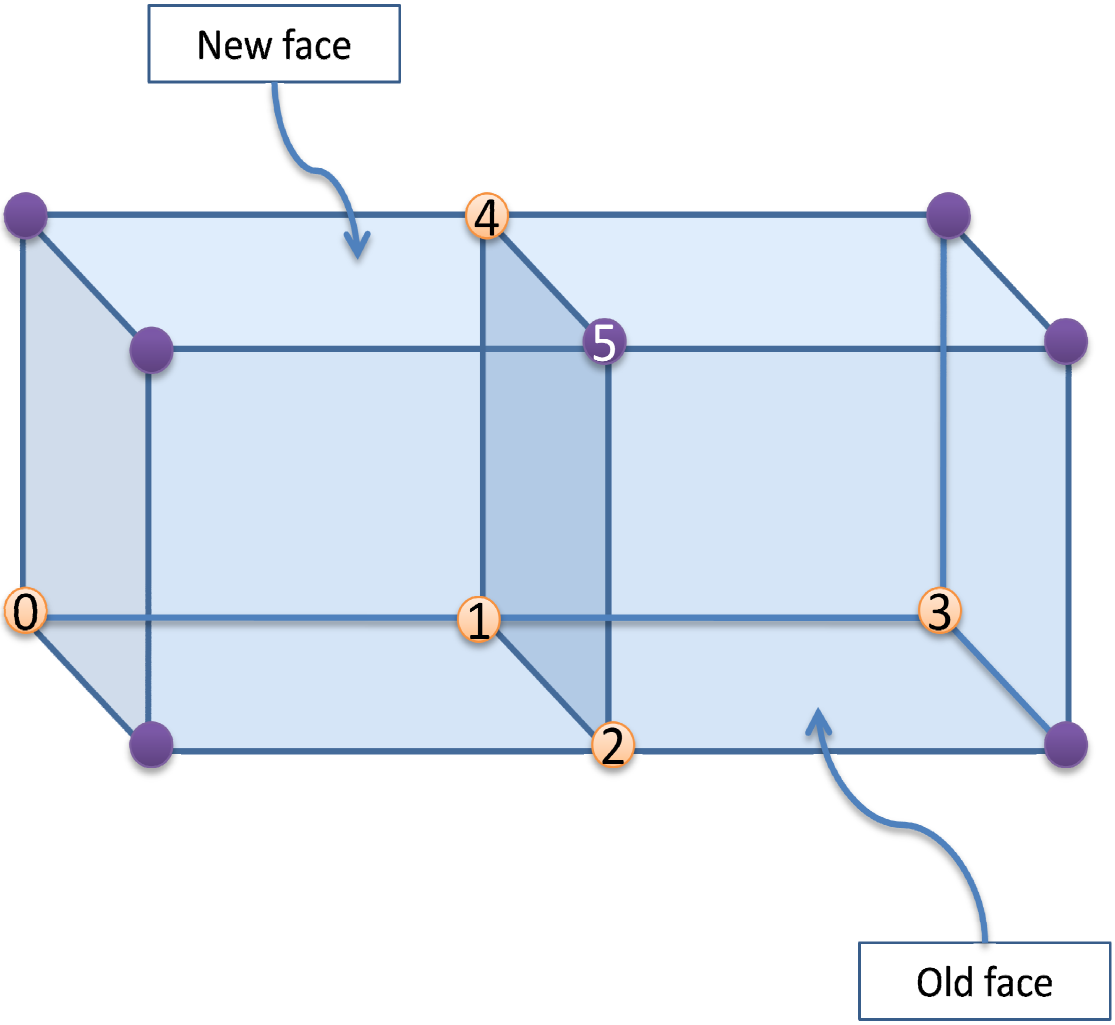


Figure (3.5): 3D transmission stencil using four timed nodes on the old face (marked as orange sphere 0,1, 2, and 3) and one timed nodes on the new face (labelled as orange sphere 4) to compute the traveltime at node 5 marked with purple sphere. The stencil uses an average of the slownesses at the four nodes, 1, 2, 4, and 5.

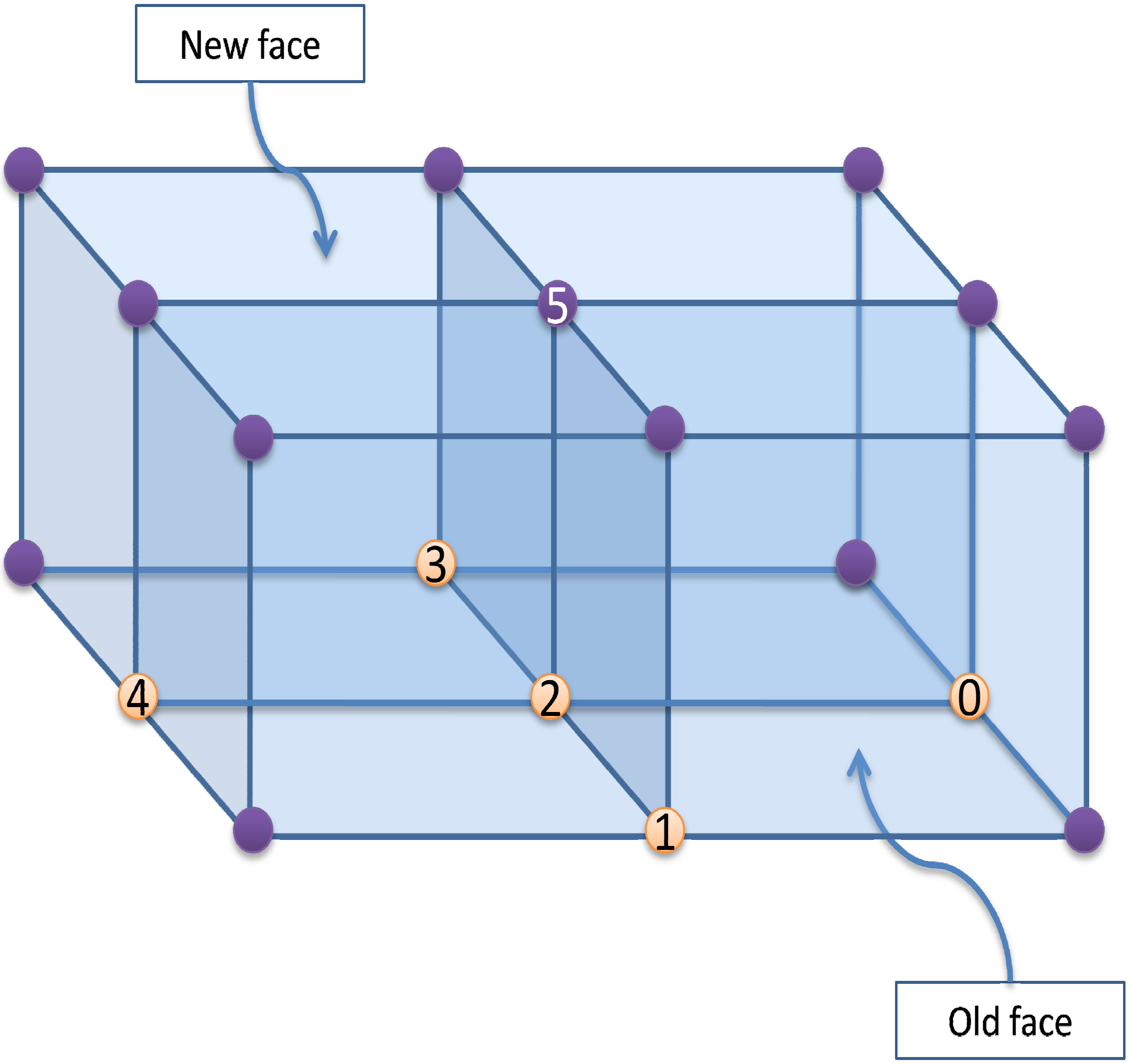


Figure (3.6): 3D transmission stencil using five timed nodes on the old face (marked as orange sphere 0, 1, 2, 3, and 4) to compute the traveltime at node 5 marked with purple sphere. The stencil uses an average of the slownesses at the two nodes, 2, and 5.

The first stage starts with the timed top face (3x3 square) as shown in Figure (3.7). This face is named as the old face. The nine known traveltimes within the top face are sorted from smallest to largest traveltime. Such known traveltimes are used to construct traveltimes on the adjacent face, denoted as the new face which is one level above the old face in the negative z-direction. The traveltime on the adjacent node on the new face to the node with smallest traveltime on the old face is attempted first through the twenty-six operators. The node is timed with the least traveltime and marked with an orange sphere. The timing process is repeated for all the remaining nodes on the new face that are adjacent to timed nodes on the old face. The result is a 3x3 timed square as shown in Figure (3.7).

The second starts with the timed bottom face (3x3 square) as shown in Figure (3.7). This face is named as the old face. The nine known traveltimes within the old face are sorted from smallest to largest traveltime. The new face is one level below the old face in the positive z-direction. The traveltime on the adjacent node on the new face to the node with smallest traveltime on the old face is attempted through the twenty-six operators. The node is timed with the least traveltime and marked with an orange sphere. The timing process is repeated for all the remaining nodes on the new face that are adjacent to timed nodes on the old face. The result is a 3x3 timed square.

The third stage starts with the timed front face (a 3x5 rectangle) as shown in Figure (3.7). This face is denoted as the old face. The fifteen known traveltimes within the old face are sorted from smallest to largest traveltime. The new face is one level to the front from the old face in the negative y-direction. The traveltime on the adjacent node on the new face to the node with smallest traveltime on the old face is attempted through the twenty-six operators. The node is timed with the least traveltime and marked with an orange sphere. The timing process is repeated for all the remaining nodes on the new face that are adjacent to timed nodes on the old face. The result is a 3x5 timed rectangle.

The fourth stage starts with the timed back face (a 3x5 rectangle) as shown in Figure (3.7). This face is labelled as the old face. The fifteen known traveltimes within the old face are sorted from smallest to largest traveltime. The new face is one level to the back from the old face in the positive y-direction. The traveltime on the adjacent node on the new face to the node with smallest traveltime on the old face is attempted through the twenty-six operators. The node is timed with the least traveltime and marked with an orange sphere. The timing process is repeated for all the remaining nodes on the new face that are adjacent to timed nodes on the old face. The result is a 3x5 timed rectangle.

The fifth stage starts with the timed left face (a 5x5 square) as shown in Figure (3.7). This face is denoted as the old face. The twenty-five known traveltimes within the old face are sorted from smallest to largest traveltime. The new face is one level to the left from the old face in the negative x-direction. The traveltime on the adjacent node on the new face to the node with smallest traveltime on the old face is attempted through the twenty-six operators. The node is timed with the least traveltime and marked with an orange sphere. The timing process is repeated for all the remaining nodes on the new face that are adjacent to timed nodes on the old face. The result is a 5x5 timed square.

The sixth stage starts with the timed right face (a 5x5 square) as shown in Figure (3.7). This face is labelled as the old face. The twenty-five known traveltimes within the old face are sorted from smallest to largest traveltime. The new face is one level to the right from the old face in the positive x-direction. The traveltime on the adjacent node on the new face to the node with smallest traveltime on the old face is attempted through the twenty-six operators. The node is timed with the least traveltime and marked with an orange sphere. The timing process is repeated for all the remaining nodes on the new face that are adjacent to timed nodes on the old face. The result is a 5x5 timed square.

When the six stages are completed, a new 5x5x5 cube is timed. The algorithm reiterates the six stages to produce a 7x7x7 cube. The process is continued in the same manner till all the nodes within the 3D medium are timed. The algorithm ensures that the first arrival traveltime is accurately obtained for each node point. The accuracy is related to the grid spacing. The smaller the grid size is, the more accurate the traveltimes.

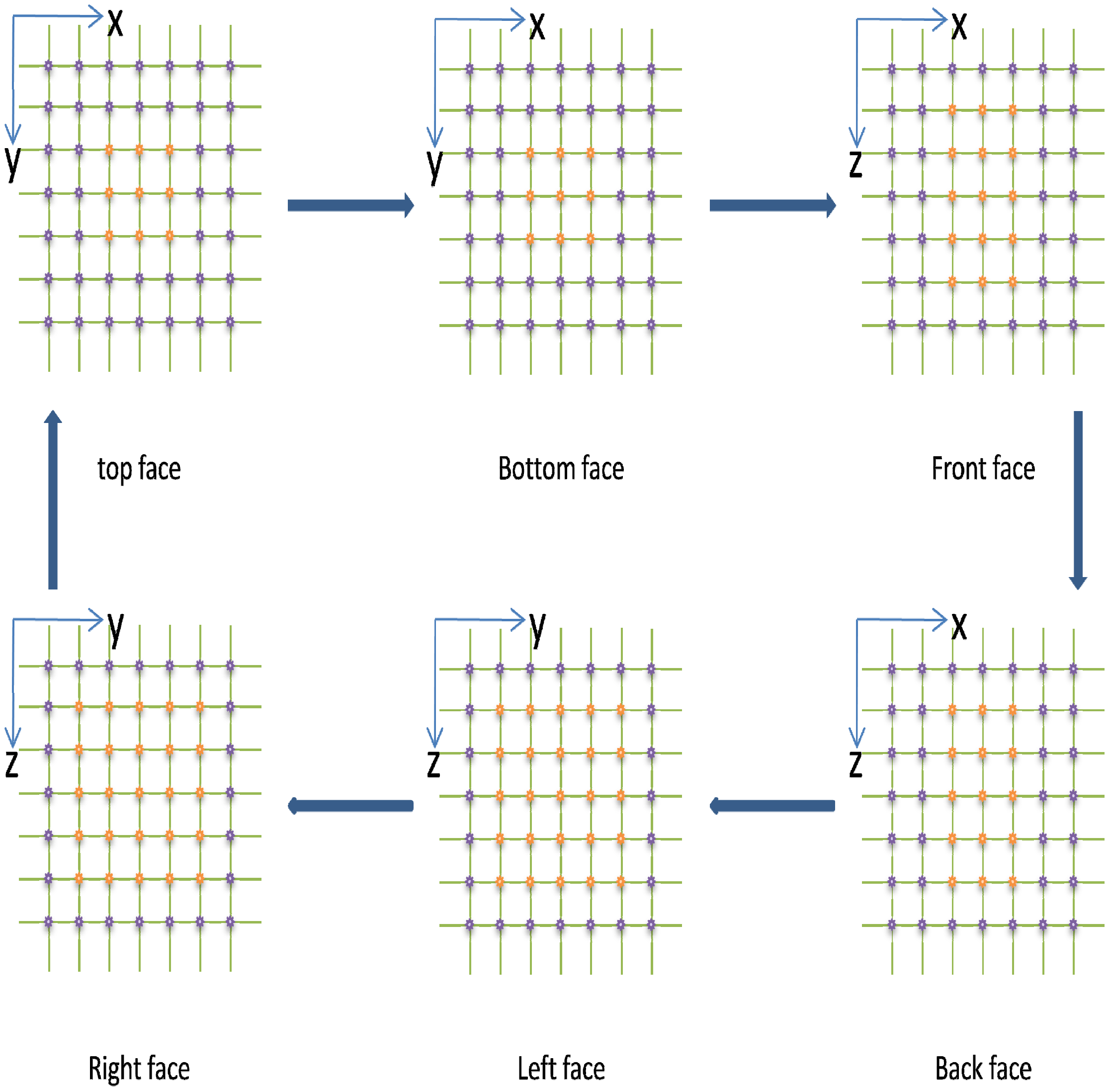


Figure (3.7): Procedure for timing bigger cube from the current timed cube, starting at the top face and finishing at the right face. The algorithm starts with a 3x3x3 timed cube and ends up with a 5x5x5 timed cube.

3.7.5 Numerical Results

I tested the algorithm on different models for computing traveltimes. Three of these are discussed here. In the first test, I used a constant velocity of 3700 m/s and two grid dimensions of the velocity model were $N_x = N_y = N_z = 201$and 101. The grid increments were set as $h = 2.5$and 5.0 m, respectively. The size of the resulting cube was 500 m on each side. The source was set to the position$\{75, 15, 380\}$. The reason for this choice was to compare calculated traveltimes with exact ones to provide an objective measure of accuracy of the algorithm. Figure (3.8) shows the geometry of the source and 121 receivers on the surface.

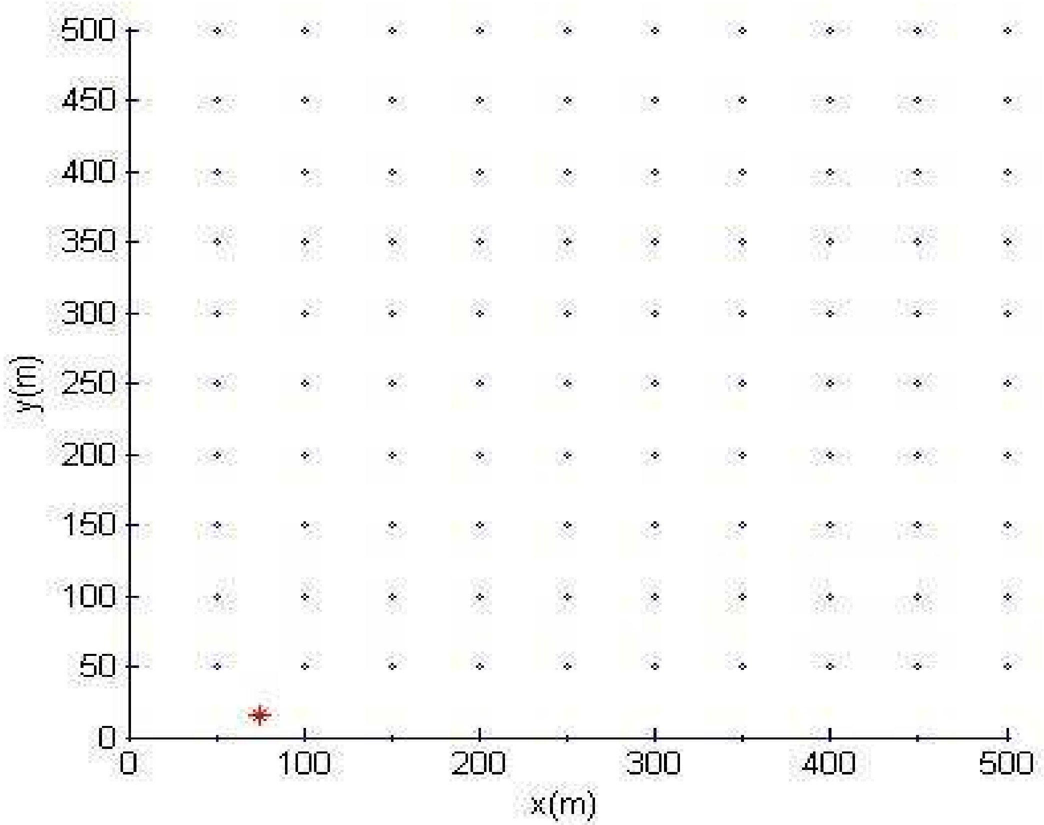


Figure (3.8): Plan view at the z=0 plane showing receivers labelled as blue diamonds. The projection of the source is marked as red asterisk. Receiver stations are situated on the top plane, z=0.

I used the algorithm to calculate traveltimes at the receiver stations for the homogenous isotropic velocity model. Figure (3.9) shows two diagrams; the upper one shows the traveltime contours and the lower diagram displays the projected traveltime surface for the 5m grid spacing. Figure (3.10) also shows two graphs; the upper one displays the absolute errors between the exact traveltimes and those computed using 5.0m grid spacing while the lower one shows the errors when grid spacing is 2.5m. It is clear that when the grid spacing was halved (reduced from 5m to 2.5m), the traveltime errors was also reduced by a factor of two. This indicates that the relationship between the traveltimes accuracy and grid spacing is linear. The program took nearly 2 seconds to compute traveltimes when grid spacing was 5.0m and approximately 17 seconds when grid spacing was 2.5m.

In the second test, I used a linear gradient velocity changing along the z-direction according to the following formula,

$$v(x, y, z) = a + bz \ \text{ m/s}; \quad a = 3000,\ b = 2.5. \tag{3.7.6}$$

I also used the same discretization as described above. Such a formula has an exact solution given as (Cerveny, 2001),

$$T\ (s\,, r\,) = \left| b^{-1} \cos h^{-1} [1 + (p_r\, p_s d^{\,2} b^{\,2}\ /\ 2)] \right|. \tag{3.7.7}$$

In equation (3.7.7), $T\ (s\,, r\,)$ represents the traveltime between s and r while p_s and p_r stand for the slowness model parameters at s and r, respectively. The distance between s and r is denoted as d. Figure (3.11) and (3.12) show traveltime contours and traveltime errors for 2.5m grid spacing and 5.0m grid spacing, respectively. As for the homogenous case, the relationship between grid spacing and traveltime accuracy is linear. Further, the errors shapes are almost the same as they depend only on grid spacing.

In the third test, I used a 3D layer cake model, which I use as a bench mark for later chapters. This model consists of six different layers with a velocity inversion between layer two and three as shown in Table (3.1). The same geometry configuration is used as in example one and two. The computed traveltime contours and errors for such a model are shown in Figure (3.13). Figure (3.14) has two plots showing differences between the ray tracing values and those computed by my algorithm for both 5.0m and 2.5m. To test the reciprocity principle, I used the same model to compute traveltimes starting at the receiver stations position. Figure (3.15) shows the traveltime errors between the forward and backward traveltimes. The difference between the forward and backward traveltimes is consistent to suggest the validity of the method.

Although the test examples were designed for relatively small models, the results show that the algorithm works very well in different settings with excellent accuracy. Such accuracy is essential as the main work of this thesis, gradient-based optimisation, involves derivatives of the traveltime function. Equally important is the validity of the reciprocity principle which is a fundamental tool that I use in this thesis for locating micro-seismic events. I think that the algorithm can produce accurate traveltimes for various 3D imaging problems.

Layer	Velocity (m/s)	Thickness (m)
1	3500	100
2	3650	100
3	3400	25
4	3700	75
5	3800	100
6	4000	100

Table (3.1): Layer cake model.

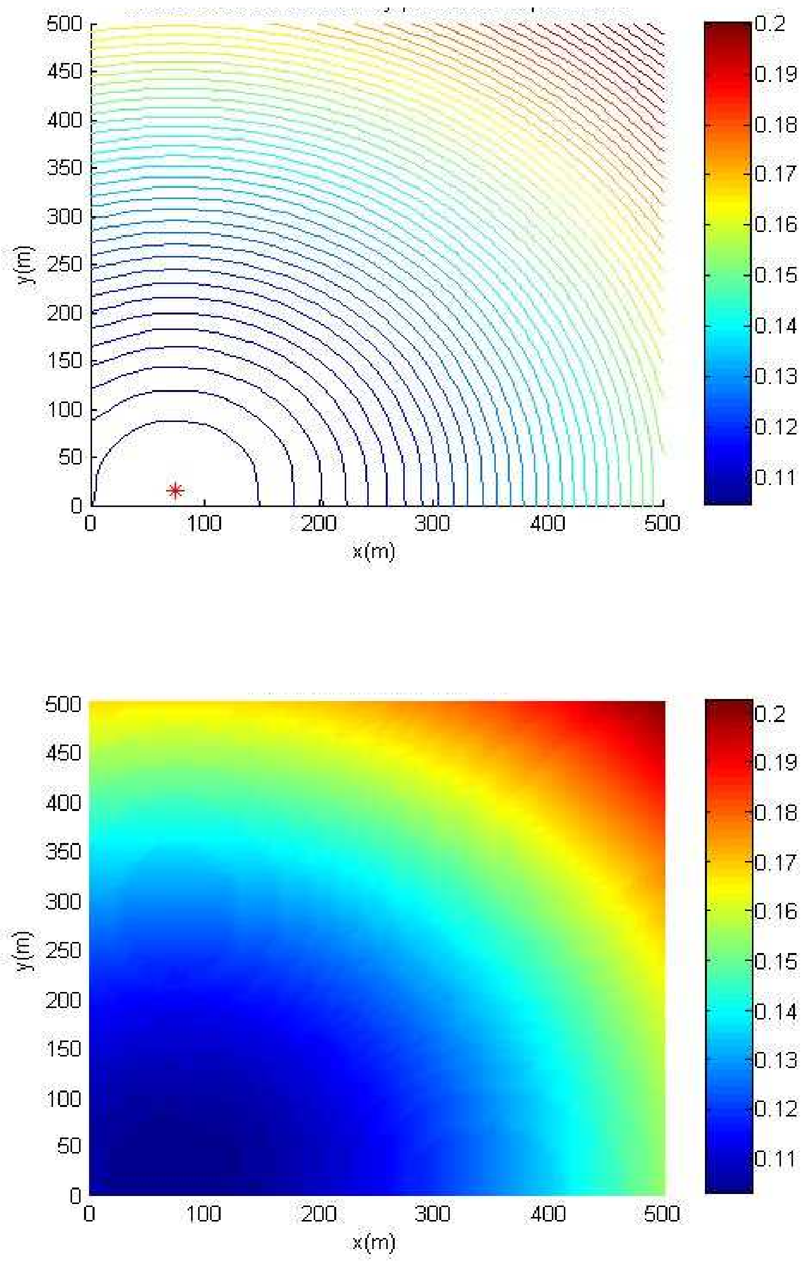


Figure (3.9): Traveltimes in seconds at the receiver stations computed for a homogeneous velocity model with a grid spacing of 5.0 meters in two different views.

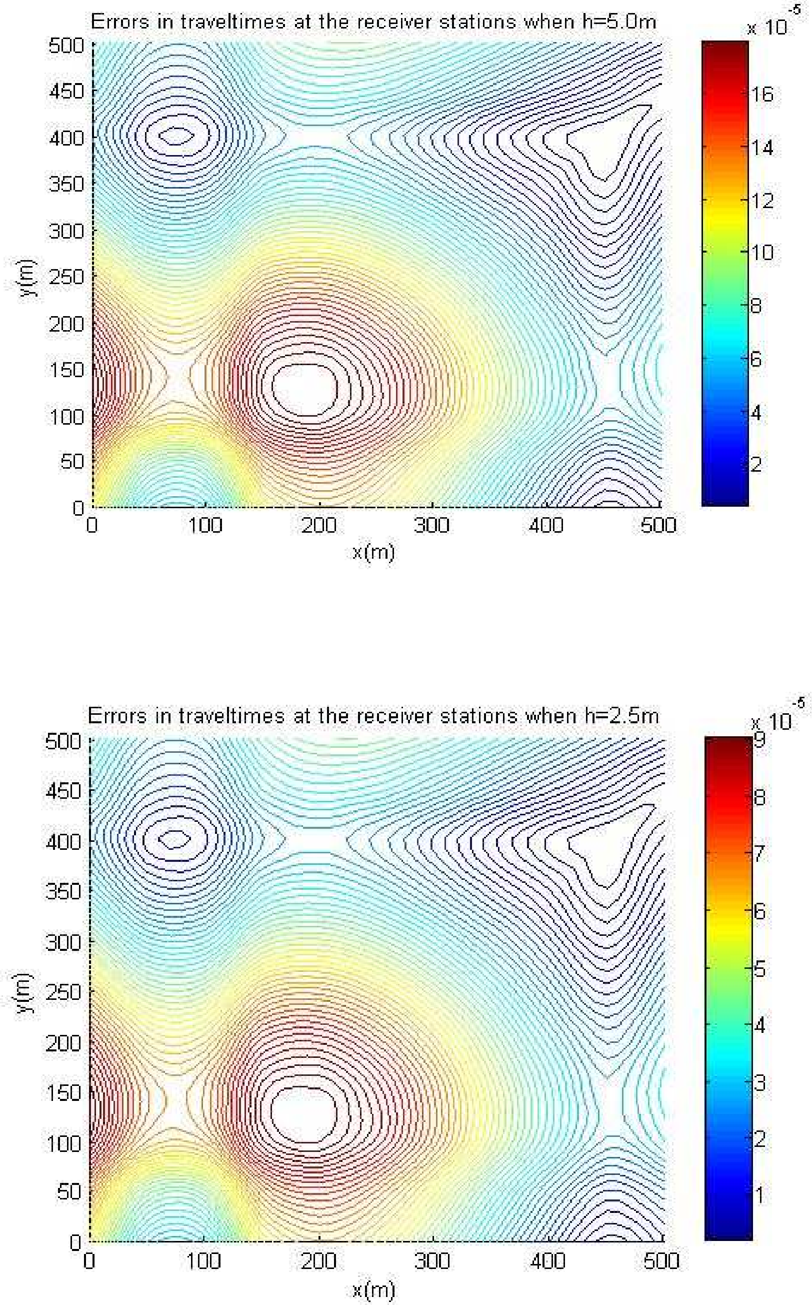


Figure (3.10): Traveltime errors in seconds at the receiver stations for the homogeneous velocity model with two different grid spacing. The colour bar indicates the accuracy for the two different grid spacing, top 5.0m and bottom 2.5m.

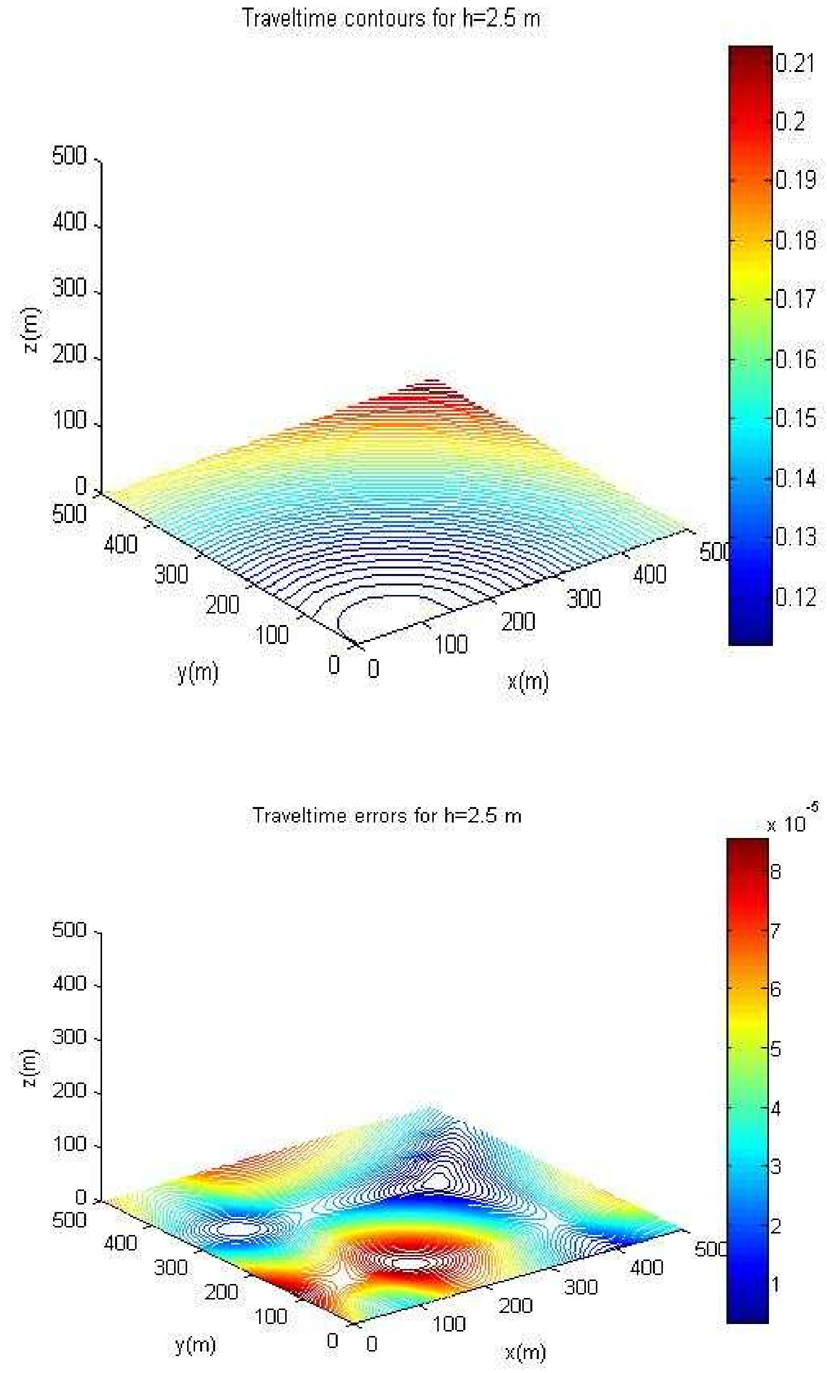


Figure (3.11): Traveltime contours and errors in seconds at the receiver stations computed for the linear gradient velocity model with a grid spacing of 2.5m.

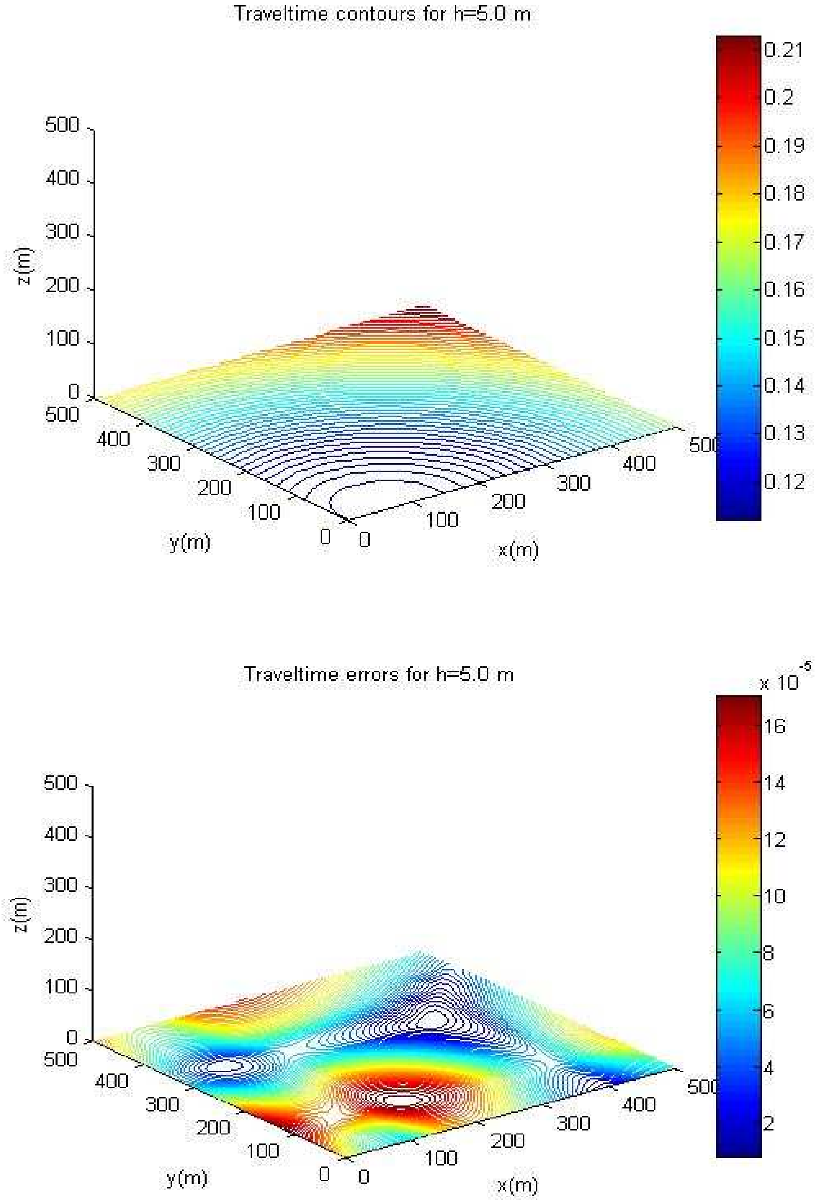
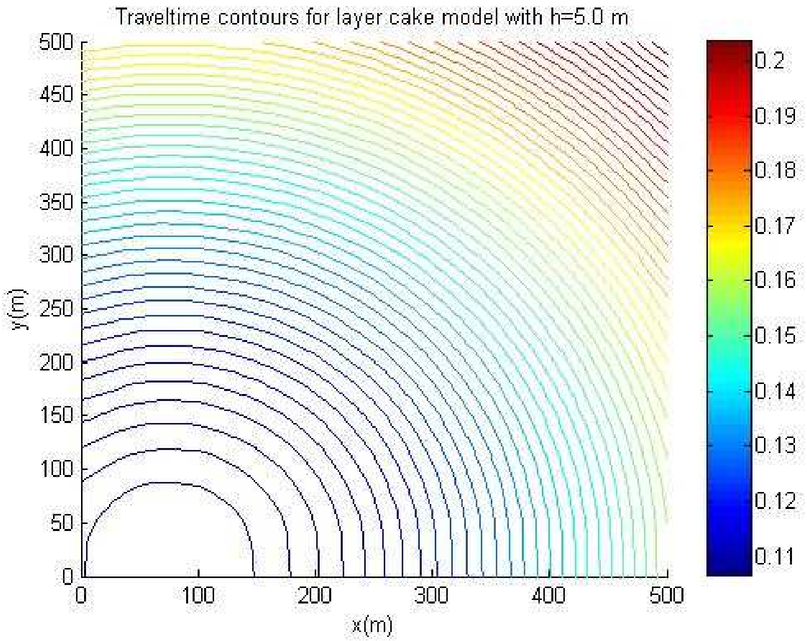


Figure (3.12): Traveltime contours and errors in seconds at the receiver stations computed for the linear gradient velocity model with a grid spacing of 5.0m.



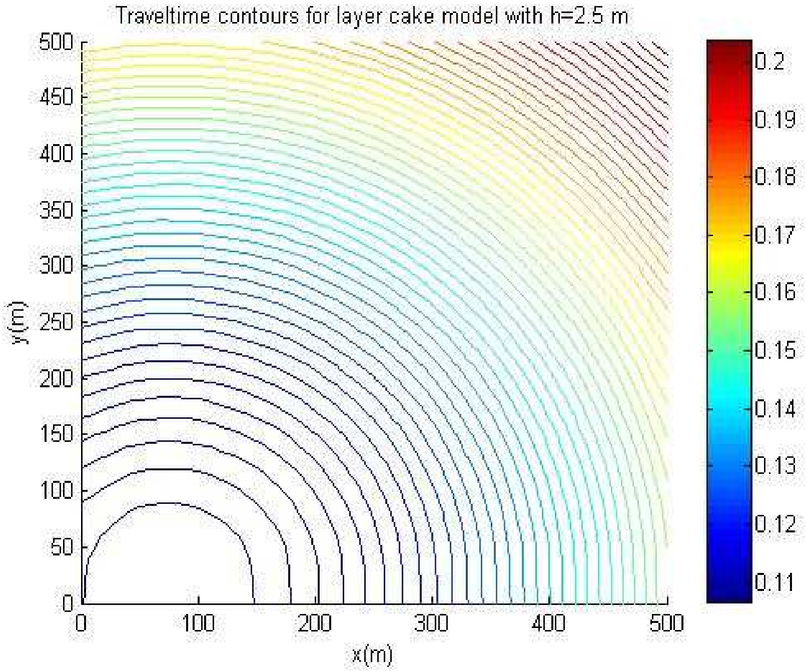


Figure (3.13): Traveltime contours in seconds at the receiver stations computed for the layer cake model (Table (3.1)) with two different grid spacing of 2.5m and 5.0m.

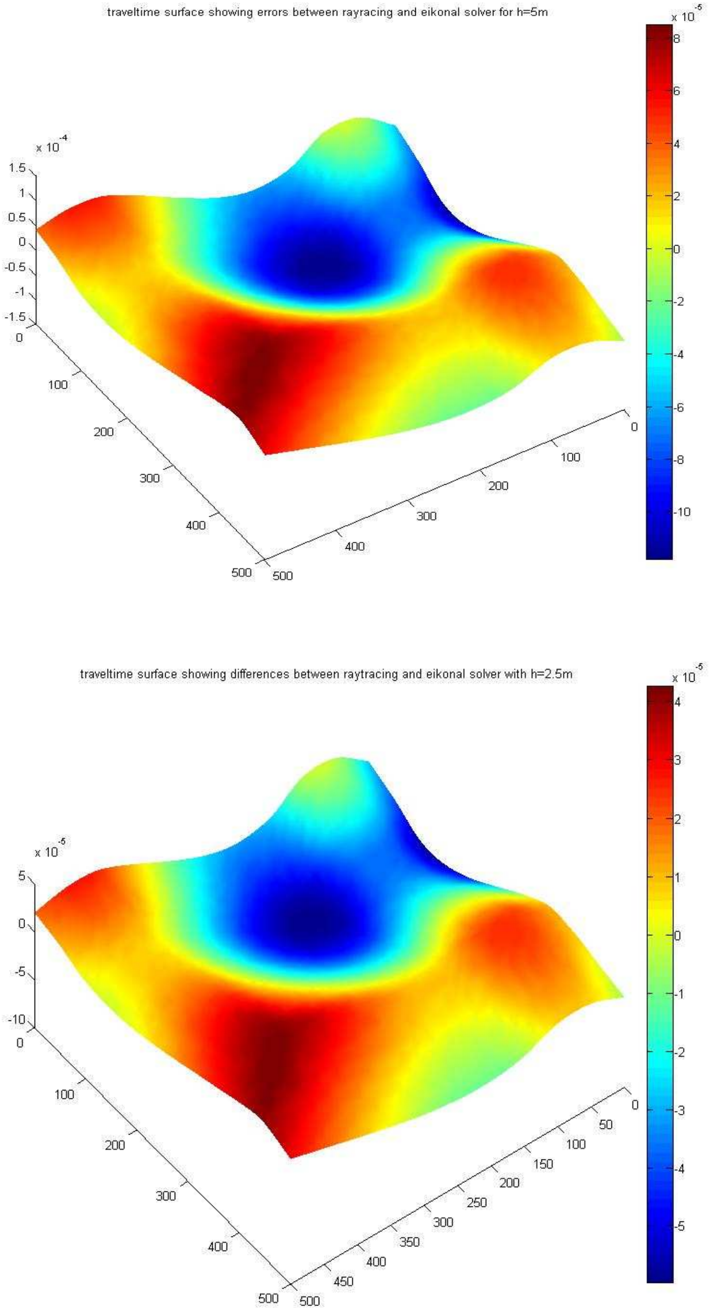


Figure (3.14): Traveltime surface at the receiver stations showing differences in seconds between ray tracing traveltimes and eikonal solver traveltimes for the layer cake model with two different grid spacings of 5.0m and 2.5m.

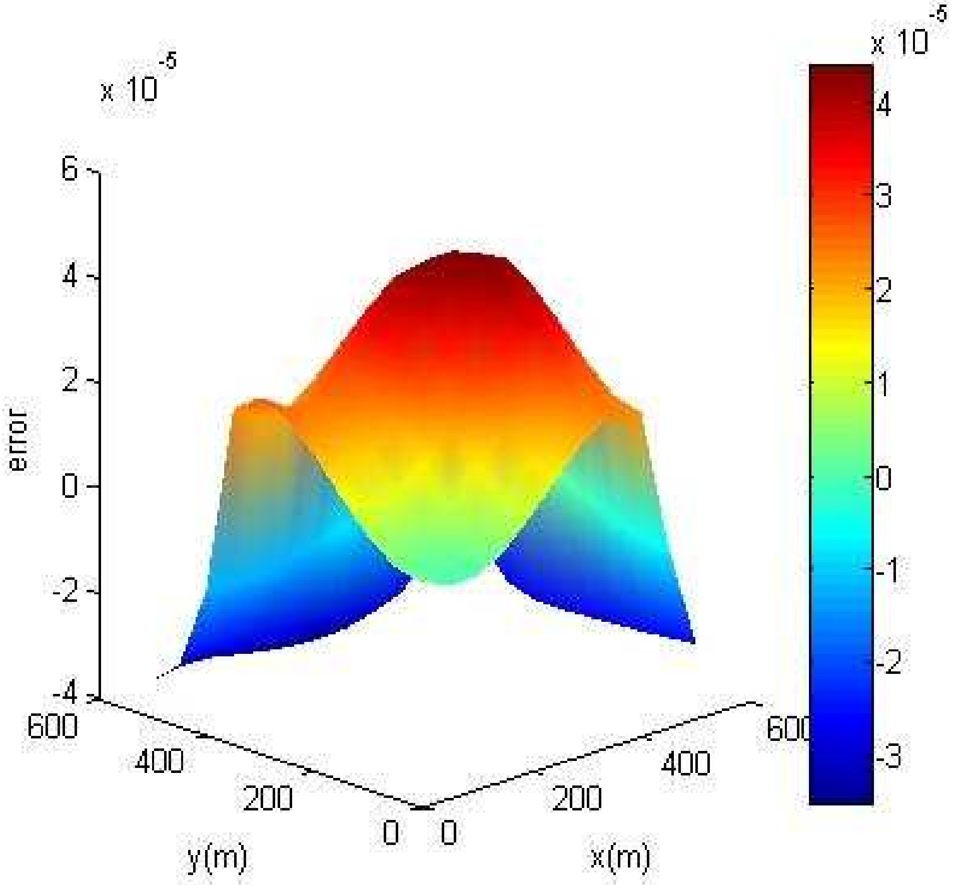


Figure (3.15): A surface showing traveltime differences between forward and backward propagations computed for the layer cake model with a grid spacing of 5.0 m. The errors for the source and all the 121 receivers are less than ±45microseconds.

3.8 Ray-path Computation

In chapter 5, I will use the ray-path components to construct the objective function which will be used by the optimisation algorithm to reconstruct the slowness model parameters. I derive algebraic expressions to trace a ray-path starting at a receiver and terminating at its associated source. Such algebraic expressions are based on the current position of the ray-path and the direction cosines at the same position, which are derived from the gradient components of the traveltimes (Hartley, 2002). The algorithm involves tracing, according to the slowness model parameters, the path of a ray starting at the receiver position and ending at the source position. When the ray passes through a discretized Earth model composed of regular cubes with isotropic slowness, the ray always terminates at the boundaries of cubes except, maybe, for the last position. That is, the algorithm allows the ray to traverse a cube once at a time based on translation to compute the ray-path segment length spent within such a cube. Within a cube, the direction cosines remain constant and the ray-path segment is a straight line due to the isotropic slowness. The new position of the ray-path is obtained and subsequently the direction cosines are recalculated at the new position.

3.8.1 Ray-path Problem

Starting at a source position S_m, traveltimes are computed at all nodes based on the eikonal equation using the algorithm in section 3.7.4. Then given a receiver position R_n within the 3D medium, the objective is to trace the least traveltime ray-path between the source and the receiver and to compute ray-path segments lengths within every traversed cell. The algorithm starts at the receiver position R_n and traces the ray-path all the way to the source position S_m through the solutions of the initial value problem,

$$l(t) = -\nabla T(x, y, z), \quad \text{given} \quad l(0)=\{R_x, R_y, R_z\}. \tag{3.8.1}$$

While tracing back the shortest ray-path, the segments lengths l_{mnq} within every traversed cell (cube) q are calculated. The receiver can be located anywhere within the cube.

3.8.2 Gradient Components of Traveltime

The algorithm initially computes the gradient components of traveltimes at the corners of the receiver-cube. A five-point stencil is used for such calculations,

$$f' = \frac{f(x-2h)+8(f(x+h)-f(x-h))-f(x+2h)}{12h}. \tag{3.8.2}$$

The stencil has local accuracy of fourth order. Moreover, the end terms in the last equation indicate that a receiver position must be at least three nodes to the inside of the medium boundaries. To compute the gradient components $\{\tau_x, \tau_y, \tau_z\}$ at an initial position, trilinear interpolation is used. For example, the eight x-components of the calculated gradient at the receiver-cube corners are first interpolated along the x-axis. This would transform the cube into a square and thus reducing x-components of the gradient to four at the corners of the square. Still, the receiver is part of the square. Second, the four x-components of the gradient at the receiver-square corners are interpolated again, but this time along the y-axis transforming the square into a line segment and further reducing x-components of the gradient to two at endpoints of the line segment. Further, the line segment passes through the receiver. Last, the two x-component of the gradient at the receiver-line end points are interpolated along the z-direction transforming the line segment into a single point corresponding to the receiver position. The process is repeated for the y-components and z-components of the gradient, thus obtaining the three components of the gradient at the receiver position.

The gradient components sign indicates which direction the ray-path components would move. For instance, a positive x-component of the gradient suggests that the x-component of the ray-path would move in the negative x-direction and vice versa. The same applies to the other components. This might lead to wrong indexing. Suppose that the receiver is positioned on the node $\{i+1, j, k\}$ and the x-component of the gradient is positive. This situation suggests that the ray-path would initially traverse through the cube $\{i, j, k\}$ instead of cube $\{i+1, j, k\}$, therefore; the indices may need to be adjusted before traversing.

3.8.3 Direction Cosines

Once the indices for the correct cell through which the ray will pass are adjusted, the components of the direction cosines $\{u_x, u_y, u_z\}$ at the receiver position are computed. Such direction cosines are calculated using the following formulas,

$$\begin{aligned} u_x &= \frac{\tau_x}{\sqrt{\tau_x^2+\tau_y^2+\tau_z^2}} \\ u_y &= \frac{\tau_y}{\sqrt{\tau_x^2+\tau_y^2+\tau_z^2}} \\ u_y &= \frac{\tau_z}{\sqrt{\tau_x^2+\tau_y^2+\tau_z^2}}. \end{aligned} \tag{3.8.3}$$

The direction cosines are the cosines of the angles between the ray-path and x, y, and z axes in a Cartesian coordinate system. They give the direction along which the ray moves from the current position (within its cube) to the next position (boundary of the next cube). In addition, the ray-path is straight within each cube due to the discretization.

3.8.4 Distance to and Position of next Point

A line in space is presented through parametric equations. Such parametric equations can be determined uniquely either by two distinct points through which the line passes or a point through which the line passes and a nonzero vector, which is parallel to the line. Mathematically the parametric equations of a line are expressed as,

$$x = x_1 + a_1 t, \; y = y_1 + a_2 t, \; z = z_1 + a_3 t. \tag{3.8.4}$$

The point $\{x_1, y_1, z_1\}$ is given and belongs to the line and the vector $\{a_1, a_2, a_3\}$ is parallel to the line. On the other hand, there are many ways to represent a plane. One useful representation for a plane is the normal form by a point on the plane and a normal vector to the plane. The plane equation is written as,

$$a(x - x_1) + b(y - y_1) + c(z - z_1) = 0 \tag{3.8.5}$$

The point $\{x_1, y_1, z_1\}$is known and belongs to the plane and $\{a, b, c\}$ is the normal vector. The distance from an arbitrary 3D point $\{x_0, y_0, z_0\}$ to the plane can be computed by using the dot product. Choose a point $\{x, y, z\}$ that belongs to the plane and let $\{a_1, a_2, a_3\}$be the vector corresponding to the two points. The distance is the projection of the vector $\{a_1, a_2, a_3\}$ onto the normal vector as shown in Figure (3.16),

$$d = \left| \frac{ax_0 + by_0 + cz_0 - d_0}{\sqrt{a^2 + b^2 + c^2}} \right|; \quad d_0 = (ax + by + cz). \tag{3.8.6}$$

If the normal vector is a unit vector, then equation (3.8.6) becomes,

$$d = \left| ax_0 + by_0 + cz_0 - d_0 \right|. \tag{3.8.7}$$

It is clear from equation (3.8.7) that $d_0$is the distance from the origin $\{0, 0, 0\}$ to the plane. The absolute value is used to ensure that the distance is non-negative.

Assume that a ray-path initial position $\{x_1, y_1, z_1\}$ and direction cosines $\{\alpha, \beta, \gamma\}$ are given. Find the distance that the ray will travel from the initial position point to a new position point on a plane whose normal distance to the origin is D and $\{a, b, c\}$ as its direction cosines along the normal to the origin as shown in Figure (3.17). First define a vector $\vec{v}$starting at the origin $\{0, 0, 0\}$ and terminating at the initial position point $\{x_1, y_1, z_1\}$,

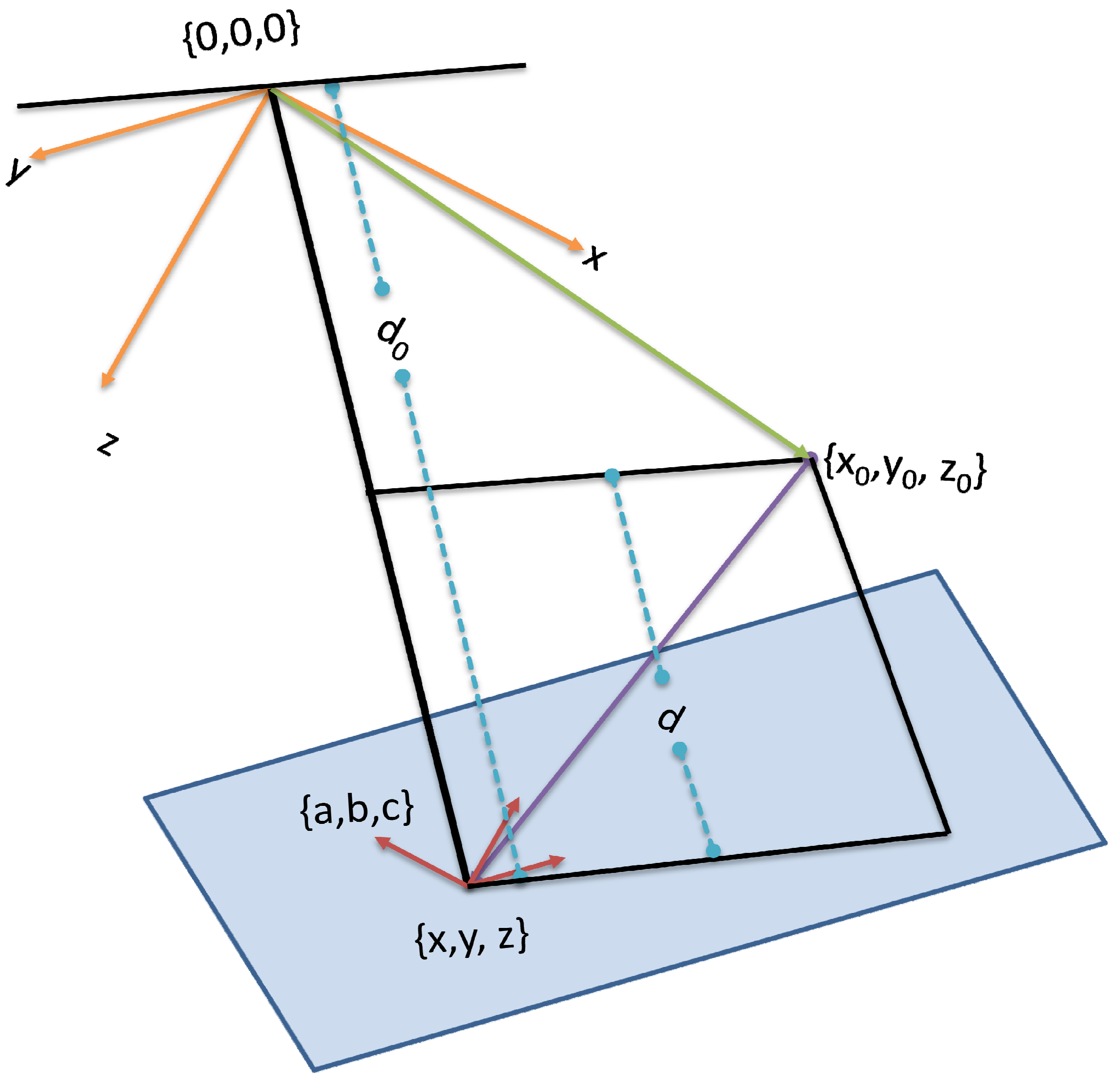


Figure (3.16): Distance from a point to a plane.

$$\vec{v} = x_1\hat{i} + y_1\hat{j} + z_1\hat{k}. \tag{3.8.8}$$

The normal vector to the plane has a length of unity and is expressed as,

$$\vec{n} = a\hat{i} + b\hat{j} + c\hat{k}. \tag{3.8.9}$$

Also, define the vector along the ray as,

$$\vec{u} = \alpha\hat{i} + \beta\hat{j} + \gamma\hat{k}. \tag{3.8.10}$$

From Figure (3.17), it is clear that the normal distance from the origin to the plane is expressed as,

$$D = d_1 + d_2. \tag{3.8.11}$$

But, the distance d_1 is the length of the projection of vector $\vec{v}$ onto $\vec{n}$ which can be obtained from the dot product of the two vectors as,

$$d_1 = |\vec{v}|\cos\theta = \frac{\vec{n}.\vec{v}}{|\vec{n}|} = ax_1 + by_1 + cz_1. \tag{3.8.12}$$

Also notice that d_2 represents the orthogonal projection of the initial point $\{x_1, y_1, z_1\}$ into the plane whereas d represents the travelled distance between the initial points and the new point on the plane. Thus, one can form two vectors: the first vector along vector $\vec{n}$ and has a magnitude of d_2; the second one along vector $\vec{u}$ and has a length of d. Using trigonometry, then the travelled distance is given as,

$$d = \frac{d_2}{\cos\theta}, \tag{3.8.13}$$

where θ represents the angle between $\vec{n}$ and $\vec{u}$. To obtain $\cos\theta$ we can use dot product,

$$\cos\theta = \frac{n.u}{|n||u|} = a\alpha + b\beta + c\gamma. \tag{3.8.14}$$

Therefore, the travelled distance is expressed as,

$$d = \frac{d_2}{a\alpha + b\beta + c\gamma} = \frac{D - d_1}{a\alpha + b\beta + c\gamma} = \frac{-(ax_1 + by_1 + cz_1 - D)}{a\alpha + b\beta + c\gamma}. \tag{3.8.15}$$

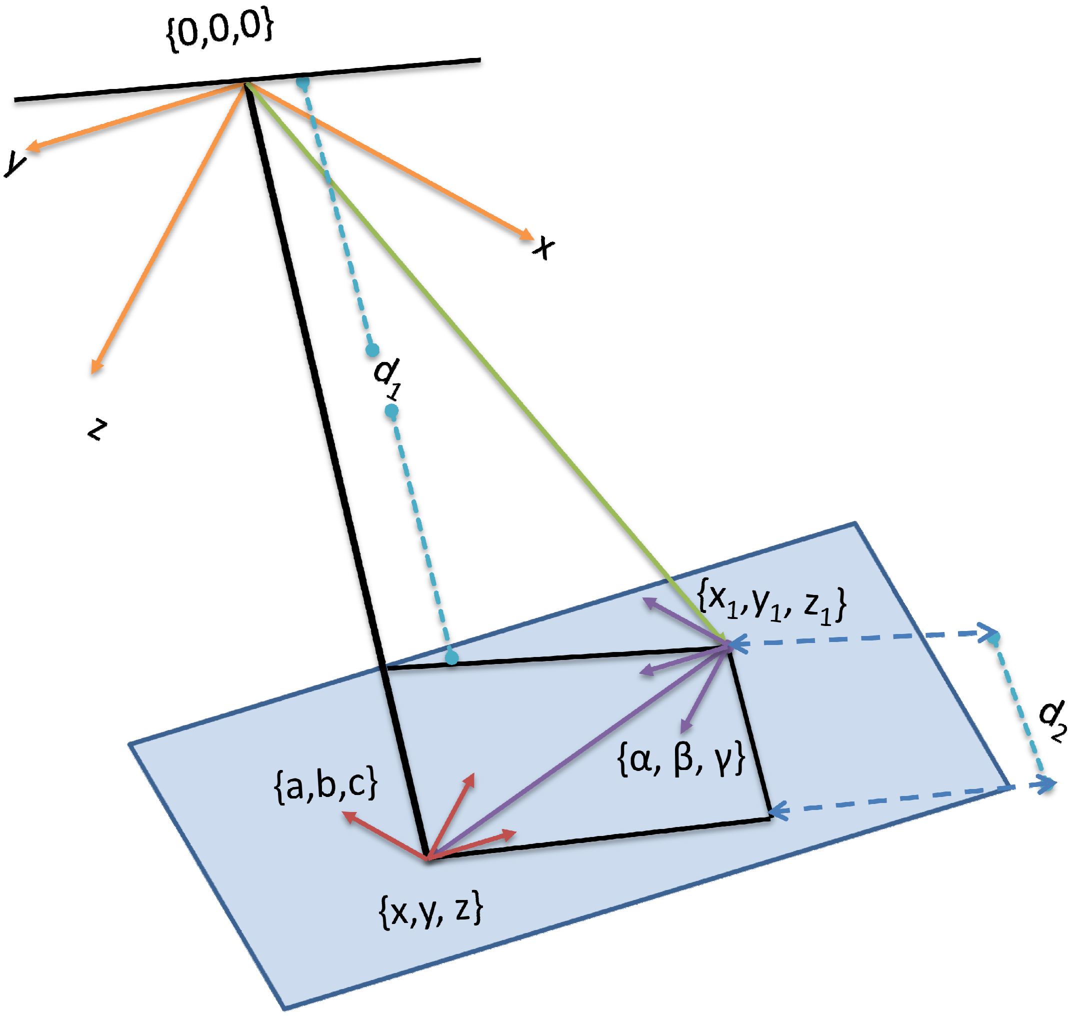


Figure (3.17): Translating a ray from point $\{x_1, y_1, z_1\}$ to point $\{x, y, z\}$ along the known direction cosines $\{\alpha, \beta, \gamma\}$.

When equation (3.8.15) is compared to equation (3.8.6), then it is obvious that d represents the travelled distance but with a negative sign. Further, if the travelled distance is projected along the direction cosines, one obtains,

$$\begin{aligned} d_x &= \left|\left(\frac{ax_1+by_1+cz_1-D_x}{a\alpha+b\beta+c\gamma}\right)\right| \\ d_y &= \left|\left(\frac{ax_1+by_1+cz_1-D_y}{a\alpha+b\beta+c\gamma}\right)\right| \\ d_z &= \left|\left(\frac{ax_1+by_1+cz_1-D_z}{a\alpha+b\beta+c\gamma}\right)\right|. \end{aligned} \tag{3.8.16}$$

The absolute value is applied to ensure that the travelled distance is always non-negative. Using equation (3.8.4) and equation (3.8.10), one can derive the coordinates for the new position $\{x, y, z\}$ as,

$$x = x_1 + \vec{u}d. \tag{3.8.17}$$

Adhering to the discretization in section (3.7.3), one obtains,

$$\begin{aligned} x &= x_1 - \alpha(\min\{d_x, d_y, d_z\}) \\ y &= y_1 - \beta(\min\{d_x, d_y, d_z\}) \\ z &= z_1 - \gamma(\min\{d_x, d_y, d_z\}). \end{aligned} \tag{3.8.18}$$

Starting at a position $\{x_1, y_1, z_1\}$ with known coordinates, I detail here how the ray moves to the next position $\{x, y, z\}$. Whether the current position is inside the cube or belongs to its boundaries, it is always surrounded by six planes, one to the top, one to the bottom, one to the front, one to the back, one to the left, and one to the right. Such configuration is a natural consequence of the discretization process defined in section 3.7.2. Further, cube boundaries are the top face, the front face, and the left face. The other faces belong to other adjacent cubes accordingly. When the current position is inside a cube, the surrounding planes are the six faces of such a cube. However, when the current position is on a face, the situation is treated differently. For instance, suppose that the current position is on the upper face of a cube but not on an edge, then the surrounding planes are the left face, the right face, the front face, the back face, the bottom face, and the sixth plane is the top face of the next cube along the negative z-direction.

Depending on the ray direction cosines, there are only three options need to be tested: the x-direction cosine will point to either the left or the right; the y-direction cosine will direct either to front or the back; the z-direction cosine will indicate either to the top or the bottom. Sections 3.8.2 and 3.8.3 provide us with the necessary tools for computing the direction cosines $\{u_x, u_y, u_z\}$ of the ray at the current position. To use equation (3.8.16), the normal distance from the origin to the plane and the normal direction cosines are required in addition to the current position as well as the ray direction cosines. It is clear that when either a top face or a bottom face is considered, then the normal direction cosines are given as $\{0, 0, \pm 1\}$ since a top face or bottom face is parallel to the xy-plane. The same applies to other situations so that the normal direction cosines are $\{\pm 1, 0, 0\}$ and $\{0, \pm 1, 0\}$ when planes are parallel to the yz-plane and xz-plane, respectively. The normal distance is obtained based on the current position and the direction cosines of the ray. For example, suppose that the z-direction cosine of the ray is negative and the current position belongs to a cube with (i, j, k) indices, then the normal distance is $(k+1)h$ where h is the grid spacing. All the other situations are calculated in the same manner.

Now that all the required parameters are obtained, directional distances are computed from equation (3.8.16) to three orthogonal planes. If any directional distance is zero, then it is set to a very large number. This is to prevent an infinite loop. Each directional distance is compared to the other two and the smallest one is selected. The smallest directional distance is required to translate the current position to the next position using equation (3.8.18). To this point, I have only calculated the first move that the ray travelled from the receiver position through the proper path. The next move is to reference the current position as the receiver position and repeat the process till it reaches the source position. The traversed cubes indexing together with their travelled ray-path segments are saved. This process is robust enough, but may have some minor problems near the source position due to the very small gradients. To address such a problem, four different situations are considered; when the source is inside a cube, when the source is on a node, when the source is on a face, but not an edge, and when the source is on an edge. In these cases, the algorithm works its way until it is close enough to the source position; a straight line is then used to connect the current position to the source position.

3.8.5 Numerical Results

Translating from one cell to the next is based on the follow equations (Hartley, 2002),

$$\begin{aligned} x^{new} &= x^{old} - \alpha l_{mnq} \\ y^{new} &= y^{old} - \beta l_{mnq} \\ z^{new} &= z^{old} - \gamma l_{mnq}, \end{aligned} \tag{3.8.19}$$

where $\{x^{new}, y^{new}, z^{new}\}$ represents the coordinates of the new position on the ray-path, and l_{mnq} stands for the shortest distance within cell q as given by equation (3.8.16). Thus, one obtains the traveltime that started at source m and collected at receiver n as,

$$T_{mn} = \sum_q l_{mnq} p_q . \tag{3.8.20}$$

Equation (3.8.20) represents a linearized relationship between slowness and traveltime and p_k denotes the slowness model parameter within the cell. It can also be used to indicate the accuracy of the ray-path computations.

I used the layer cake velocity models used in section 3.7.5 to calculate the ray-path lengths. I further used equation (3.8.20) to compute traveltimes. The differences between the true traveltimes and those calculated using equation (3.8.20) are shown in Figure (3.18). It is clear from the results that the algorithm for computing the ray-path between a specified source and receiver is so accurate that the errors are negligible. This is important because the ray-path lengths are used for constructing the objective function.

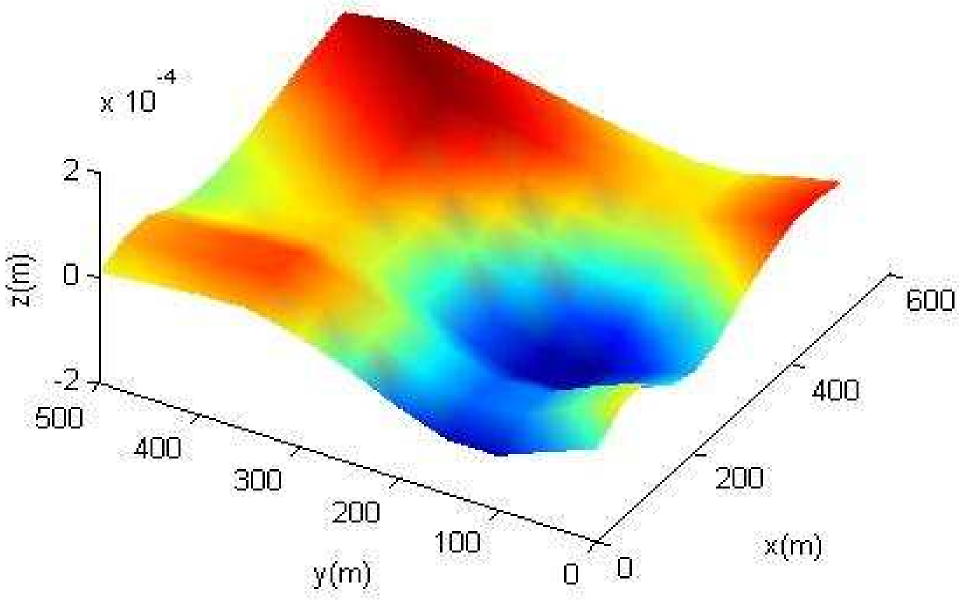


Figure (3.18): A surface showing traveltime differences between forward propagations and those derived using equation (3.8.20) for the layer cake model with grid spacing of 5.0 m. The errors for all the 121 receivers are less than ±200microseconds.

3.9 Conclusion

In this chapter I have developed the forward modelling and ray-path tracing for computing traveltimes and ray-path segments, respectively. The motivation is to develop all the necessary tools needed for locating induced micro-seismic events within a hydrocarbon reservoir and for reconstructing a detailed velocity model through optimisation. The accuracy of the computed traveltimes and ray-paths are very good. The algorithms are developed as object oriented C++ codes within Microsoft Visual Studio platform.

References

Cao, S., and S. Greenhalgh, 1994, Finite-difference solution of the eikonal equation using an efficient, first-arrival, wavefron tracking scheme: Geophysics, **59**, no. 4, 632-643.

Cerveny, V., 2001, Seismic Ray Theory: Cambridge University Press.

Chapman, C., 2004, Fundamentals of Seismic Wave Propagation: Cambridge University Press.

Gazdag, J., 2000, Travel-time computations for true-amplitude migration of constant-offset seismic data: Exploration Geophysics, **31**, 300-303.

Hartley, B.M., 2002, Exact travel time calculations for simple three-dimensional Earth models in seismic exploration using computer algebra: Computers & Geosciences, **28**, 327–336.

Hole, J.A., and B.C. Zelt, 1995, 3-D finite-difference reflection traveltimes: Geophys. J. Int., **121**, 427-434.

Julian, B.R., and D. Gubbins, 1977, Three-dimensional seismic ray tracing: J. Geophys. Res., **43**, 95-114.

Kim, S., 2002, 3-D eikonal solvers: First-arrival traveltimes: Geophysics, **67**, no. 4, 1225–1231.

Kim, S., and R. Cook, 1999, 3-D traveltime computation using second-order ENO scheme: Geophysics, **64**, no. 6, 1867–1876.

Lafond, C.F., and A.R. Levander, 1990, Fast and accurate dynamic raytracing in heterogeneous media: Bulletin of the Seismological Society of America, **80**, no. 5, 1284-1296.

Liner, C.L., 2004, Elements of 3D Seismology: PennWell Corporation.

Moser, T.J., 1991, Shortest path calculation of seismic rays: Geophysics, **56**, no. 1, 59-67.

Podvin, P., and Lecomte, 1991, Finite difference computation of traveltimes in very contrasted velocity models: a massively parallel approach and its associated tools: Geophys. J. Int., **105**, 271 - 284.

Qin, F., Y. Luo, K.B. Olsen, W. Cai, and G.T. Schuster, 1992, Finite-difference solution of the eikonal equation along expanding wavefronts: Geophysics, **57**, no. 3, 478-487.

Robinson, E.A., and D. Clark, 2003, The eikonal equation and the secret Pythagorean theorem: The Leading Edge, **22**, no. 8, 749-750.

Sava, P., and S. Fomel, 2001, 3-D traveltime computation using Huygens wavefront tracing: Geophysics, **66**, no. 3, 883–889.

Schneider, W.A., K.A. Ranzinger, A.H. Balchs, and C. Kruse, 1992, A dynamic programming approach to first arrival traveltime computation in media with arbitrarily distributed velocities: Geophysics **57**, no. 1, 39-50.

Sethian, J.A., and A.M. Popovici, 1999, 3-D traveltime computation using the fast marching method: Geophysics, **64**, no. 2, 516-523.

Vidale, J.E., 1990, Finite-difference calculation of traveltimes in three-dimensional: Geophysics, **55**, no. 5, 521-526.

Every reasonable effort has been made to acknowledge the owners of copyright material. I would be pleased to hear from any copyright owner who has been omitted or incorrectly acknowledged.

Chapter 4

Source Location and Origin Time

4.1 Introduction

The main topic of this chapter is the development of a technique for estimating micro-seismic event position and origin time in a 3-dimensional heterogeneous medium with the help of the tools developed in Chapter 3. Given a 3D discretized medium and its corresponding slowness model parameters, I determine absolute positions coordinates and origin times individually for all the micro-seismic events by using an efficient and robust systematic grid search algorithm. To estimate an event position coordinates, the algorithm builds an objective function by fitting the model response traveltimes to a finite set of observed data through the use of the l_1 norm. The algorithm avoids using the derivatives, commonly used in linearized methods. Furthermore, the global minimum is appropriately obtained and hence the correct positions. An error analysis is presented to confirm the robustness of the algorithm.

4.2 Micro-seismic Event Model

When an induced micro-seismic event occurs within a hydrocarbon reservoir, it produces hi-frequency seismic body waves (P- wave and S-wave) propagating with different phase velocities through the 3D medium surrounding the event. The time it takes the seismic wave to travel from a source point in space to a receiver point within the medium depends on the medium’s velocity distribution. This time is known as traveltime. If the medium has normal dispersion, the frequency of the travelling seismic wave also affects the traveltime: the higher the frequency, the longer the traveltime. Using Fermat’s principle, we can represent the arrival time at a receiver station as a path integral as given in Chapter 1,

$$T_{mn} = T_m^0 + \int_{\Gamma_{mn}} p(x, y, z)\, dl, \qquad (4.2.1)$$

where T_{mn} represents the arrival time of a body wave for a micro-seismic event originated at the source S_m and recorded at receiver station R_n and T_m^0 stands for the time at which the micro-seismic event S_m occurred and known as origin time. The Fermat ray-path Γ_{mn} is the path of least overall traveltime from S_m to R_n, the slowness field of the medium is denoted by $p(x, y, z)$, and dl is an infinitesimal segment of the ray-path length.

To accurately locate the position coordinates and origin time of a micro-seismic event, one needs to have many receiver stations well positioned within the hydrocarbon reservoir in order to obtain a wide aperture. Besides, the signal-to-noise ratio of the waveform is equally important. One needs to pick arrival times either manually or automatically. Possibly, the most accurate way to obtain arrival times is to manually pick the onset of either P- or S-waveform, but it is very time consuming. Still, the picked arrival times would always have some errors attributed to the background noise and the sampling rate of the waveform: the higher the sampling rate is, the more accurate the picked arrival times are.

In an isotropic homogenous medium with slowness p, equation (4.2.1) is expressed as,

$$T_{mn} = T_m^0 + p\sqrt{(x_{receiver} - x_{source})^2 + (y_{receiver} - y_{source})^2 + (z_{receiver} - z_{source})^2}. \qquad (4.2.2)$$

Such an equation shows that the relationship between the arrival time and source position is nonlinear even with constant slowness model. This also suggests that accurate receiver station coordinates are essential.

Further, because a ray-path depends on the slowness model, equation (4.2.1) is nonlinear with respect to slowness as well as source location. Therefore, the function that maps recorded arrival times into micro-seismic location and origin time is nonlinear regardless of the velocity model.

The micro-seismic event can be thought of as a point source in both space and time. That is, the event is located in a 3D medium space at point $\{x^0, y^0, z^0\}$ and originated at time T^0. The event location is known as hypocentre. Throughout this chapter, I assume that the slowness model parameters, the receiver position coordinates, and the recorded arrival time, known as observed data, are known. In Chapter 6, the assumption that the velocity is known accurately will be relaxed. I need to determine both the event origin time and the source position coordinates. It will be clear as I describe the methodology that the ray-path is not needed.

4.2.1 Background

Micro-seismic events are produced within a hydrocarbon reservoir as a result of production activities. With such passive sources, we lack information about both their positions and timings. Determining the location of these events is a nonlinear inverse problem and under certain situations is multi-modal (Ruzek and Kvasnicka, 2001) ; nevertheless, it is essential for monitoring, characterizing, and/or imaging the hydrocarbon reservoir. Many techniques have been developed for using the observed seismic arrival times and a fixed velocity model in order to determine the event position and the event origin time.

One class of techniques is based on tri-axial hodogram analysis assuming that the velocity model within the concerned medium is homogenous and isotropic (Drew et al., 2008; Moriya et al., 1994; Nagano et al., 1989; Soma et al., 2007). Such methods would still work with less accuracy for 1D velocity model, and would require many vertical receiver stations to cover a large aperture (Warpinski et al., 1997). Such a class is considered not adequate for our problem.

Another class of methods is based on derivatives by iteratively solving a linearized problem (Ronde et al., 2007; Waldhauser and Ellsworth, 2000; Zhang and Thumber, 2006; Zhang and Thurber, 2003). Although the iterative updating techniques can make use of both absolute and relative arrival times (Zhang and Thurber, 2003) and can be used to simultaneously solve for velocity model and locations (Zhang and Thumber, 2006), they are very sensitive to the velocity model (Michelini and Lomax, 2004) which may be related to the calculation of traveltime derivatives (Aldridge et al., 2003). This class of methods also depends heavily on the initial model parameters as the problem is nonlinear and multimodal (Ronde et al., 2007). Therefore, this class could be trapped in a relative minimum of the objective function.

A third class of techniques is migration-based using the full wavefield to time reverse the recorded seismic wavefield to focus the energy at the origin time and at the hypocentre of the event (Baker et al., 2005; Gajewski and Tessmer, 2005; McMechan et al., 1985; Rentsch et al., 2007). The accuracy of picking the first arrival times for P- and S-wave is not an issue for this class (Rentsch et al., 2007); nonetheless, these techniques require a spatially dense recording network deployed in the surrounding area of the source (McMechan et al., 1985) and accurate velocity model (Gajewski and Tessmer, 2005). They are also extremely time-consuming methods as they use the full wavefield.

A fourth class of methods is search-based using systematic or probabilistic search over predefined parameters to locate the hypocentre and origin time (Aldridge, 2000; Aldridge et al., 2003; Lomax et al., 2007; Nelson and Vidale, 1990; Prugger and Gendzwill, 1988; Ruzek and Kvasnicka, 2001; Vesnaver et al., 2008). This class which avoids the calculation of the derivatives and seek a global minimum of the objective function to generate more accurate solutions (Aldridge et al., 2003; Ruzek and Kvasnicka, 2001), still requires the identification of seismic phases and the picking of the first arrival times for either P- or S-wave (Rentsch et al., 2007).

The search-based methods are comparatively easy to implement and robust to optimise when used for obtaining the event location and origin time. In addition, they seek a global solution for the nonlinear and multimodal objective function and thus have a better chance of obtaining superior results than the other methods. Since the problem on hand requires a robust method that converges to true location and origin time, I choose to develop a straightforward systematic grid search algorithm. The grid search algorithm requires more computations (Aldridge et al., 2003). However, it can be modified easily to reduce such calculations (Vesnaver et al., 2008), though such computations are needed to attain more accurate location and origin time of a microseismic event.

4.2.2 Theory and Methodology

Let us assume that we have a set of $N_s N_r$ recorded arrival times resulted from N_s sources and recorded at N_r receiver stations. An induced micro-seismic event is initiated at a source position, and then the generated seismic wave propagates through the medium and is recorded later at various receiver stations with different arrival times. The sources and receiver stations can be anywhere within the vicinity of the reservoir. The 3D medium is discretized according to section 3.7.2, into regular grids resulting in regular cubes and assigned with corresponding slowness model parameters at the corresponding grid points.

Let $\{x_m^s, y_m^s, z_m^s\}$ represents the hypocentre of a micro-seismic event m, $\{x_n^r, y_n^r, z_n^r\}$ denotes the position coordinates of a receiver station n which needs not be on a node point, and $t_{nq}(x_q, y_q, z_q)$ is the computed traveltime started at receiver station n and terminated at node q with coordinates $\{x_q, y_q, z_q\}$. The traveltimes $t_{nq}(x_q, y_q, z_q)$ are calculated using the algorithm described in section 3.7. By definition,

$$t_{nq}(x_q, y_q, z_q) = \int_{\Gamma_{nq}} p\left(x, y, z\right) dl . \tag{4.2.3}$$

Using equation (4.2.1) and (4.2.3), one can obtain,

$$T_{mn} \cong T_m^0 + t_{nm}(x_m, y_m, z_m). \tag{4.2.4}$$

The symbol $\cong$ means that the two sides are approximately equal. Equation (4.2.4) is used to build the objective function,

$$\varphi(x_q^s, y_q^s, z_q^s, T_m^0) = \sum_{n=1}^{N_r} w_n \left\| T_{mn} - (T_m^0 + t_{nq}(x_q, y_q, z_q)) \right\|. \tag{4.2.5}$$

One needs to optimise the objective function for four parameters representing the source position coordinates and the origin time. Further, the objective function is nonlinear with respect to its parameters and could be multi-modal. The double bar in equation (4.2.5) represents a norm, a means for measuring the misfit between observed and calculated data, and w_n is a weighting factor attached to its corresponding observed data.

The most commonly used norms are the l_1 norm and the l_2 norm as described in Chapter 2. Although the algorithm can use either of the two, I use the l_1 norm since it is protected from outliers and because the computation of derivatives is not required. Therefore, the objective function with l_1 norm reads as,

$$\varphi(x_q^s, y_q^s, z_q^s, T_m^0) = \sum_{n=1}^{N_r} w_n \left| T_{mn} - (T_m^0 + t_{nq}(x_q, y_q, z_q)) \right|. \tag{4.2.6}$$

Let me introduce a new variable measuring the difference between observed and calculated data at every single node,

$$\tau_{mnq} = T_{mn} - t_{nq}. \tag{4.2.7}$$

Keep in mind that the forward modelling algorithm, explained in Chapter 3, generates traveltimes $t_{nq}(x_q, y_q, z_q)$ for all nodes for fixed n within the 3D medium. Systematically searching all the grids, we are guaranteed that equation (4.2.7) reduces to T_m^0 at the hypocentre node according to the reciprocity principle. If the last step is repeated for the other receiver stations, one obtains the same result. That is, equation (4.2.7) reduces to T_m^0 for all receiver stations at the hypocentre node. A new l_1 measure is introduced,

$$E_m(x_q^s, y_q^s, z_q^s) = \frac{1}{N_r - 1}\sum_{j=1}^{N_r-1}\sum_{k=j+1}^{N_r}\left|\tau_{mjq} - \tau_{mkq}\right|. \tag{4.2.8}$$

Equation (4.2.8) represents an objective function with only three parameters for source m and attains its global minimum value at the hypocentre position coordinates $\{x_m^s, y_m^s, z_m^s\}$; consequently, obtaining position coordinates for the concerned source. Using equation (4.2.7), we obtain the average origin time as,

$$T_m^0 = \frac{1}{N_r}\sum_{n=1}^{N_r}\tau_{mnq}(x_m^s, y_m^s, z_m^s). \tag{4.2.9}$$

The methodology is repeated for all other sources to obtain the position coordinates and origin times. In doing so, the forward modelling algorithm is executed N_r times for each source point. Collectively, one needs to execute the forward modelling program $N_s N_r$ times to obtain all the sources hypocentres and origin times. The algorithm is very greedy for the memory and due to equation (4.2.7) may have problem when the discretized 3D medium has a very large number of nodes with several receiver stations. Furthermore, the forward modelling algorithm is in direct relation with the medium sampling as explained in section 3.7.2.

In developing the methodology, there has been exclusive emphasis on the computed traveltimes which I presented in Chapter 3. The forward modelling algorithm maps the slowness model parameters into traveltimes according to the eikonal equation and medium sampling. Therefore, the slowness model can be either homogenous or heterogeneous and the heterogeneity can be in one-, two-, or three-dimensions. The ray-path is a function of the slowness model and hence related to source position indirectly. Because I assume that the slowness model is fixed, the ray-path remains fixed too.

4.2.3 Source Position and Origin Time Algorithm

All the node points are populated with their corresponding slowness parameters. Starting at a receiver station R_n, traveltimes are computed at all nodes using the algorithm of section 3.7. The source location is assumed to be anywhere within the 3D medium on a grid point. When the source point is actually not on a grid point, the nearest grid point to the source position is determined and labelled as the source location. There are two approaches to determine the hypocentres. In the first approach, I use equation (4.2.7) to subtract the computed traveltimes at every single node from the observed arrival time related to a particular shot S_m and recorded at receiver station R_n and save the result in an array. To reduce traveltime computation, the last process is repeated for all sources. At this stage, I have computed equation (4.2.7) for all sources for only one receiver station. To finish this approach, I apply the above processes starting with the traveltime computation at the other receiver stations. By doing so, I have covered all the receiver stations. Equation (4.2.8) is used to obtain the hypocentres followed by equation (4.2.9) to calculate origin times for all seismic events one by one. Although this approach requires N_r executions of the forward modelling algorithm and thus reduces the computing time, it demands excessively huge amount of memory. Such a requirement makes this approach unattractive, especially when you have a computer with limited amount of memory that cannot hold all calculations.

In the second approach, I use equation (4.2.7) to subtract the computed traveltimes at every single node from the observed arrival time related to a particular source S_m and recorded at receiver station R_n and save the result in an array. To reduce memory allocation, the last process is repeated for all receiver stations. At this point, I have computed equation (4.2.7) for all receiver stations for only one source. Equation (4.2.8) is used to obtain the hypocentre followed by equation (4.2.9) to compute origin time for the seismic event in hand. To finish this approach, I apply the above processes starting with the traveltime computation to the other sources. By doing so, I have recovered all the sources hypocentres and origin times individually. Although this approach requires $N_s N_r$ executions of the forward modelling algorithm and thus increases the computing time, it demands less amount of memory. Such a feature makes this approach attractive, especially when you have a computer with limited amount of memory and many receiver stations.

Now, let me revisit the first approach to improve the source position coordinates and origin time as well as memory allocation. First, I start with a coarse discretization say one tenth of the original one in each direction. This would result in a reduction in the nodes number by a thousand. Subsequently, this reduction would lead to fewer calculations and thus much less computing time and memory allocation. Still, this would produce inaccurate hypocentre coordinates and origin time. Second, I finely discretized the model say twice the original model in each direction. This would result in an increase in the nodes number by a factor of eight. Such an increase would lead to more calculations and thus much more computing time. I then use equation (4.2.7) to subtract the computed traveltimes around the hypocentre node obtained in the first step from the observed arrival times related to all sources S_m such that $m = 1, 2, ..., N_s$ and recorded at receiver station R_n and save the result in an array. It is sufficient to choose eighty fine nodes along each direction around the coarse hypocentre such that there are forty in the positive direction and forty in the negative direction. This would result in an array of $512,000$ cells per each source for each receiver. If the original discretization has 500 nodes in each direction, the result will be an array of $125,000,000$ cells per each source per each receiver! Not to forget that the coarse discretization will result in only $125,000$ cells.

4.2.4 Numerical Results

I used a layer cake velocity model to test the above described algorithm. The velocity model consists of seven layers with different velocities and thickness as listed in Table (4.1) and shown in Figure (4.1). The velocity model contains inversions between the third and fourth layers and between the fifth and sixth layers. Figure (4.2) shows the acquisition geometry I used to test the algorithm. There are three monitoring wells, each with 15 receivers labelled as blue triangles with a 25m vertical separation and positioned at depth between 100m and 450m. The wells occupied three corners of cuboidal model while the fourth corner is saved for the injecting well. I used eight micro-seismic events located within a 25m-layer at a depth between 430m and 450m near the injecting well. The micro-seismic events locations are outside the triangulated array. Such geometry configuration could represent a recording setup used as a permanent monitoring of a producing hydrocarbon reservoir or a CO_2 sequestration reservoir. The arrival times for the eight seismic events are calculated as a P-wave started at each source position and recorded at the receivers' locations using a ray tracing algorithm resulting in 360 observed data points coming from 3x15=45 receiver stations x 8 microseismic events.

Layer	Velocity (m/s)	Thickness(m)
1	3500	75
2	3650	125
3	4000	50
4	3700	75
5	3800	100
6	3400	25
7	4000	50

Table (4.1): Velocity model consists of seven layers.

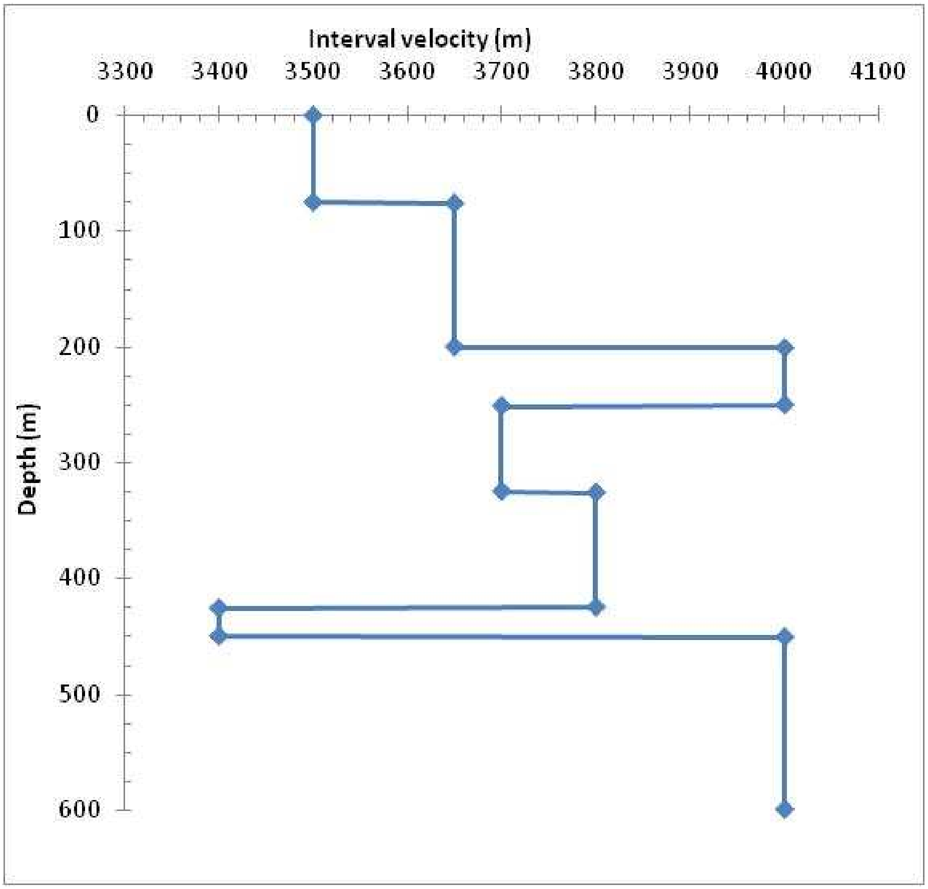


Figure (4.1): Layer cake model showing 7 layers. The z-direction is increasing downward.

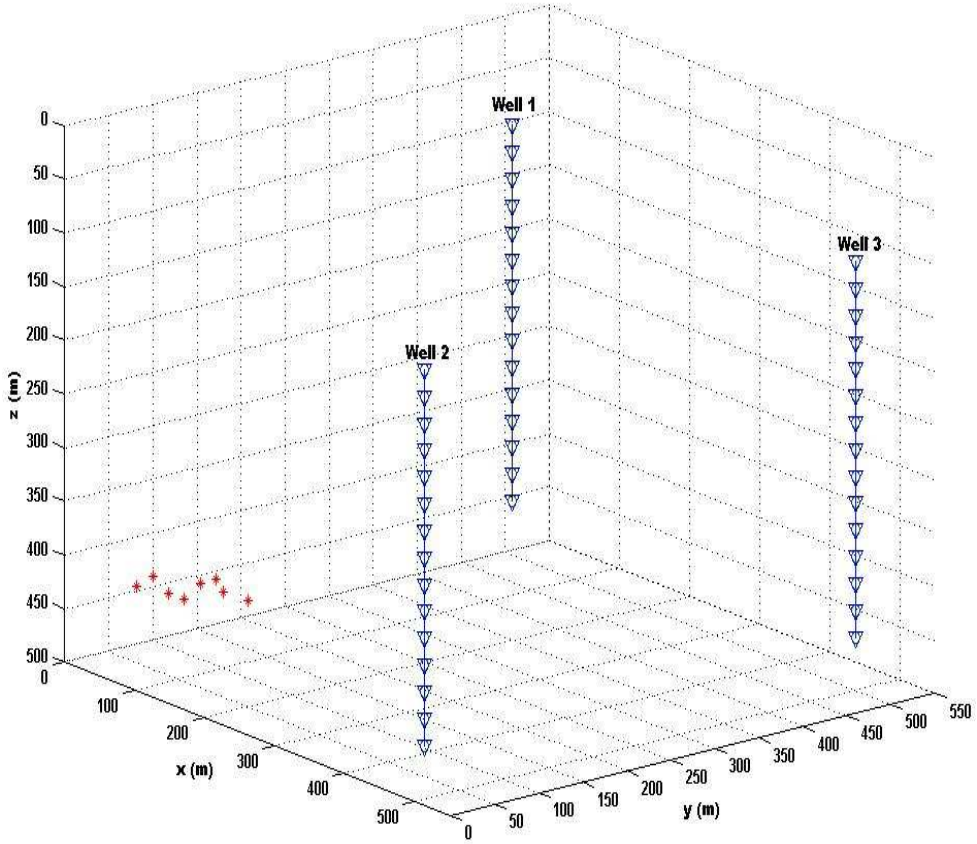


Figure (4.2): 3D view showing three monitoring wells with receivers labelled as blue triangles. The sources are marked as red asterisks and located besides a virtual injecting well. The z-direction is increasing downward.

Figure (4.3) shows volume rendering of the objective function values for the first source. The values of the objective function are computed at every node according to,

$$f(x_q, y_q, z_q) = \sum_{n=1}^{N_r} \left| T_{mn} - t_{nq} \right|, \tag{4.2.10}$$

where I used all the 45 receivers within the three monitoring wells. It is worth mentioning that the objective function shape changes according to the number and positions of the used receiver stations. The objective function values vary between 0.05467 and 3.2785. To illustrate the global minimiser of the objective function, I produced an iso-surface with a value of 0.15 (near the minimum value of the objective function) and embedded it within the volume. It follows that the global minimiser agrees with the source location. In Figure (4.4), I generated an iso-surface with a value of 0.70 and embedded it within the volume to show that there exists a local minimiser. This confirms that the objective function has at least one local minimum besides the global one. In other words, the objective function for this source is multimodal. Therefore, if a gradient based technique is used to locate the source position, it may converge to the local minimiser when the initial model parameters are within the neighbourhood of the local minimiser. Figure (4.5) is a vertical slice along $x = 14$ to confirm the existence of the local minimum.

Table (4.2) shows the original parameters for the hypocentres and origin time. The algorithm accurately obtains the sources location coordinates and origin times and such results are shown in Table (4.3). It is clear from the results that the algorithm for computing hypocentre and origin time are almost exact for error free data.

I used many different simulations for the first source with varying number of geophones within each monitoring well for estimating the hypocentres and origin times. When I use only the receivers within one well, only the z-component were accurately recovered. When all the receivers within any two wells are used, the location parameters are obtained accurately. Further, when the upper 10 receivers or the upper 5 receivers or the upper 3 receivers within any two wells are used, the source parameters are mapped accurately. But, when only the upper receiver within any two wells is used, the recovered source parameters are very different from the true ones.

When the lower 10 receivers or the lower 5 receivers or the lower 3 receivers or the lower 2 receivers within any two wells are used, the source parameters are obtained precisely. But, when only the lower receiver within any two wells is used, the retrieved source parameters are far from the true ones. When the receivers are randomly chosen such that the minimum number of receivers per well is two, the source location parameters are obtained correctly. However, whenever I use only two receivers from two different wells such that I have one receiver per well, I obtain inaccurate source parameters regardless of the receivers' locations.

When I use the three wells with more than one receiver per well, the exact source parameters are obtained. Further, when only the upper receiver within each well is used, the source location parameters are mapped precisely. But, this fails when the middle receiver within each well is chosen. Still, the results are precise when the three receivers are chosen with different levels.

The results described above suggest that we can obtain micro-seismic events parameters when using only two monitoring wells with receiver stations planted at different levels within each monitoring well. Still, when we have only one monitoring well with many 3-component receivers at different levels, we may obtain the source parameters by constraining the algorithm according to the triaxial hodogram method. Having said that, I think it is better to have at least three monitoring wells to image the micro-seismic events within a hydrocarbon reservoir because such events have very low amplitude. Further, we need to have many receiver stations in order to identify and pick the events.

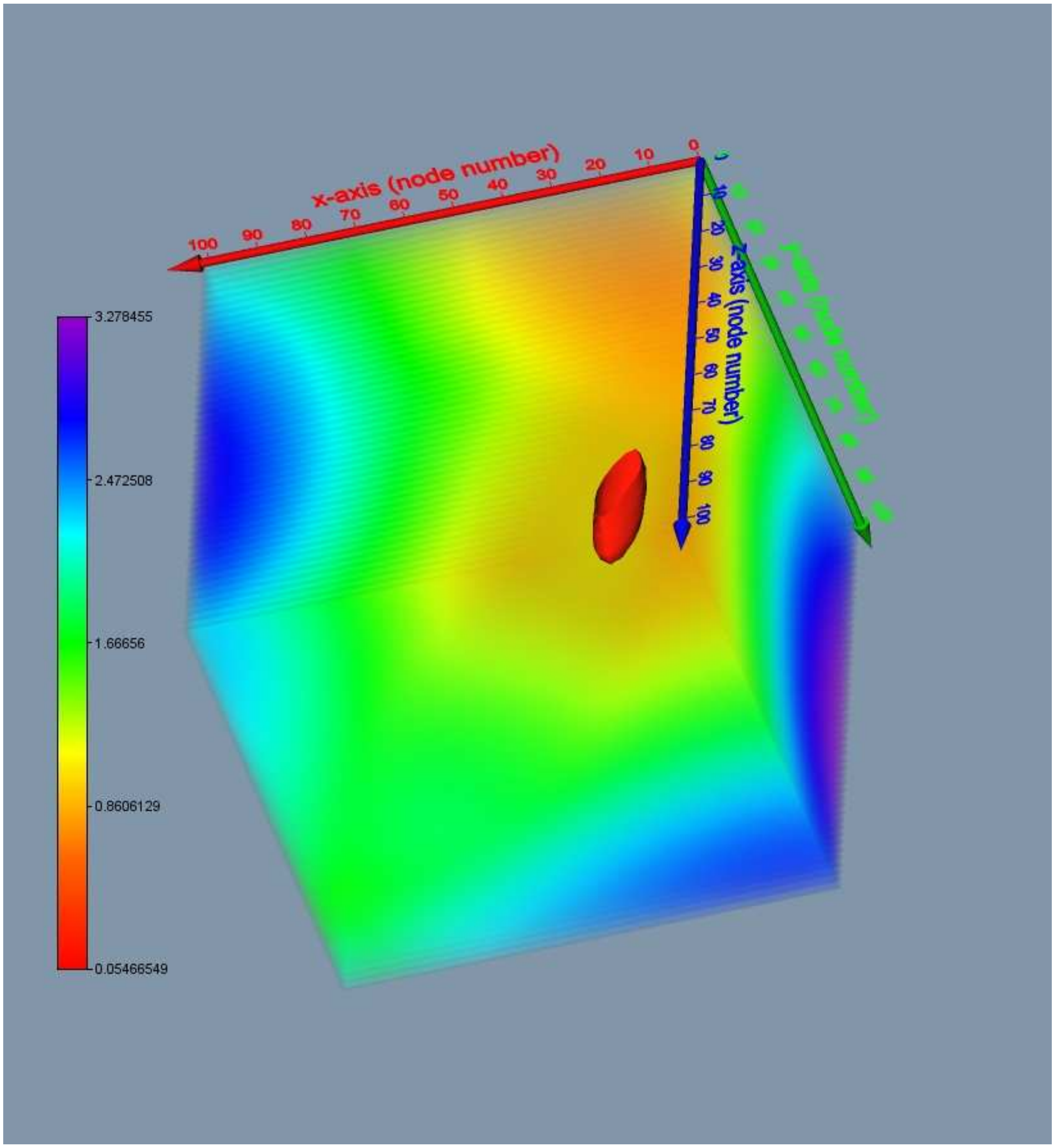


Figure (4.3): Volume rendering of the objective function for the first source together with iso-surface to indicate the source location. The iso-surface is generated with a value of 0.15 which is comparatively near to the minimum value. The axes are labelled with grid points.

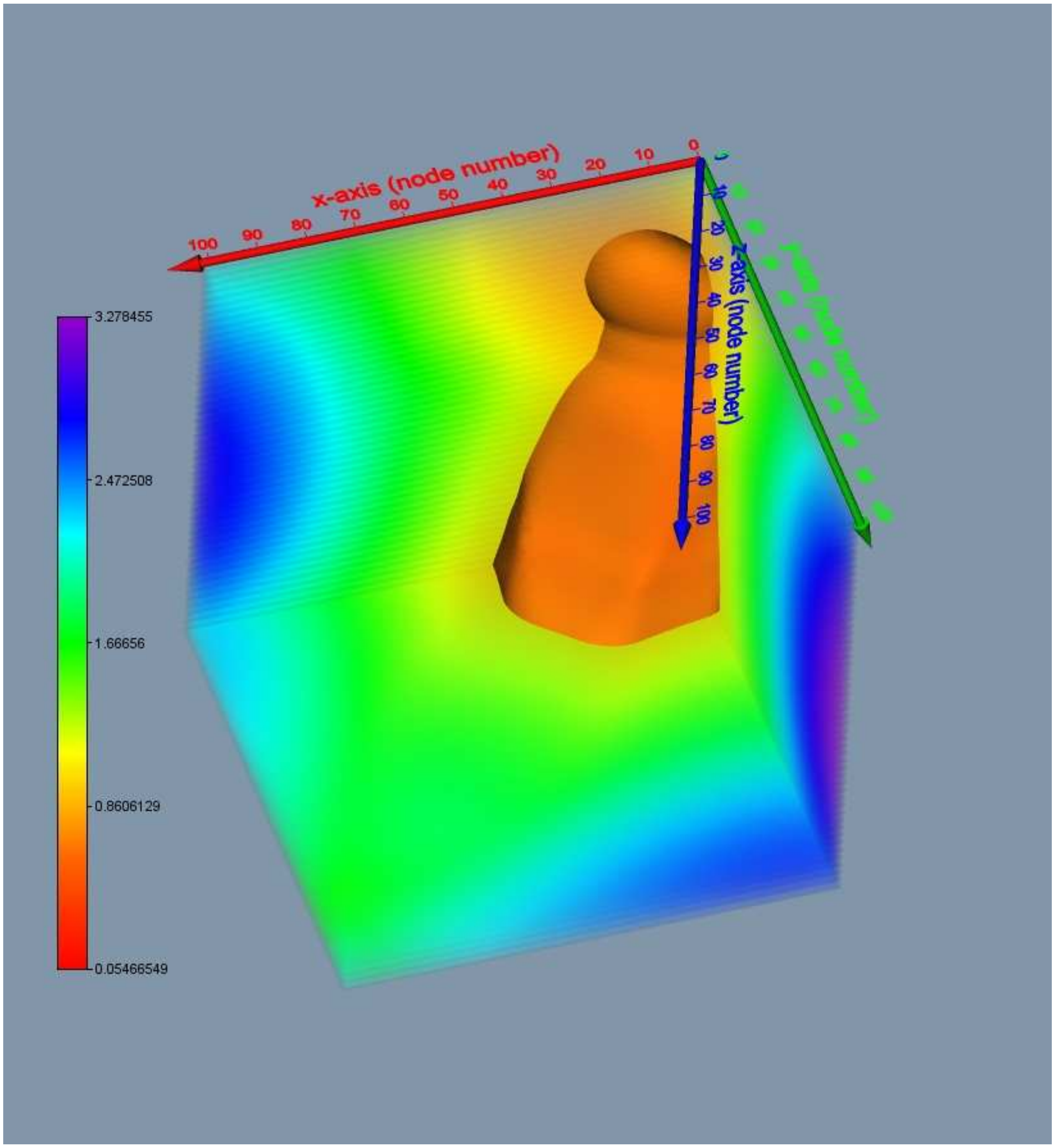


Figure (4.4): Volume rendering of the objective function for the first source together with iso-surface to confirm the existence of a local minimiser. The iso-surface is produced with a value of 0.70 which is relatively far from the minimum value. The axes are labelled with grid points.

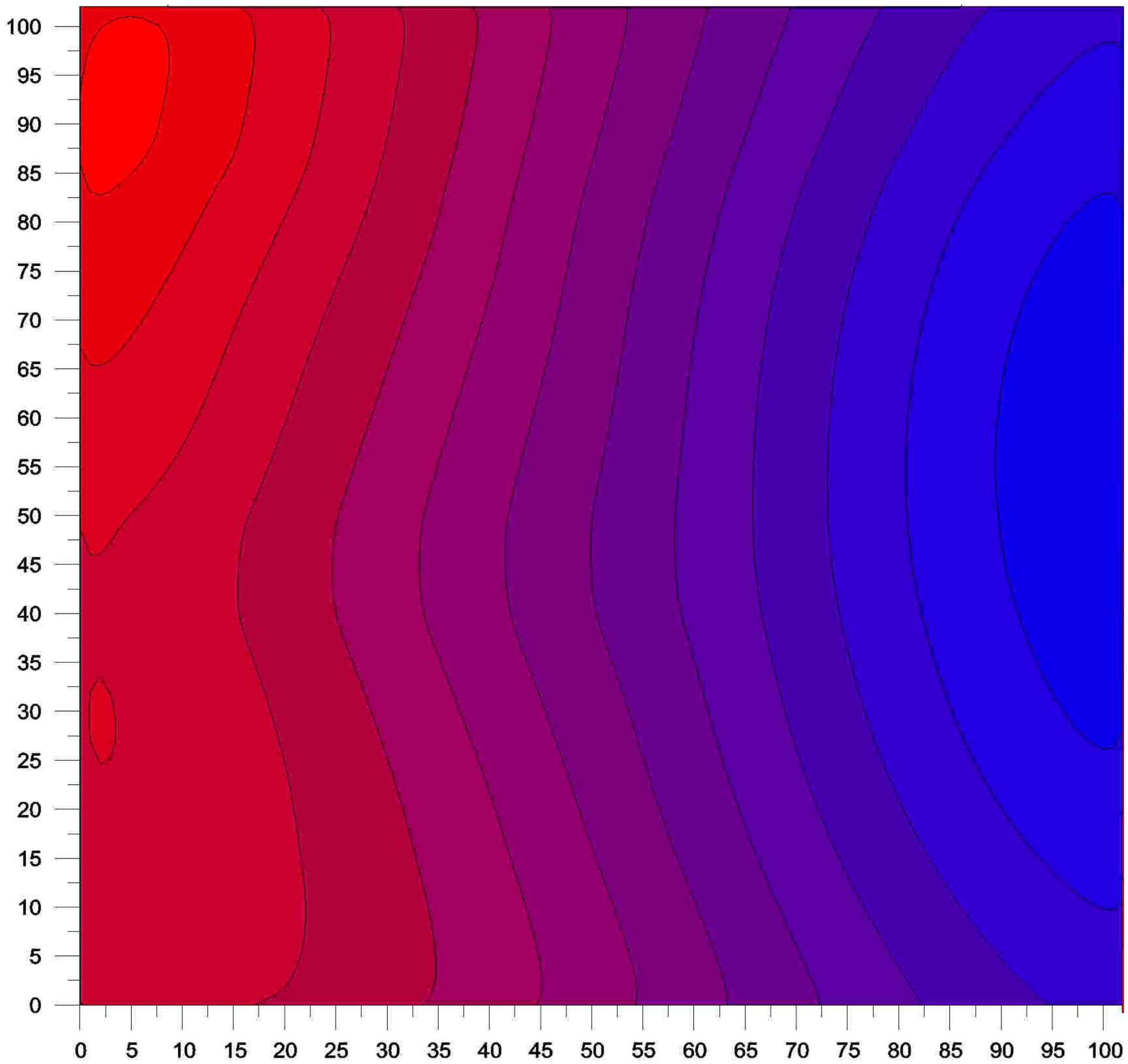
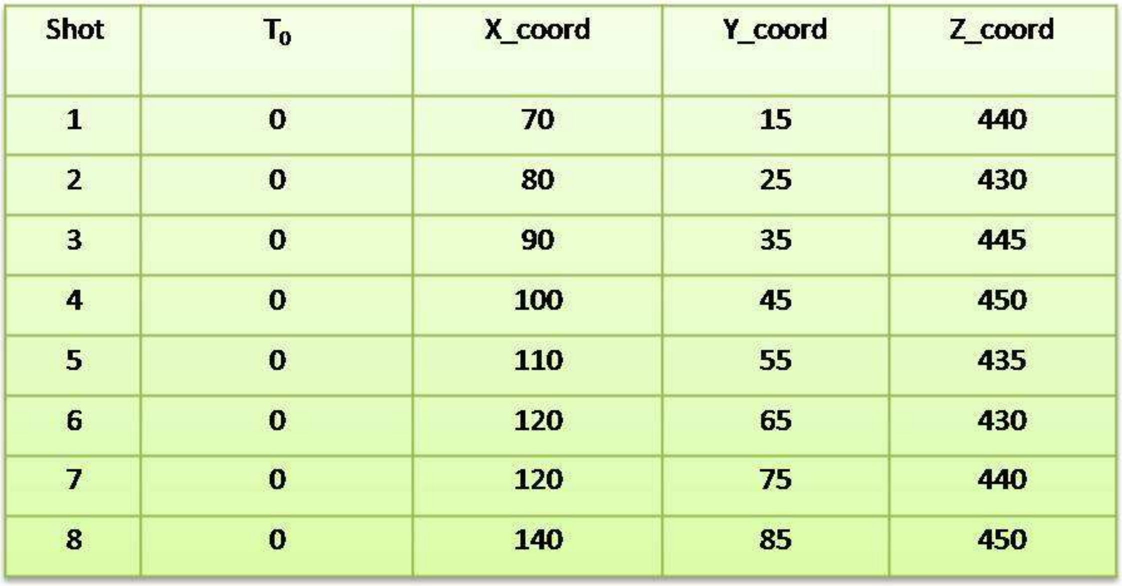
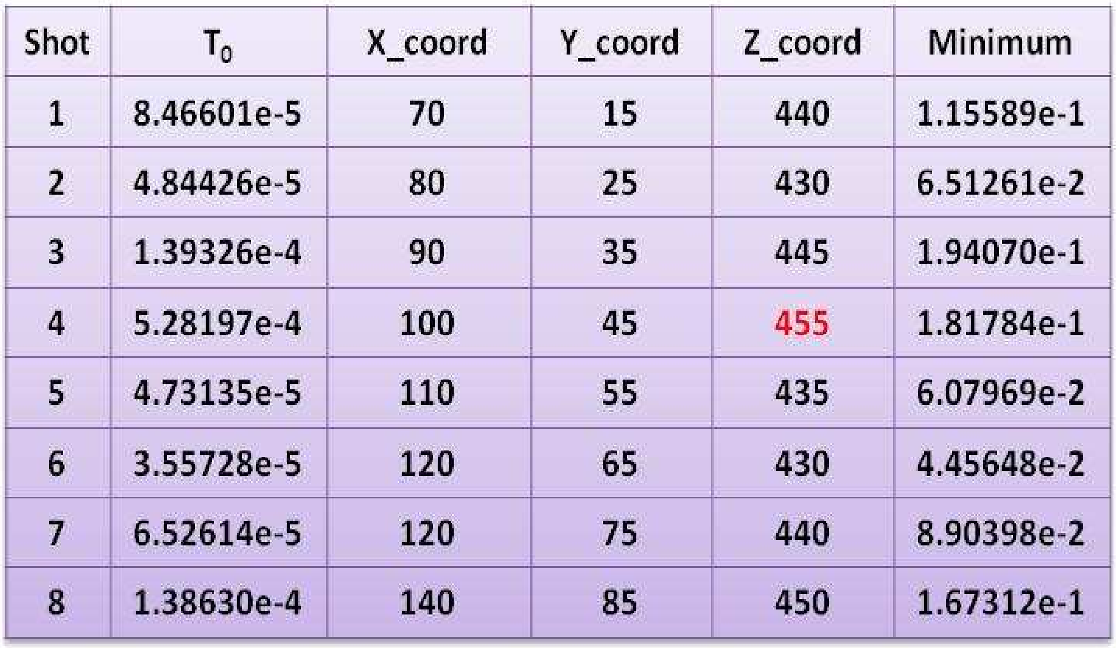


Figure (4.5): A vertical slice along the line x=14 of the objective function for the first source to confirm the existence of a local minimiser. The axes are labelled with grid points.



Shot	T_0	X_coord	Y_coord	Z_coord
1	0	70	15	440
2	0	80	25	430
3	0	90	35	445
4	0	100	45	450
5	0	110	55	435
6	0	120	65	430
7	0	120	75	440
8	0	140	85	450

Table (4.2): Position coordinates and origin times for the eight sources.



Shot	T_0	X_coord	Y_coord	Z_coord	Minimum
1	8.46601e-5	70	15	440	1.15589e-1
2	4.84426e-5	80	25	430	6.51261e-2
3	1.39326e-4	90	35	445	1.94070e-1
4	5.28197e-4	100	45	455	1.81784e-1
5	4.73135e-5	110	55	435	6.07969e-2
6	3.55728e-5	120	65	430	4.45648e-2
7	6.52614e-5	120	75	440	8.90398e-2
8	1.38630e-4	140	85	450	1.67312e-1

Table (4.3): Recovered position coordinates and origin times for the eight sources.

4.3 Error Analysis

There are three main parameters that could introduce uncertainties into the results. These parameters are the velocity model, the picked arrival times and the receiver positions. A possible fourth uncertainty is the calculated traveltimes which was discussed in section 3.7.5. The forward modelling algorithm was shown to work well in different 3D settings and can be controlled, by reducing the grid spacing, to produce accurate traveltimes. Therefore, the error analysis is limited to the velocity model, the picked arrival times and the receiver positions. The analysis is performed on a single microseismic event with four independent random variables. The exact values of the four random variables of such event are {0, 100, 45, 440}, corresponding to origin time, x-coordinate, y-coordinate and z-coordinate, respectively. Out of the 45 receivers, I used only 27 receivers representing the upper 9 receivers within each monitoring well.

There are a total of 304 variables that could introduce errors into the results; 7 variables belong to the velocity model; 216 variables belong to the picked arrival times coming from 27 receivers and 8 events; 81 variables belong to the receiver positions' coordinates. In order to do the error analysis, each variable belonging to a particular error source is treated independently. For example, the seven layers of the velocity model listed in Table (4.1) have seven separate random errors. Initially, random errors are simulated for all variables that belong to one error source while maintaining the remaining variables belonging to the other two error sources fixed. The simulated random errors are added to each variable individually. This process provides three independent simulation experiments; one for each error source. Finally, random errors for the three error sources are simulated and introduced into all the 304 variables simultaneously. Random errors are simulated 250 times for each error experiment.

First, I contaminated the value of each layer that belongs to the velocity model with ±3% random errors. The random velocity errors for the first layer are shown in Figure (4.6).

The random velocity errors for the other six remaining layers are not shown as they have similar trend. The random velocity errors have induced errors in the origin time which have a normal probability distribution with its expected mean μ_t=30μs and standard deviation σ_t=210μs as shown in Figure (4.8). The mean value is very close to the true value of the origin time and the standard deviation has relatively small variability. The velocity random errors have, also, introduced normally distributed errors to the x-coordinate, y-coordinate and z-coordinate as shown in Figure (4.8), Figure (4.9) and Figure (4.10), respectively. The respective means and standard deviations for the error distributions of the hypocentre parameters are μ_x=.16m & σ_x=4.67m, μ_y=.04m & σ_y=4.94m and μ_z=1.14m & σ_z=5.54m. Thus, the expected means are very close to the true values of the hypocentre coordinates. Further, the variability due to the standard deviations is small. This suggests that when the velocity model has ±3% errors, the algorithm with a 95% confidence is capable of retrieving the exact values of the origin time and hypocentre parameters with a maximum error of ±0.42ms and ±10m, respectively. The standard deviations for the four random variables' error distributions are relatively small confirming that the effect of the ±3% random velocity errors on the source parameters is very minor.

Second, I added ±5 (ms) random error into the observed arrival times. The random time errors for the observed arrival time belonging to the first receiver and the first event are shown in Figure (4.11). The remaining 215 picked arrival times have similar random time errors. Like the velocity random errors, the time random errors have introduced normally distributed errors to the origin time with μ_t=78μs and σ_t= 541μs as shown in Figure (4.12). The expected mean is still close to the exact value of the origin time whereas the variability is wider compared to that of the first experiment. Similarly, the time random errors have produced normally distributed errors in the hypocentre parameters as shown in Figure (4.13), Figure (4.14) and Figure (4.15). The corresponding means and standard deviations for the hypocentre parameters' error distributions are μ_x=.76m & σ_x=9.02m, μ_y=-.10m & σ_y=8.11m and μ_z=2.52m &

σ_z=15.19m. On other words, when the arrival observed times have ±5ms errors, the algorithm with a 95% confidence is able to recover the exact values of the origin time and hypocenter parameters with a maximum error of ±1.0ms and ±30m, respectively. The source parameters are more sensitive to the random time errors as explained by the comparatively large standard deviations for the error distributions of the four random variables. This is especially apparent in the z-coordinate parameter.

Third, I introduced ±5m random errors into the receiver position coordinates. The random position errors for the x-coordinate of the first receiver position are shown in Figure (4.16). The random position errors for the other 80 remaining coordinates are not shown as they have similar trend. The receiver position random errors have introduced errors to the origin time which are normally distributed with μ_t=4.4µs and σ=166µs as shown in Figure (4.17). The effect of the receiver position random errors on the origin time is the least compared to the other two error sources. Further, the receiver position random errors have induced normally distributed errors to the hypocentre parameters as shown in Figure (4.18), Figure (4.19) and Figure (4.20), respectively. The matching means and standard deviations for the error distributions of the hypocentre parameters are μ_x=.48m & σ_x=1.9m, μ_y=.34m & σ_y=1.6m and μ_z=1.04m & σ_z=3.5m. Consequently, when the receiver positions have ±5ms errors, the algorithm with a 95% confidence is able to recover the exact values of origin time and the hypocentre parameters with a maximum error of ±0.37ms and ±7m, respectively. The source parameters are the least sensitive to the random receiver position errors.

Last, random errors are generated as specified above and added into all random variables simultaneously. This process simulates the real situation. It has induced errors to the origin time. The distribution of such induced errors is normal with μ_t=54.4µs and σ=207µs as shown in Figure (4.21). Figures (2.22), (2.23) and (2.24) show that the simulated random errors have produced normally distributed errors in the hypocentre parameters. The expected means and standard deviations of such distributions are

μ_x=0.64m & σ_x=4.9m, μ_y=0.50m & σ_y=5.34m and μ_z=0.60µs & σ_z=5.50m. For this experiment, the algorithm with a 95% confidence is capable of recovering the exact values of the origin time and the hypocentre parameters with a maximum error of ±0.414ms and ±9m, respectively. This suggests that the algorithm is robust in obtaining the event parameters even with moderate errors.

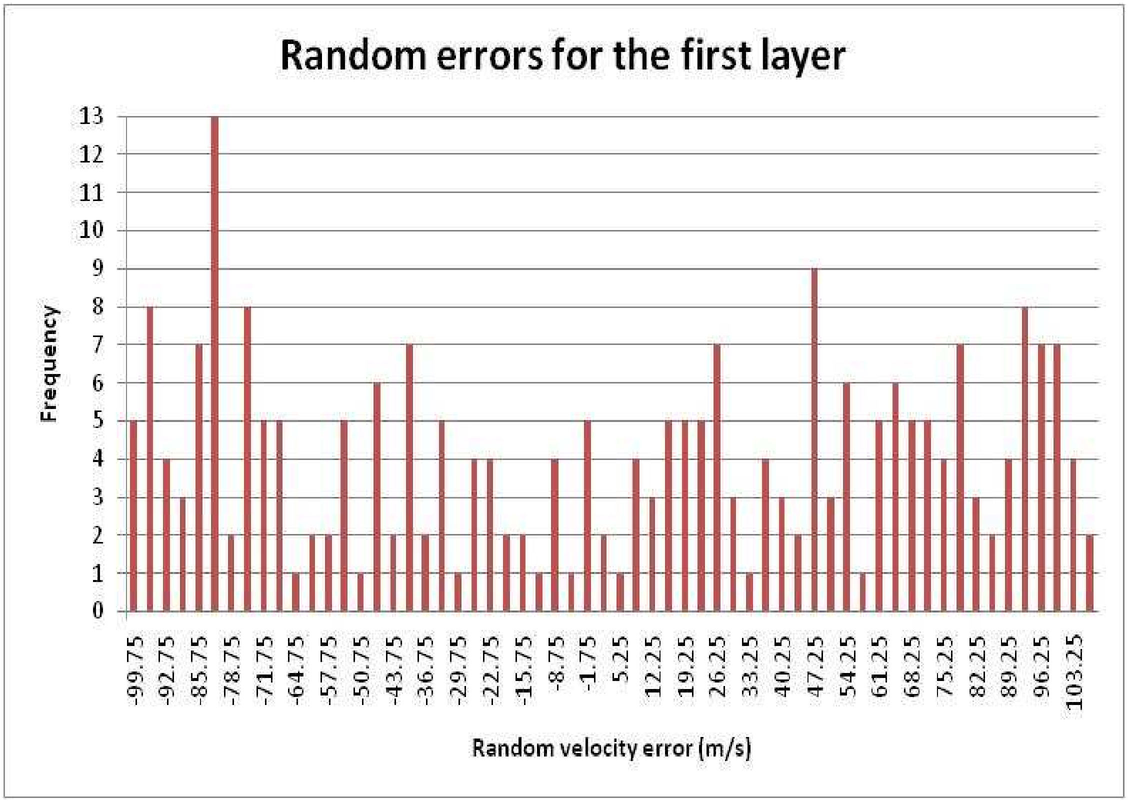


Figure (4.6): Simulation of ±3% random errors for the velocity of the first layer. The velocity random errors were generated 250 times.

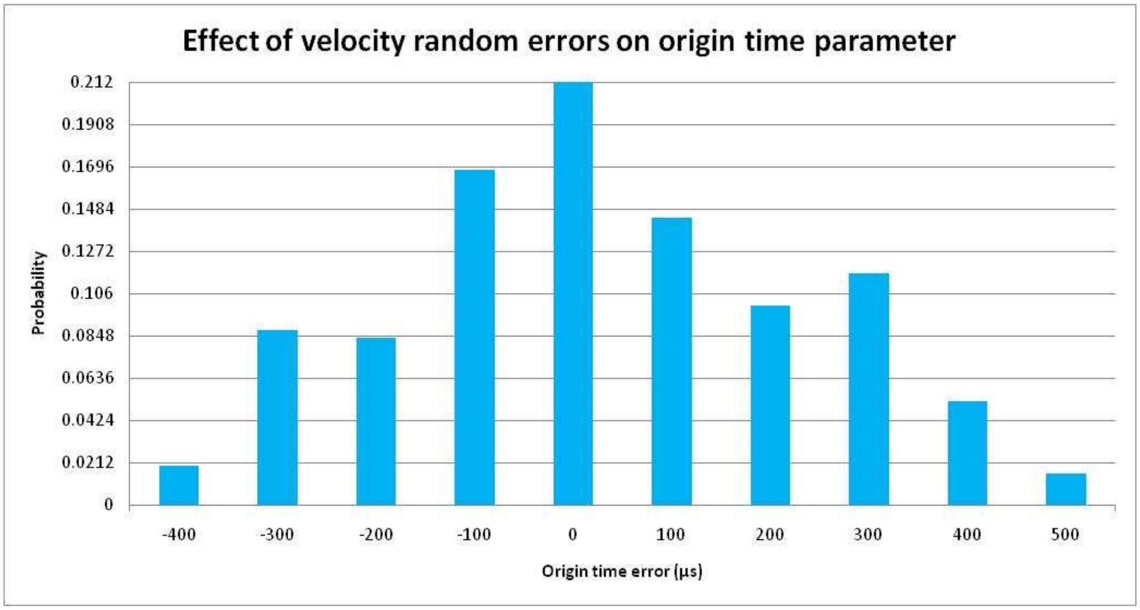


Figure (4.7): Histogram showing errors in origin time with their probabilities after introducing random errors to the velocity model.

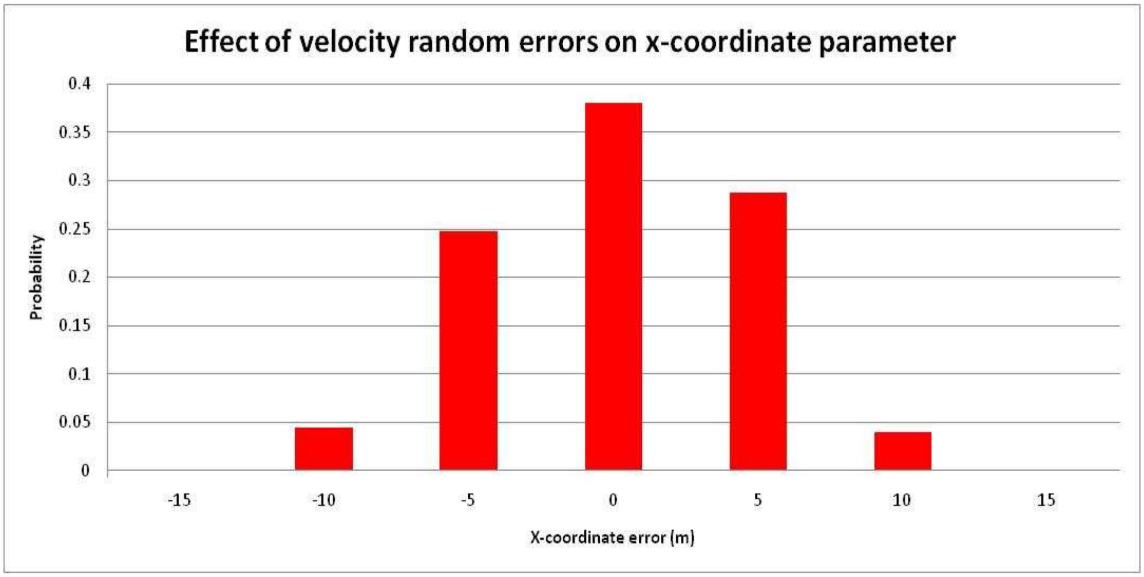


Figure (4.8): Histogram showing errors in hypocentre x-coordinate with their probabilities after introducing random errors to the velocity model.

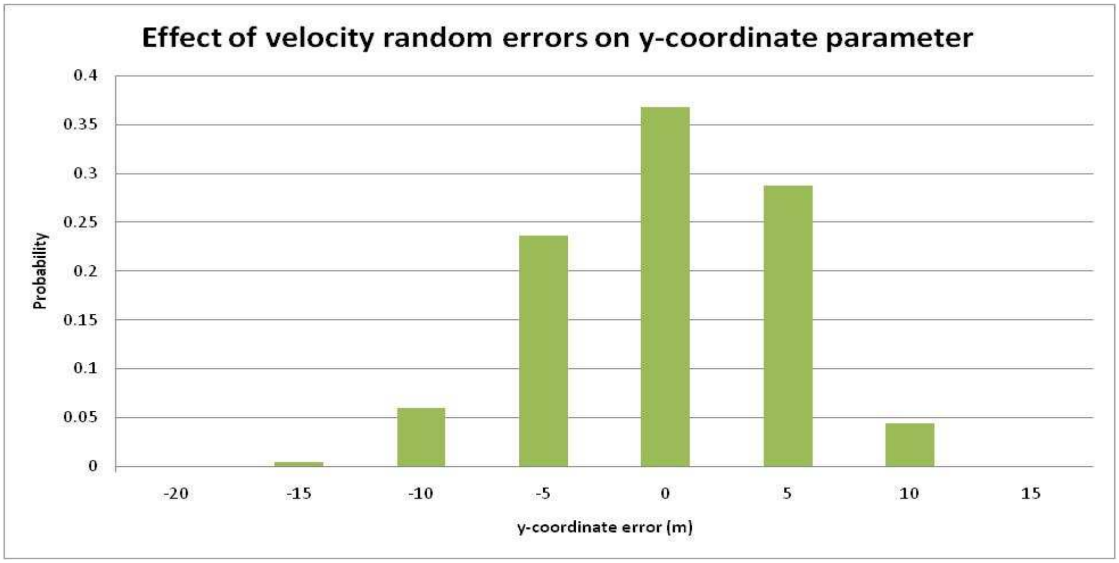


Figure (4.9): Histogram showing errors in hypocentre y-coordinate with their probabilities after introducing random errors to the velocity model.

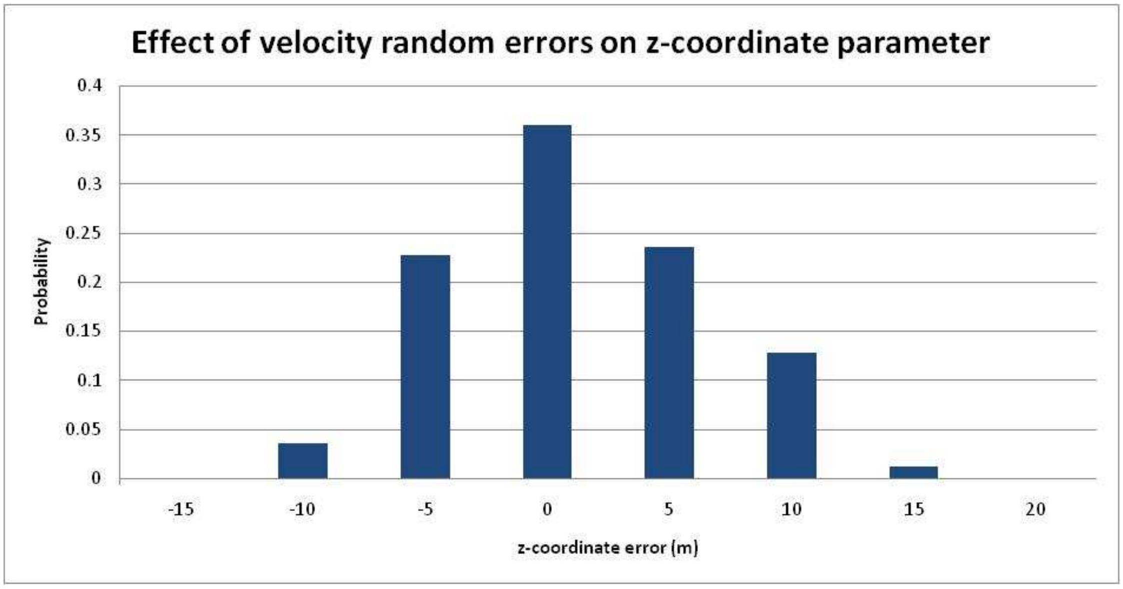


Figure (4.10): Histogram showing errors in hypocentre z-coordinate with their probabilities after introducing random errors to the velocity model.

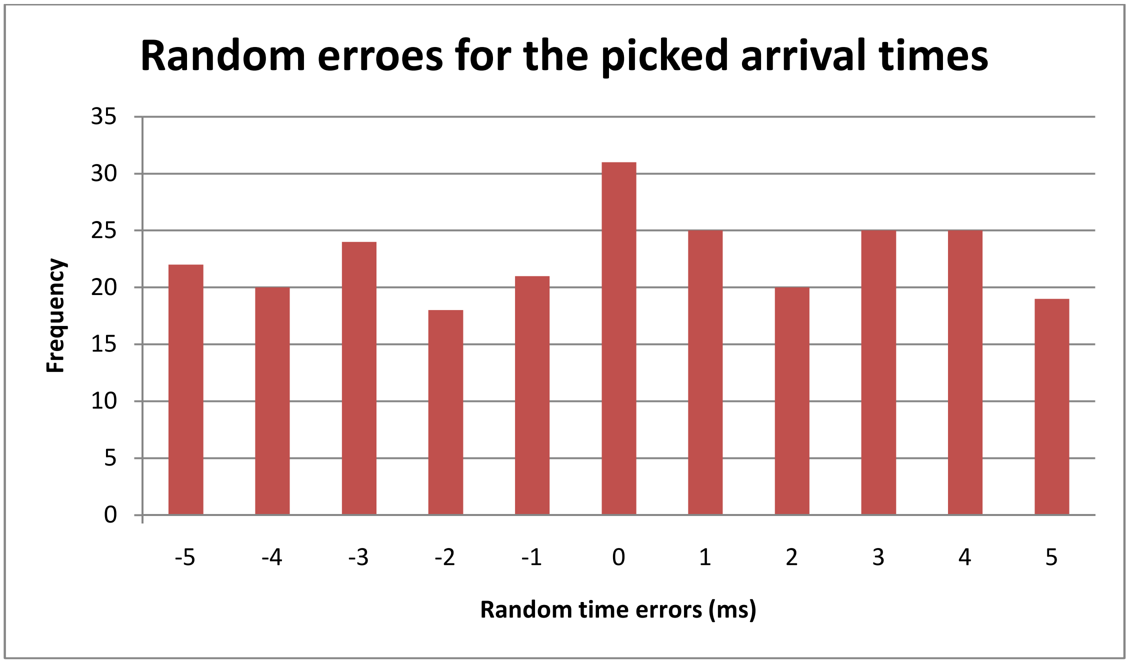


Figure (4.11): Simulation of ±5ms random errors for the picked arrival time related to the first receiver. The time random errors were generated 250 times.

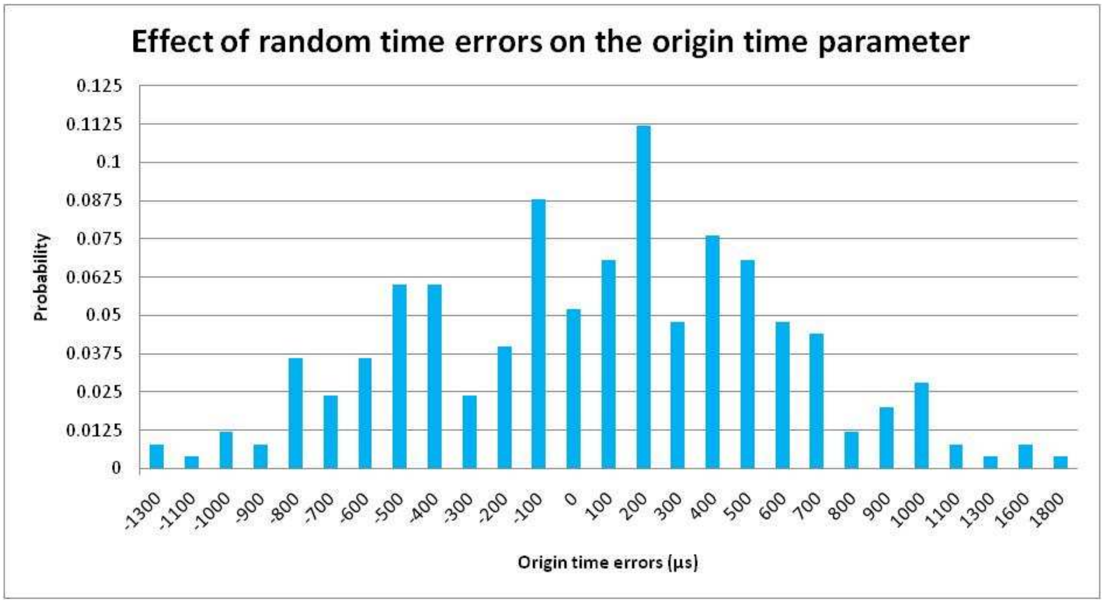


Figure (4.12): Histogram showing errors in origin time with their probabilities after introducing random errors to the picked arrival times.

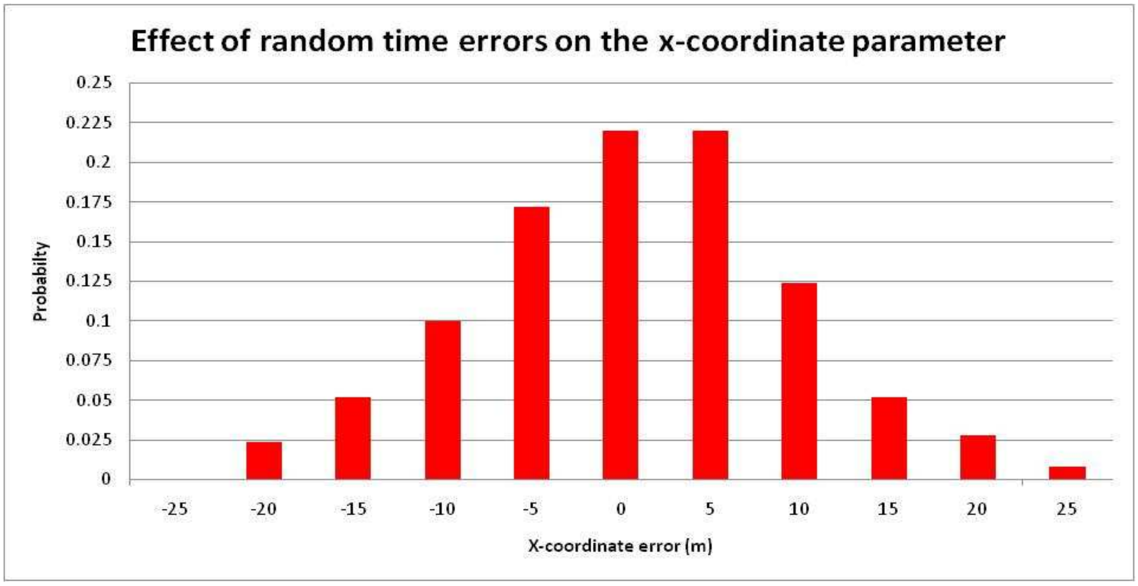


Figure (4.13): Histogram showing errors in hypocentre x-coordinate with their probabilities after introducing random errors to the picked arrival times.

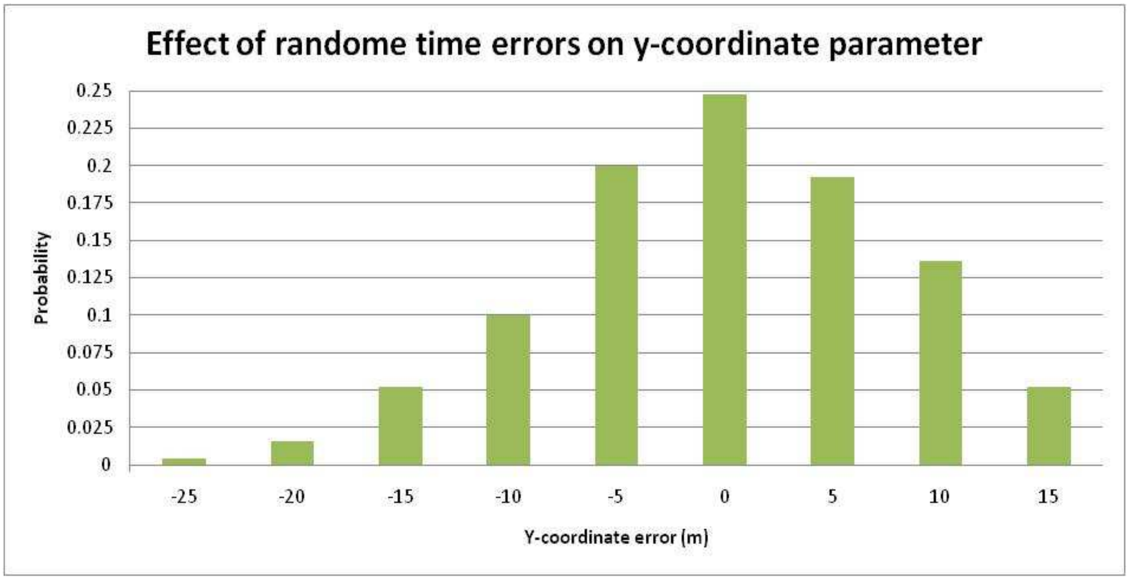


Figure (4.14): Histogram showing errors in hypocentre y-coordinate with their probabilities after introducing random errors to the picked arrival times.

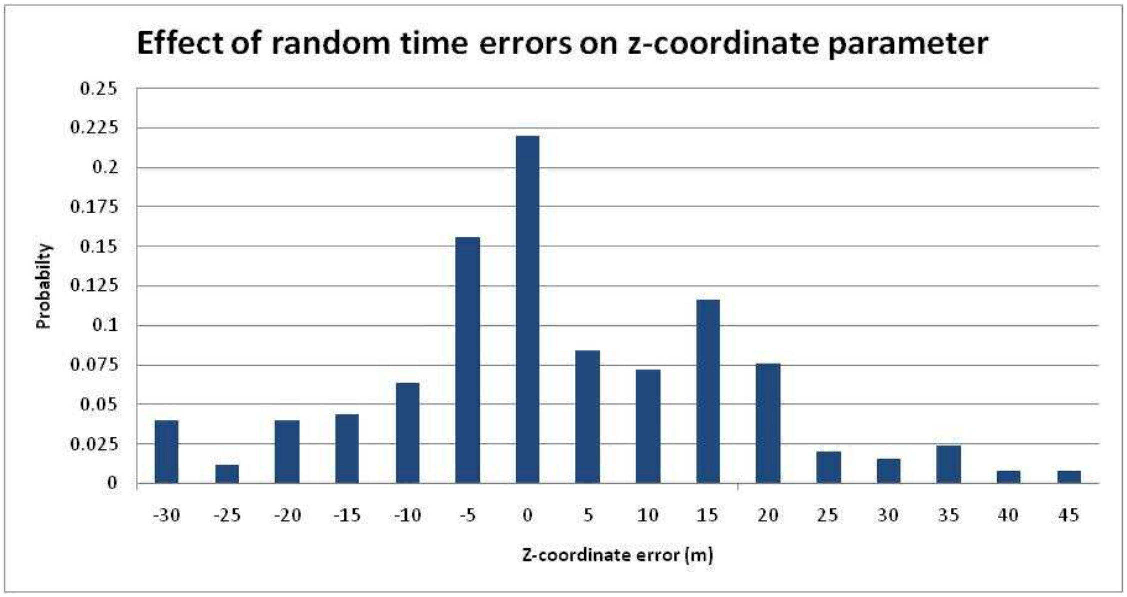


Figure (4.15): Histogram showing errors in hypocentre z-coordinate with their probabilities after introducing random errors to the picked arrival times.

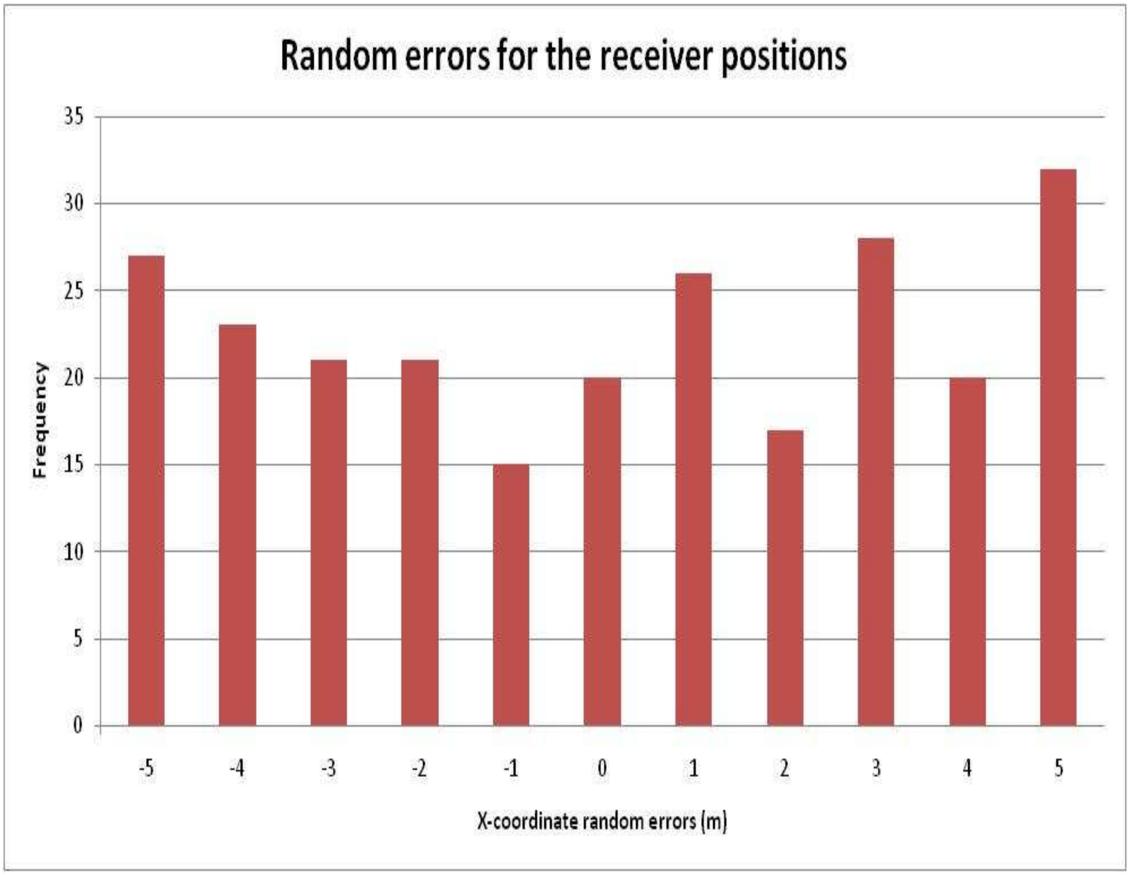


Figure (4.16): Simulation of ±5m random errors for the x-coordinate related to the first receiver. The position random errors were generated 250 times.

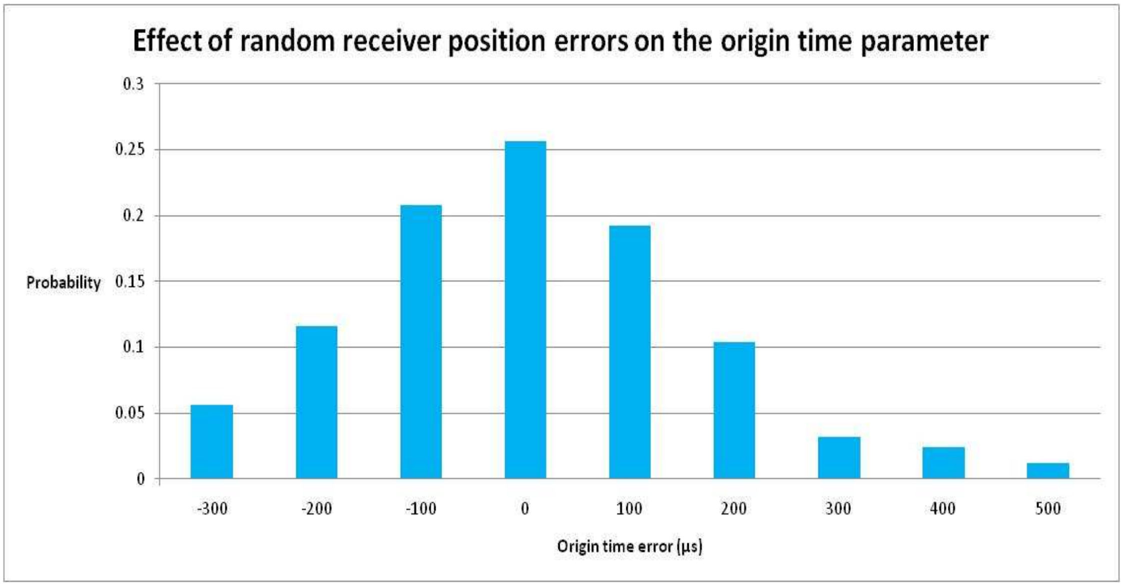


Figure (4.17): Histogram showing errors in origin time with their probabilities after introducing random errors to the receiver positions.

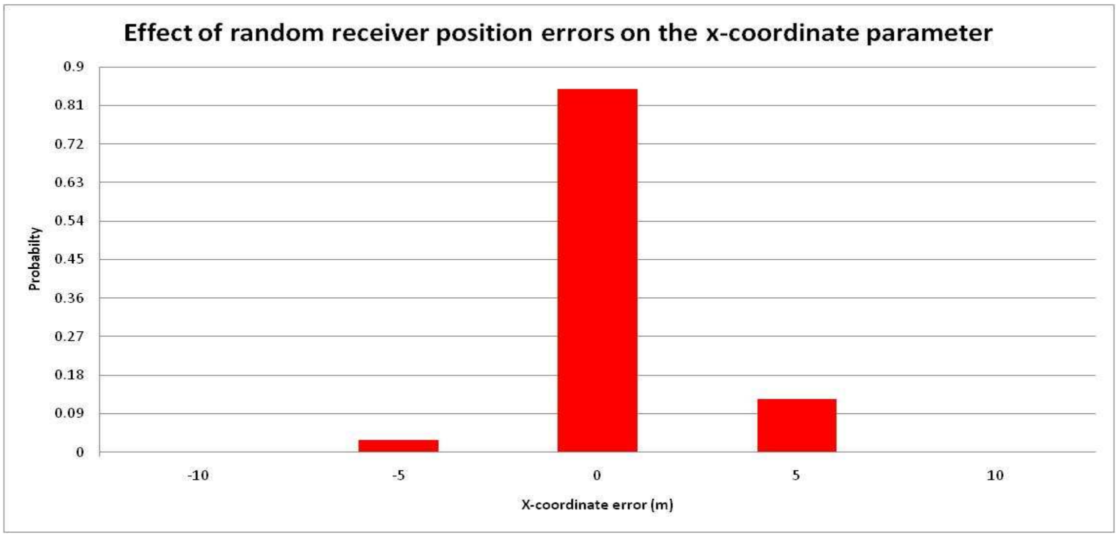


Figure (4.18): Histogram showing errors in hypocentre x-coordinate with their probabilities after introducing random errors to the receiver positions.

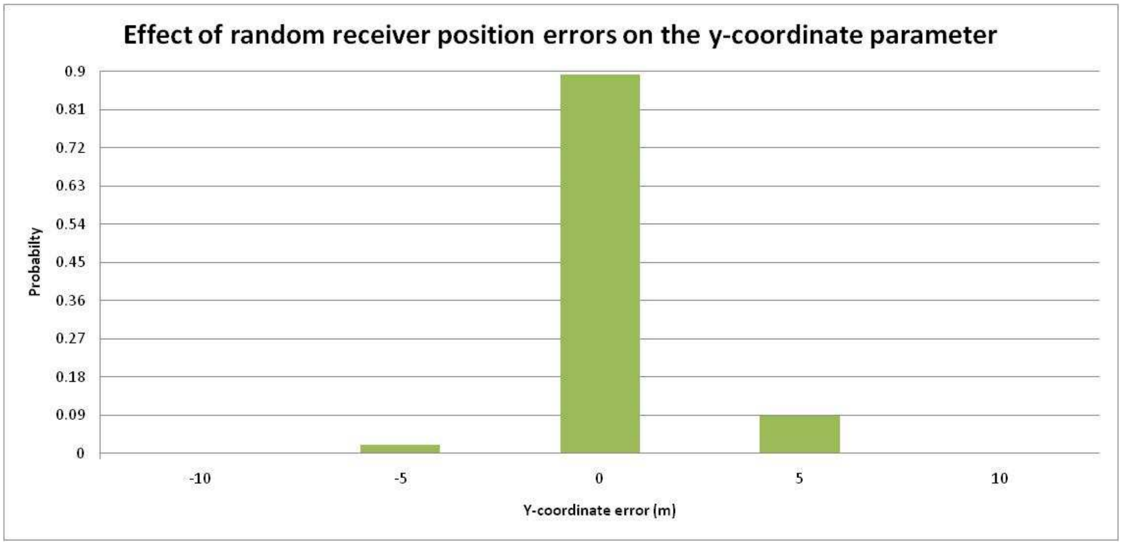


Figure (4.19): Histogram showing errors in hypocentre y-coordinate with their probabilities after introducing random errors to the receiver positions.

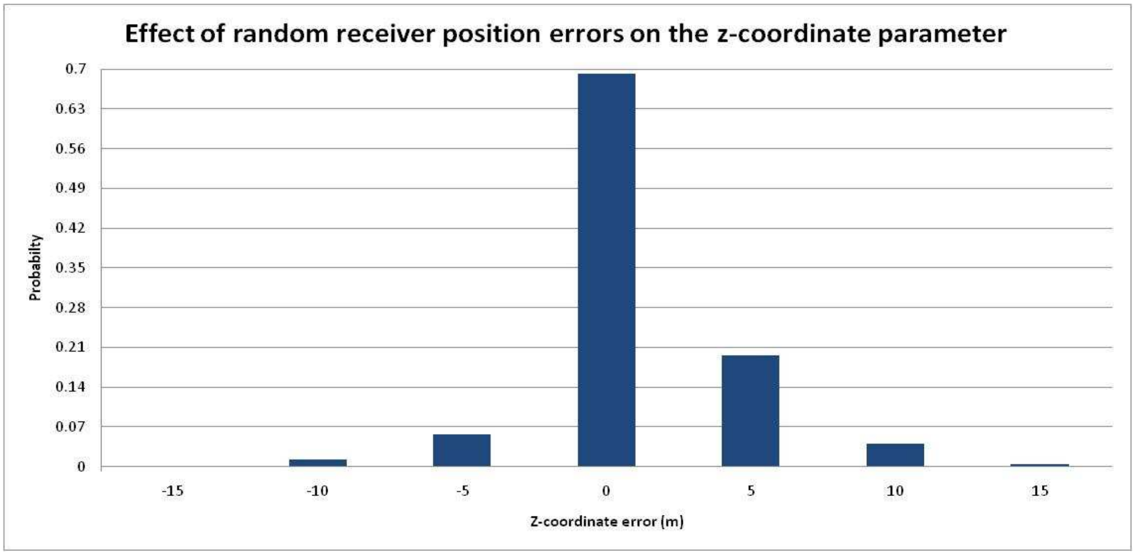


Figure (4.20): Histogram showing errors in hypocentre z-coordinate with their probabilities after introducing random errors to the receiver positions.

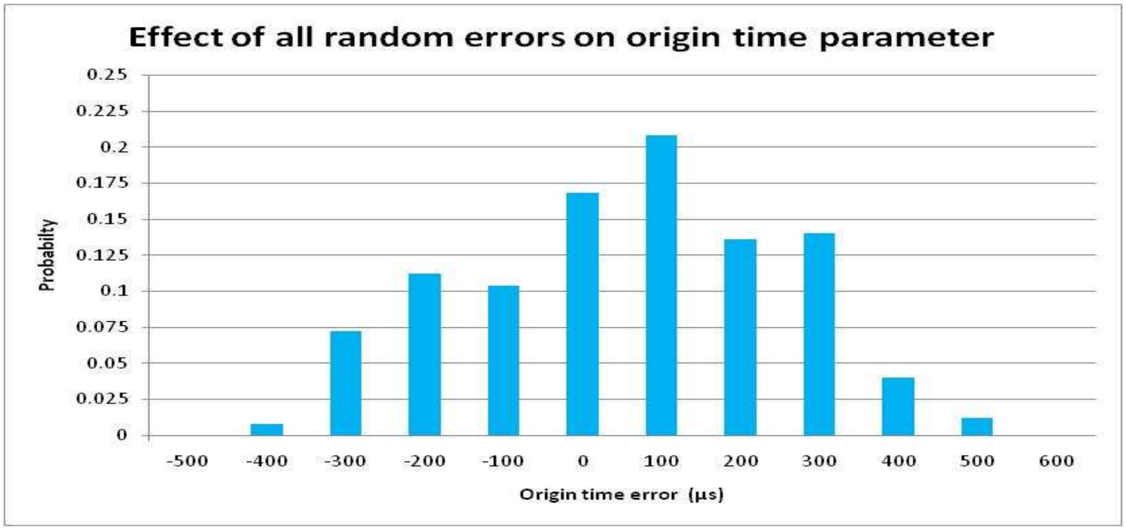


Figure (4.21): Histogram showing errors in origin time with their probabilities after introducing random errors to the 115 variables.

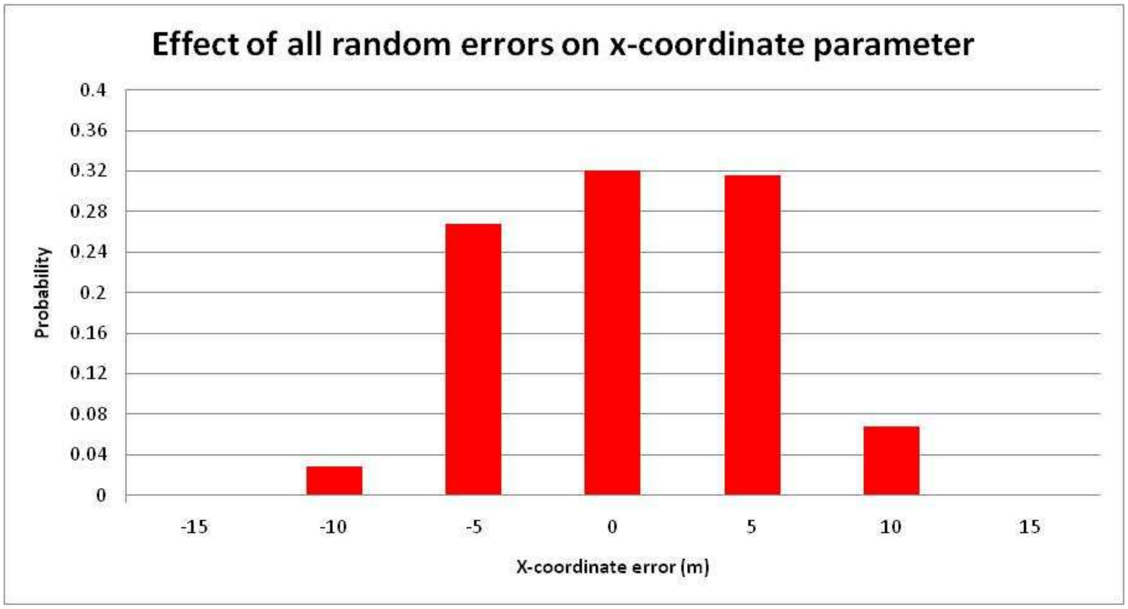


Figure (4.22): Histogram showing errors in hypocentre x-coordinate with their probabilities after introducing random errors to the 115 variables.

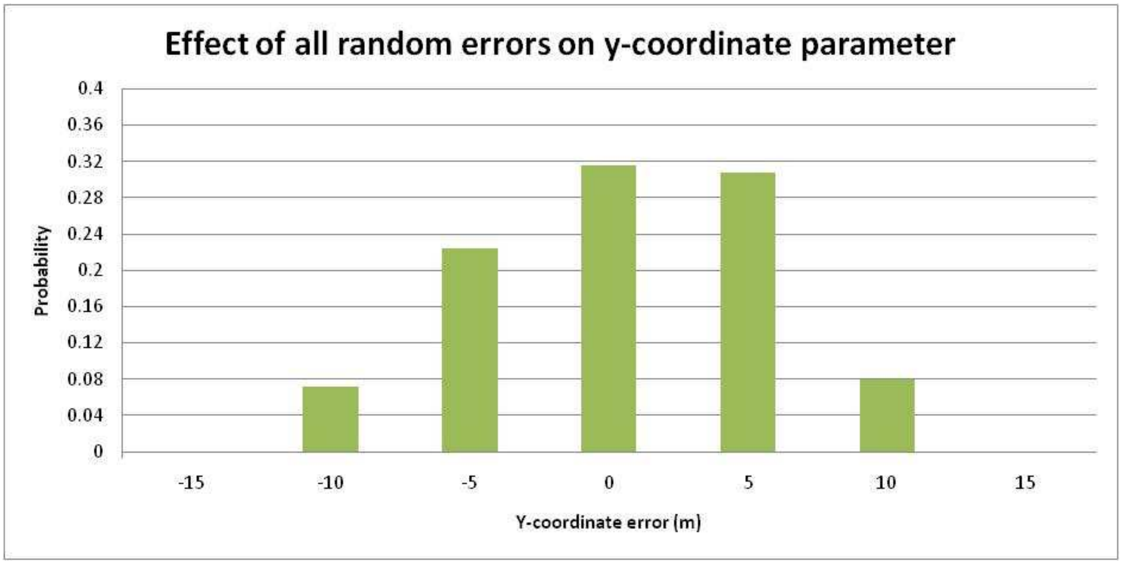


Figure (4.23): Histogram showing errors in hypocentre y-coordinate with their probabilities after introducing random errors to the 115 variables.

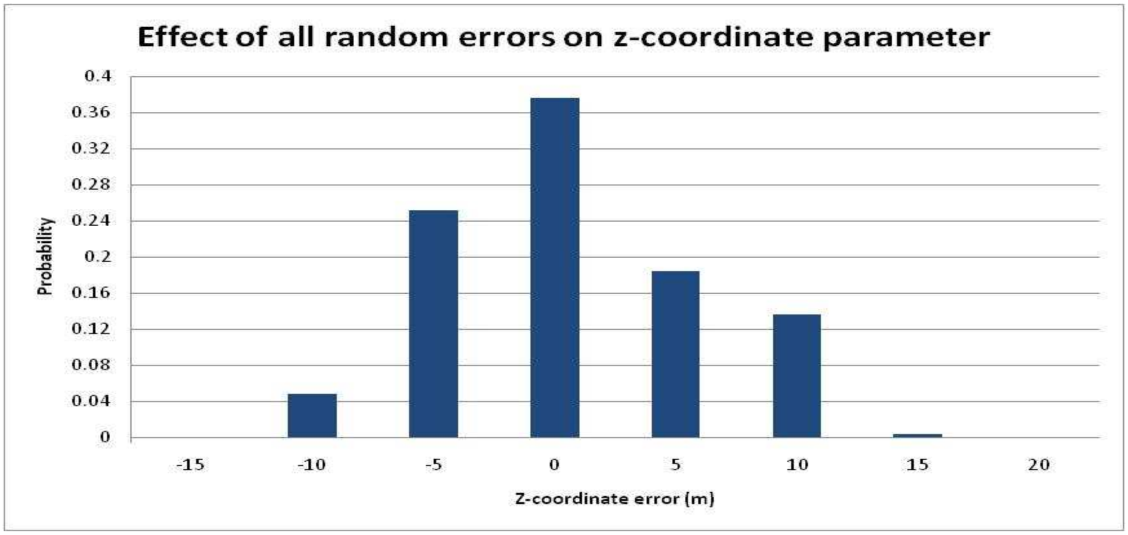


Figure (4.24): Histogram showing errors in hypocentre z-coordinate with their probabilities after introducing random errors to all 115 variables.

4.4 Conclusion

The use of induced micro-seismic events within a hydrocarbon reservoir for monitoring, characterising and/or imaging has recently attracted an increasing attention as an emerging technology. Such passive sources, however, lack information about both the positions and the timings. Therefore, they can be extremely useful if the origin time and event positions are accurately mapped. In this chapter, I have developed an algorithm for obtaining the hypocentre coordinates and the origin time of a micro-seismic event of a nonlinear and possibly multimodal objective function, constructed through the use of the l_1 norm. The algorithm seeks the global minimiser without considering the gradient. The Numerical simulations have shown that the computed position parameters of a micro-seismic event depend on the number and locations of the receiver stations, the velocity model and the picked arrival times. I have shown that the position parameters are accurately obtainable even if two monitoring wells are used within a 3D medium with as many as two receivers per well providing that velocity model, picked arrival times and receivers positions are not contaminated with errors. I have also shown that the algorithm is robust against random errors in the observable on derived parameters.

The motivation is to develop an efficient and robust algorithm needed for locating induced micro-seismic events within a hydrocarbon reservoir through a grid search optimisation. The accuracy of such algorithm is a direct function of the velocity model, grid size and receiver stations locations. Observed data, picked arrival times, is also equally important and related to the errors in hypocentre coordinates and origin time of a micro-seismic event. I have developed the algorithms as object oriented C++ codes within Microsoft Visual Studio platform.

References

Aldridge, D.F., 2000, Fast grid search algorithm for seismic source location: Sandia National Laboratories technical report SAND2000-1765.

Aldridge, D.F., L.C. Bartel, N.P. Symons, and N.R. Warpinski, 2003, Grid search algorithm for 3D seismic source location: Soc. Exploration Geophysicists Expanded Abstracts, **22**, 730-733.

Baker, T., R. Granat, and R.W. Clayton, 2005, Real-time earthquake location using Kirchhoff reconstruction Bulletin of the Seismological Society of America, **95**, 699–707.

Drew, J., R. White, and J. Wolfe, 2008, Microseismic event azimuth estimation: establishing a relationship between hodogram linearity and uncertainty in event azimuth: Presented at SEG Las Vegas 2008 Annual Meeting.

Gajewski, D., and E. Tessmer, 2005, Reverse modelling for seismic event characterization: Geophys. J. Int., **163**, no. 3, 276-284.

Lomax, A., A. Michielini, and A. Curtis, 2007, Earthquake Location, Direct, Global-Search Methods, *in* Meyers, B., ed., Springer.

McMechan, G.A., J.H. Luetgert, and W.D. Mooney, 1985, Imaging of earthquake sources in Long Valley Caldera, California, 1983: Bulletin of the Seismological Society of America, **75**, 1005–1020.

Michelini, A., and A. Lomax, 2004, The effect of velocity structure errors on double-difference earthquake location: Geophysical research letters, **31**, 1-4.

Moriya, H., K. Nagano, and H. Niitsuma, 1994, Precise source location of AE doublets by spectral matrix analysis of triaxial hodogram: Geophyiscs, **59**, no. 1, 36-45.

Nagano, K., H. Niitsuma, and N. Chubachi, 1989, Automatic algorithm for triaxial hodogram source location in downhole acoustic emission measurement: Geophysics, **54**, no. 4, 508-513.

Nelson, G.D., and J.E. Vidale, 1990, Earthquake locations by 3-D finite-difference travel times: Bulletin of the Seismological Society of, **80**, no. 2, 395-410.

Prugger, A.F., and D.J. Gendzwill, 1988, Microearthquake location: a nonlinear approach that makes use of a simplex stepping procedure: Bulletin of the Seismological Society of America, **78**, 799-815.

Rentsch, S., S. Buske, S. Lüth, and S.A. Shapiro, 2007, Fast location of seismicity: A migration-type approach with application to hydraulic-fracturing data: Geophyiscs, **72**, no. 1, S33-S40.

Ronde, A.A.d., D.P. Dobson, P.G. Meredith, and S.A. Boon, 2007, Three-dimensional location and waveform analysis of microseismicity in multi-anvil experiments: Geophys. J. Int., **29**, 1-13.

Ruzek, B., and M. Kvasnicka, 2001, Differential evolution algorithm in earthquake hypocenter location: Pure and Applied Geophysics, **158**, no. 4, 667-693.

Soma, N., H. Niitsuma, and R. Baria, 2007, Reflection imaging of deep reservoir structure based on three-dimensional hodogram analysis of multicomponent microseismic waveforms: Journal of Geophysical Reserach, **112**.

Vesnaver, A., L. Lovisa, and G. Böhm, 2008, Full 3D relocation of microseisms for reservoir monitoring: Presented at SEG Las Vegas 2008 Annual Meeting.

Waldhauser, F., and W.L. Ellsworth, 2000, A Double-Difference Earthquake Location Algorithm: Method and Application to the Northern Hayward Fault, California: Bulletin of the Seismological Society of America, **90**, no. 6, 1353-1368.

Warpinski, N.R., J.E. Uhl, and B.P. Engler, 1997, Review of Hydraulic Fracture Mapping Using Advanced Accelerometer-Based Receiver Systems: Presented at Natural gas conference

Zhang, H., and C. Thumber, 2006, Development and Applications of Double-difference Seismic Tomography: Pure and Applied Geophysics, **163**, 373-403.

Zhang, H., and C.H. Thurber, 2003, Double-Difference Tomography: The Method and Its Application to the Hayward Fault, California: Bulletin of the Seismological Society of America, **93**, no. 6, 1875-1889.

Every reasonable effort has been made to acknowledge the owners of copyright material. I would be pleased to hear from any copyright owner who has been omitted or incorrectly acknowledged.

Chapter 5

Ray-path and Slowness Model

5.1 Overview

The main topic of this chapter is the development of a technique for constructing a detailed slowness model for a hydrocarbon reservoir. First, I estimate the ray-path for each micro-seismic event in a three-dimensional heterogeneous medium with the help of the tools developed in Chapter 3. Given a 3D discretized medium and the coordinate positions for both the micro-seismic event and the recording stations, I determine the ray-path segments within every traversed cell. Second, I construct an objective function for each micro-seismic event by fitting an initial model response traveltimes to a finite set of observed arrival times through the use of the l_2 norm. Consequently, the algorithm builds an objective function for all events together with its gradient. I then use an iterative quasi-Newton method employing the BFGS updating formula for the construction of the slowness model. The algorithm produces the inverse of the Hessian matrix as a by product as quality control on the solution. The method is very robust for obtaining the global minimiser when the initial model is close to the true solution.

5.2 Introduction

Constructing a slowness model of a medium from a finite set of observed arrival times is an essential problem in seismic traveltime inversion (seismic tomography). One needs to know the position coordinates for the sources and the receiver stations in addition to the ray-paths between each source-receiver pair. This problem is a nonlinear inverse problem as the rays depend on the slowness model. Further, because the true model is a piecewise continuous function of location and the measured data is finite and contaminated with noise, the problem is ill-posed and possibly ill-conditioned (Snieder, 1998).

Different approaches are used for the reconstruction of the slowness model of this nonlinear problem. One approach is to use perturbation theory; however, this approach assumes that both the forward and inverse operators are regular (Snieder, 1990). This approach is based on local perturbation theory (Trampert, 1998). When the nonlinearity is strong, one needs to use many terms to accurately reconstruct the model, hence making the approach unmanageable. This method works well for weakly nonlinear problems.

Another approach is to use iterative gradient based optimisations (Menke, 1984; Tarantola, 2005). These methods are generally robust in locating a local minimiser of the objective function, and they depend on the initial model. This problem can be fixed by starting the method at many initial solutions; which may be expensive. Still, they require the computation of the gradient at each iteration unless the problem is weakly nonlinear. In their favour, such methods allow prior information and error information to be incorporated naturally into the objective function (Tarantola, 2005).

A third approach is to use global optimisations (Sen and Stoffa, 1995). These methods are simple to implement and capable eventually of locating the global minimiser, but at an exponential cost (Sambridge, 1998; Trampert, 1998). This makes the approach impractical for large size problem such as tomography.

The objective of developing an algorithm for induced micro-seismic events is the opportunity to construct a detailed heterogeneous slowness model for a hydrocarbon reservoir by comparing computed traveltimes with the observed arrival times. This is essential for predicting accurate positions coordinates and origin times of the induced micro-seismic events.

One difficulty is that the problem is nonlinear and thus the objective function may have multiple minima, so we may require using the global optimisation approach which can be exceptionally expensive. We also may need to include the prior and error information into the objective function and this stops us from using the perturbative approach. Gradient based methods are comparatively fast and can lead to the global minimiser when we have a good starting model; which is the case for our problem. They are attractive for large scale problems because they require less memory, particularly when the forward operator is sparse, as is the case for traveltime inversion. Obtaining the resolution operator is another difficulty we are faced with when dealing with the nonlinear problem. Still, we can overcome this problem by using either the conjugate gradient method or the variable metric method with the BFGS updating formula which has many other favourable features as stated in Chapter 2 (Berryman, 1990; Berryman, 2000a, b; Sen and Stoffa, 1995). Using the variable metric method would give a better chance of obtaining superior results (Ravindran et al., 2006).

5.3 Theory and Methodology

The transmission arrival time of a passive micro-seismic body-wave recorded at a particular buried sensor resulting from the production activities within a hydrocarbon reservoir is expressed using the Fermat's principle as a path integral,

$$T^{obs} = T^{0} + \int_{\Gamma} p\left(x, y, z\right) dl, \tag{5.3.1}$$

where T^{obs} represents the arrival time of an event originated at a micro-seismic source position and recorded at a receiver station with known coordinates and T^{0} is the time at which the micro-seismic event occurred. The ray-path Γ represents the path that minimises the traveltime with respect to all paths between the source position and the receiver station, the slowness of the medium is denoted by $p(x, y, z)$, and dl is an infinitesimal segment of the ray-path length as shown in Figure (5.1). The unknowns in equation (5.3.1) that we need to determine are the velocity field (slowness field) and the ray-path using the recorded arrival times. Furthermore, the relationship between the recorded arrival time and the slowness function is nonlinear as the ray-path depends on the slowness field.

In section 3.7, I presented a reliable technique for tracing the path of a ray starting at a receiver station position and terminating at the micro-seismic event position. The method computes the local gradient of the traveltime function at the interfaces and follows the steepest descent direction. While tracing, the algorithm computes the ray-path segment lengths within each cube traversed by the ray as shown in Figure (5.2). I also showed that traveltimes are accurately determined using the ray-path estimation. Thus, we can replace traveltime with ray-path segments to build the objective function.

Let us assume that we have a micro-seismic event m together with its position coordinates $\{x_m^s, y_m^s, z_m^s\}$ and origin time T_m^0. I will relax this assumption in Chapter 6. In other words, I will deal with the general problem such that the source coordinates, the origin time, the ray-path, and the slowness model are all unknown. Still, when the source position coordinates and origin time are not available, I use the method I developed in Chapter 4 to obtain such parameters. Using the tracing technique described in Chapter 3, the traveltime that starts at source m and is recorded at receiver station n with known coordinates $\{x_n^r, y_n^r, z_n^r\}$, is given as,

$$T_{mn}^c = \sum_{k=1}^{N_c} l_{mnk} \, p_k \qquad (5.3.2)$$

This equation was derived in Chapter 3 and it represents a linearized relationship for the computed traveltime T_{mn}^c in terms of p_k which denotes the slowness model parameter within cell k and l_{mnk} representing the ray-path segment length spent within cell k such that N_c is the number of cells within the considered model. This indexing is well defined for any cube within the 3D model and agrees with our discretization described in Chapter 3.

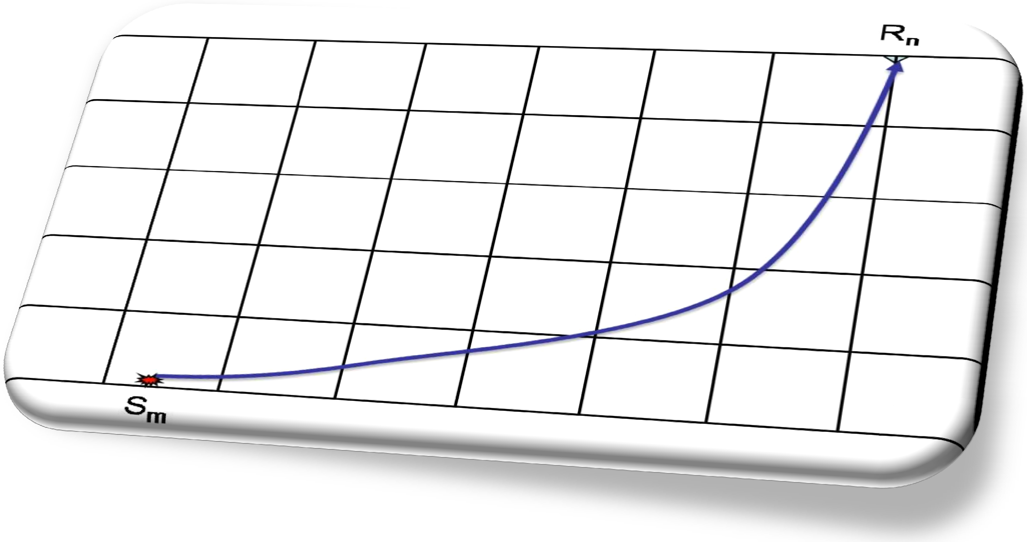


Figure (5.1): A continuous path of a ray which was originated at the source position S_m and travelled through the medium before being recorded at the receiver station R_m.

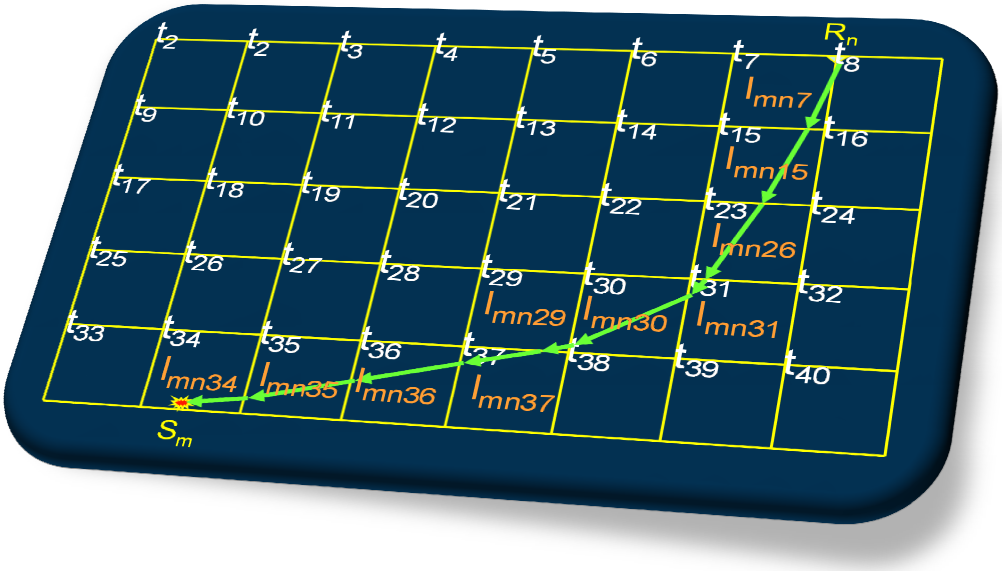


Figure (5.2): A back traced ray-path in 3D medium starting at the receiver station R_m and ending at the source location S_m with the corresponding ray segments l_{mnk} within each traversed cell.

It is worth mentioning that a ray, in general, passes through few cells as shown in Figure (5.2). The untraced cells in the model have no contribution to equation (5.3.2).

The traveltime in equation (5.3.2) is calculated using some initial slowness model with the help of the method developed in section 3.7; consequently, the ray-path is only an approximation to the true one. Since we are dealing with a nonlinear problem, the objective function may have more than one local minimum. Therefore, it is crucial for such a problem that we have a good starting model in order to obtain the global minimum representing the true solution. In our case, we are dealing with a hydrocarbon reservoir that, we expect, has gone through different probes leading to a good velocity model to start with.

To retrieve the slowness model accurately, there have to be many receiver stations well positioned within the hydrocarbon reservoir in order to obtain a wide aperture. The coordinates of such receiver stations, also, need be determined with high accuracy to reduce the errors in the reconstructed slowness model (Lehmann, 2007). The same applies for the sources when the situation allows, as is the case when we are probing the subsurface with vertical seismic profiling (VSP). Equally important are the picked arrival times.

Following the discussion in section 2.3, the objective function for the above problem expressed in l_2 norm is given as least squares,

$$f_{mn}\left(p\right)=\frac{1}{2}(T^{o}_{mn}-L(p))^{t}(T^{o}_{mn}-L(p)), \tag{5.3.3}$$

where $L(p)$ represents the nonlinear forward modelling operator. With the help of linearization, we replace the nonlinear operator $L(p)$ in equation (5.3.3) with its linearized form represented by equation (5.3.2) to obtain,

$$f_{mn}(p)=\frac{1}{2}(T_{mn}^{o}-\sum_{k=1}^{N_c} l_{mnk}\,p_k)^t\,(T_{mn}^{o}-\sum_{k=1}^{N_c} l_{mnk}\,p_k). \qquad (5.3.4)$$

Still, one can constrain the objective function in (5.3.4) according to section 2.3. Such constraining is sometimes necessary, especially when the problem is under-determined or uneven as described in Chapter 2. The reconstruction depends on how accurately the observed data T_{mn}^{o} is known. That includes the measuring precision and the errors estimation. When the observed data is accurate, the reconstructed model needs to fit the measured data precisely. On the other hand, the determined slowness model needs to explain the recorded data approximately when the observed data is imprecise.

Let us assume that we have a set of $N_s N_r$ recorded arrival times resulted from N_s sources and recorded at N_r receiver stations. An induced micro-seismic event is initiated as a point source, and then the generated seismic wave propagates through the medium and is later recorded at various receiver stations at different arrival times. The sources and receivers can be anywhere within the vicinity of the reservoir, but they must be close to each other to be able to detect low amplitude waves. This configuration will lead to as many as $N_s N_r$ objective functions such that each source-receiver pair has a unique function as described in the above paragraphs. The objective function for the system is the summation of all such objective functions,

$$F(p)=\frac{1}{2}\sum_{m=1}^{N_s}\sum_{n=1}^{N_r}[(T_{mn}^{o}-\sum_{k=1}^{N_c} l_{mnk}\,p_k)^t\,(T_{mn}^{o}-\sum_{k=1}^{N_c} l_{mnk}\,p_k)]. \qquad (5.3.5)$$

In generating the objective function, the forward modelling algorithm is executed once for each source point to generate traveltimes T_{mn}^{c} at all receiver stations while the ray tracing algorithm is executed N_r times to compute the ray-path components for each source-receiver pair. Collectively, one needs to sequentially execute the forward modelling program N_s times and the ray tracing algorithm $N_s N_r$ times. In computing, this process can be parallelised in a straightforward way. The gradient of the objective function (5.3.5) is given as,

$$g = \nabla F(p) = \frac{1}{2}\sum_{m=1}^{N_s}\sum_{n=1}^{N_r}\frac{\partial}{\partial p_j}(T_{mn}^{o} - \sum_{k=1}^{N_c} l_{mnk}\, p_k)^t (T_{mn}^{o} - \sum_{k=1}^{N_c} l_{mnk}\, p_k). \qquad (5.3.6)$$

The gradient in (5.3.6) can be further simplified as,

$$g_j = -\sum_{m=1}^{N_s}\sum_{n=1}^{N_r} l_{mnj}\,(T_{mn}^{o} - \sum_{k=1}^{N_c} l_{mnk}\, p_k). \qquad (5.3.7)$$

Basically, the objective function and the gradient depend on the multiplication of two entities, the ray-path components and the slowness parameters. I approach the problem in two phases, the construction phase and the optimisation phase. During the construction process, the slowness parameters (slowness model) are kept fixed while the ray-path parameters are being generated. During the optimisation process, on the other hand, the ray-path parameters are held fixed while the slowness parameters are optimised according to the gradient, using the variable metric method with the BFGS updating formula. The construction stage demands more computations because it involves the forward modelling algorithm. Therefore, I direct more resources to the optimisation process, hoping to minimise the interaction with the forward modelling algorithm and thus reducing the computations.

I will use the two equations (5.3.5) and (5.3.7) with the variable metric method deploying the BFGS updating formula as well as an initial model to reconstruct a sequence of slowness models that, hopefully, will converge to the true model. The approach must not allow the generation of the solution in a single iteration unless the problem is linear. In the next section, I will detail the algorithm that generates the objective function together with its gradient. Later, I will describe the algorithm for the variable metric method using the BFGS updating formula.

5.4 Objective Function and Gradient Construction Algorithm

All the node points are populated with their corresponding initial slowness parameters as shown in Figure (5.3). Starting at a source position S_n, traveltimes are computed at all nodes using the algorithm of section 3.7 as shown in Figure (5.4). Traveltimes at all receiver stations relevant to the current source are obtained if receivers are located on node points; otherwise, a trilinear interpolation, as described in Chapter 3, is used to obtain the traveltimes for such receivers. The computed traveltimes are subtracted from their corresponding observed arrival times at each receiver station for the present source.

Starting at a receiver station R_n , the components of the ray-path between source S_m and receiver R_n are calculated using the algorithm described in section 3.8 as shown in Figure (5.2) and the corresponding objective function is obtained according to equation (5.3.4). This process is repeated for all remaining receivers attached to the current source to generate as many as N_r objective functions. The above procedure is repeated for all remaining sources. Subsequently, the objective function is constructed according to the formulation in (5.3.5). In constructing the objective function, the algorithm retains a unique indexing to every component of the ray-path. The indexing includes the source number, the receiver number, cell number and bed number together with the corresponding slowness value. A bed is a horizontal sheet with a thickness of one cell. Other entities, such as the prior information and/or weights, can be included easily. Ray-path components are multiplied with their corresponding slownesses to finish the construction process of the objective function.

The construction of the gradient is accomplished through three steps. First, individual objective functions are differentiated separately term by term. Second, a quick sort is applied to the derivatives to regroup them according to the cell index. Third, all the derivatives that belong to the same bed (cell) are added to generate a single gradient element attached to such bed. This ensures that each traced bed has its own gradient

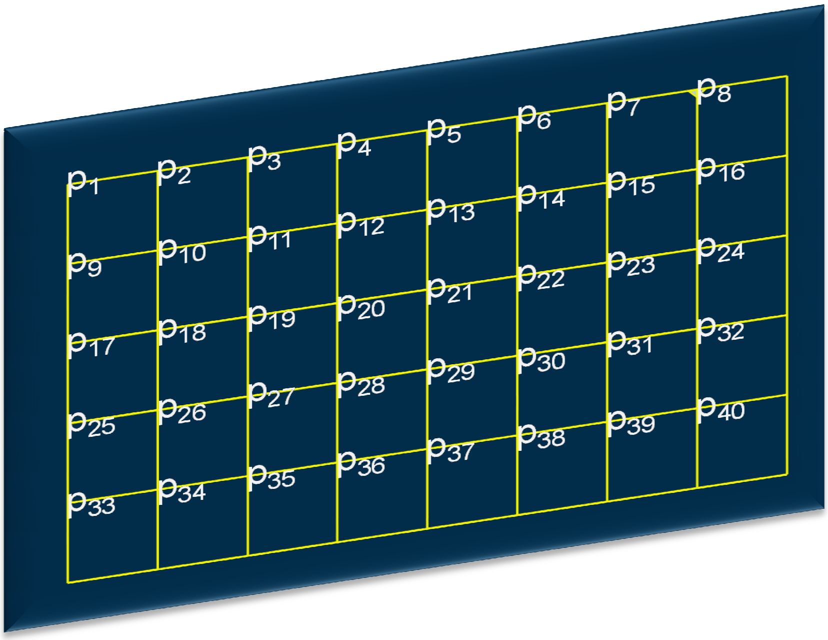


Figure (5.3): A vertical slice of the 3D medium with the nodes populated with their corresponding current slowness model parameters.

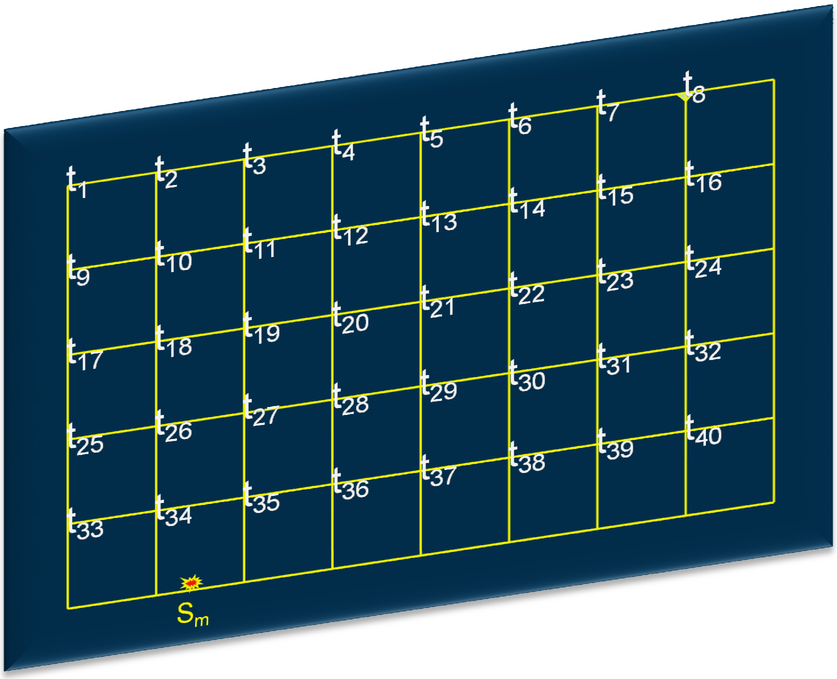


Figure (5.4): A vertical slice of the 3D medium with the nodes populated with their corresponding computed traveltimes according to the current slowness model.

element. This process is not exclusive to horizontal beddings. It can accommodate all kinds of structure formulations. Because the algorithm is cell based, it can be modified to fit any geological structure. In building the gradient, there is only one occasion where multiplication has to be performed. Still, the algorithm depends heavily on the indexing system used to construct the objective function.

5.5 Optimisation Algorithm

Starting with the objective function and its gradient as well as the current slowness model parameters, the aim is to reconstruct improved slowness model parameters through optimisation. I use the variable metric method because it has many important features as described in Chapter 2. In particular, I use the BFGS formula for updating the Hessian matrix inverse because of its superiority and robustness. One remarkable feature of the method is its fast convergence rate. The algorithm starts as the steepest descent method in first step, but in a few iterations it modifies itself aggressively towards the Newton method. When the number of estimated slowness parameters is large, the method may have a problem allocating the required memory for the inverse of the Hessian matrix. This may not be a problem in the future with the advent of new generation computers in addition to cheap memory.

The algorithm starts evaluating the function and the gradient components at the initial model parameters. The gradient is a column vector with n components where n is the number of the slowness model parameters. The initial estimation of the Hessian inverse is set to the identity matrix. Therefore, the initial descent direction is a column vector such that its components are the steepest descent directions. The curvature is computed according to the positive definite property,

$$\boldsymbol{g}^t H \boldsymbol{g} \geq 0, \tag{5.4.1}$$

where H is the Hessian inverse and $\boldsymbol{g}$ is the gradient. But, we have already computed the descent direction which is given by,

$$\boldsymbol{s} = -H\boldsymbol{g}. \tag{5.4.2}$$

Thus, the curvature is actually calculated as,

$$-\boldsymbol{g}^t \boldsymbol{s} \geq 0. \tag{5.4.3}$$

The curvature is an indication of how positive definite is the Hessian inverse. It can also be used to test and decide when to terminate the algorithm.

The initial model parameters update is given as,

$$\boldsymbol{p}_{k+1} = \boldsymbol{p}_k + \alpha_k \boldsymbol{s}_k. \tag{5.4.4}$$

Up to this point, I have obtained only the descent direction $\boldsymbol{s}$ and of course the current model $\boldsymbol{p}_k$ is given. So, I am left with the step length parameter α_k to optimise. This is obtained by the use of a line search method as described in Chapter 2. The algorithm has a number of line search methods involving the golden section search method, the quadratic and cubic curve fitting as described in section 2.4.1. Each one of these methods has two stages: the bracketing stage and the sectioning stage. During the bracketing stage, we aim to bracket the step length parameter within an interval as described in section 2.4.3. The starting interval is set to $[0,1]$ with $c_1 = 0.01$ and $\alpha_1 = 1$. The sectioning stage, as described in section 2.4.3, is designed to iteratively optimise the step length parameter by interpolation. During this stage, we have two points representing the end points of the bracketing interval with their function valuations. Also, we have the gradient at one point or the two points. When the gradient is known at the two points, we use the cubic interpolator otherwise the quadratic interpolator is used. At the end of the sectioning stage, an optimised step length parameter is obtained. Consequently, the slowness model parameters are updated according to equation (5.4.4).

The objective function and the gradient are re-evaluated using the new slowness model parameters. The stopping criteria as described in section 2.4.4 are checked to determine if the new model is accurate enough to terminate the algorithm. I have come to the conclusion that between 3 to 5 iterations are needed to induce enough updates before returning to the construction phase to update the ray-paths components through ray tracing. Such updates are needed to reconstruct a new objective function together with its new gradient.

The current as well as the new slowness model parameters and the current with the new gradient components are used to update the Hessian inverse according to the BFGS formula as given by equation (2.4.55). The algorithm iteratively continues in the same manner until it encounters a stopping criterion, as described in section 2.4.4.

5.6 Numerical Results

I used a layer cake velocity model to test the algorithm described above. The velocity model consists of seven layers with different velocities and thicknesses as listed in Table (5.1). The velocity model contains inversions between the third and fourth layers and between the fifth and sixth layers. There are three monitoring wells, each with fifteen receivers labelled as blue circles with a 25m vertical separation and positioned at depth between 100m and 450m as shown in Figure (4.2). The wells occupied three corners of cuboidal model while the fourth corner is saved for an imaginary injecting/extraction well. I used eight micro-seismic events located within a 25m-layer at a depth between 430m and 450m near the injecting/extraction well as listed in Table (4.2). Such geometry could represent a recording setup used as a permanent monitoring of a producing hydrocarbon reservoir or a CO_2 sequestration reservoir. The arrival times for the eight seismic events are calculated as a P-wave started at each source position and recorded at the receivers' locations using a ray tracing algorithm resulting in 360 observed data points (Nadri, 2008). Figure (5.5) has eight curves with different colours for the eight different micro-seismic events. Each curve represents the arrival times at the geophones located in monitoring well 3. The curves for the other monitoring wells have similar trends. Therefore, they are not shown here.

Layer	Velocity (m/s)	Thickness(m)
1	3500	75
2	3650	125
3	4000	50
4	3700	75
5	3800	100
6	3400	25
7	4000	50

Table (5.1): Velocity model consists of seven layers.

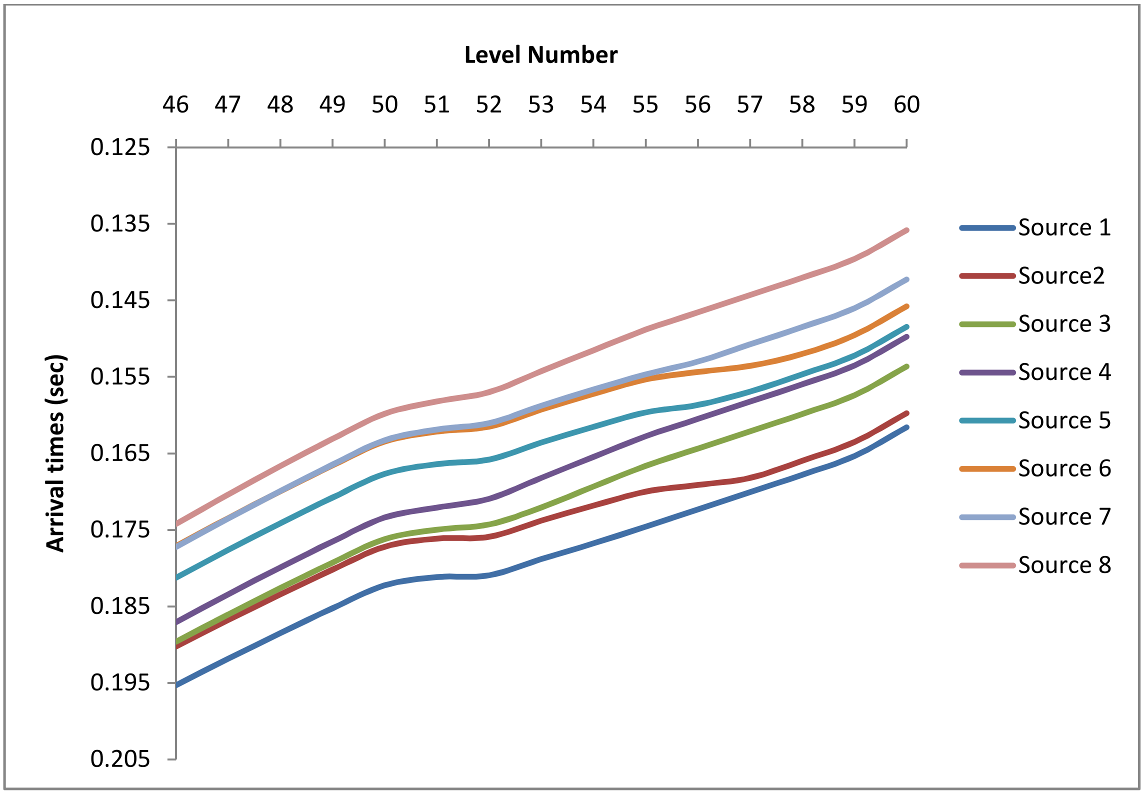


Figure (5.5): Arrival times generated for the true model at fifteen levels within monitoring well 3. Level 46 is the shallowest at a depth of 100m. The levels are incremented at 25m.

We assume that the positions coordinates for the sources as well as the receiver stations and the origin times are known. The initial velocity model parameters are set to 3000. Because of the geometry setting for the sources positions and the receiver stations, the first layer, part of the second layer, part of the sixth layer and the last layer are not updated. A grid size of 5m is used for constructing the traveltimes according to the algorithm, described in section 3.7. Figure (5.6) shows traveltimes computed for the initial model at fifteen levels within monitoring well 3 for the eight sources. Level 46 is the shallowest at a depth of 100m. The following levels are incremented at 25m. The ray tracing algorithm is based on the same setting for computing the ray-paths components for each source-receiver pair. Figure (5.7), on the other hand, shows the ray-paths lengths computed for the initial model using ray tracing at fifteen levels within monitoring well 3 for all sources. Differences between computed traveltimes using forward modelling algorithm and calculated traveltimes using equation (5.3.2) for the initial model at fifteen levels within monitoring well 3 are shown in Figure (5.8). This is to confirm the robustness of the algorithm. In addition, differences between generated arrival times for the true model and computed traveltimes for the initial model at the fifteen levels within monitoring well 3 are presented in Figure (5.9).

The construction of the objective function and the gradient are based on horizontal beddings with a constant thickness of 5m, a height of one cell. The default setting is cell based. However, this will result in poorly conditioned problem because it will result in an uneven sampling of the medium by induced elastic waves. Thus, some cells of the medium are sufficiently sampled while other cells remain severely under sampled. The gradient components at different beds are computed for the initial model and are shown in Figure (5.10). Figure (5.11), on the other hand, shows the number of rays passing different beds for the initial model and have similar trend to the gradient.

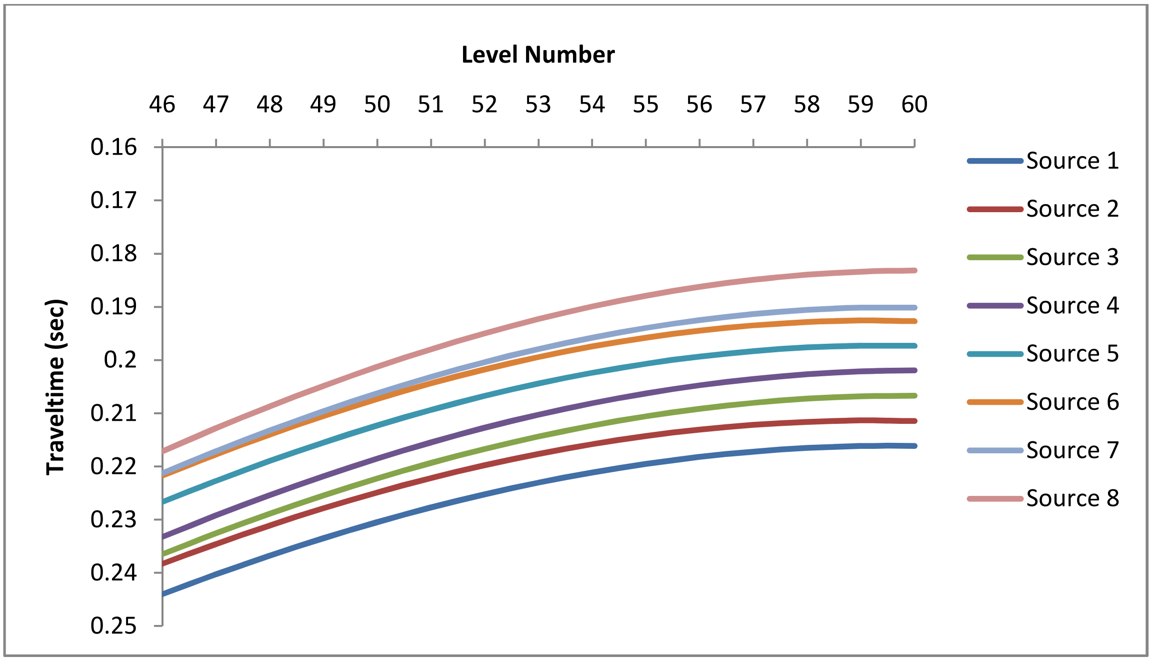


Figure (5.6): Traveltimes computed for the initial model at 15 levels within monitoring well 3. Level 46 is the shallowest at a depth of 100m. The levels are incremented at 25m.

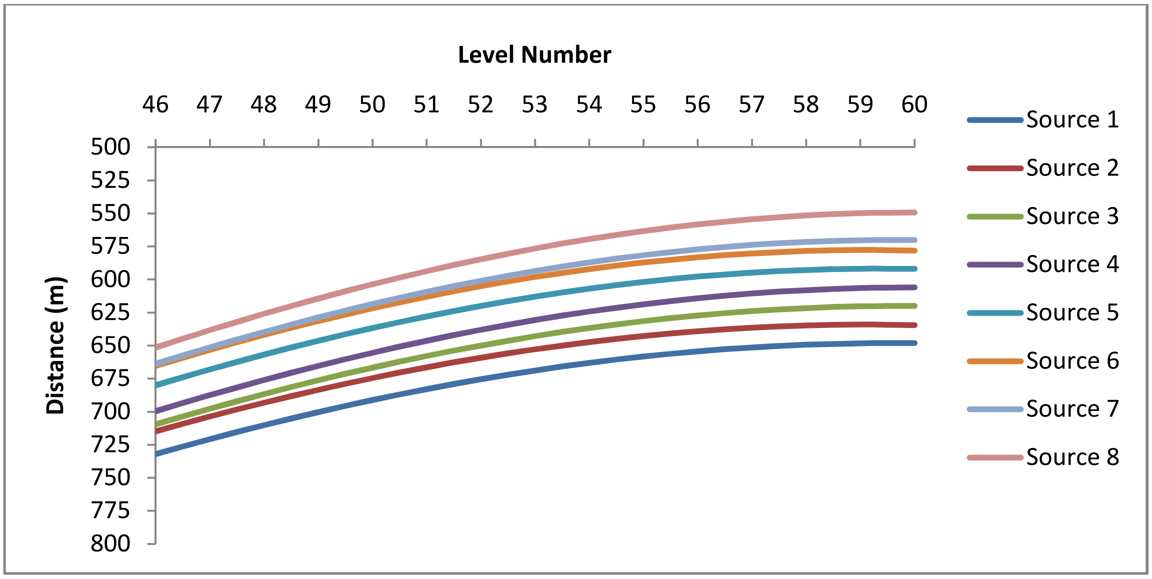


Figure (5.7): Ray-paths lengths computed for the initial model using ray tracing at 15 levels within monitoring well 3. Level 46 is the shallowest at a depth of 100m. The levels are incremented at 25m.

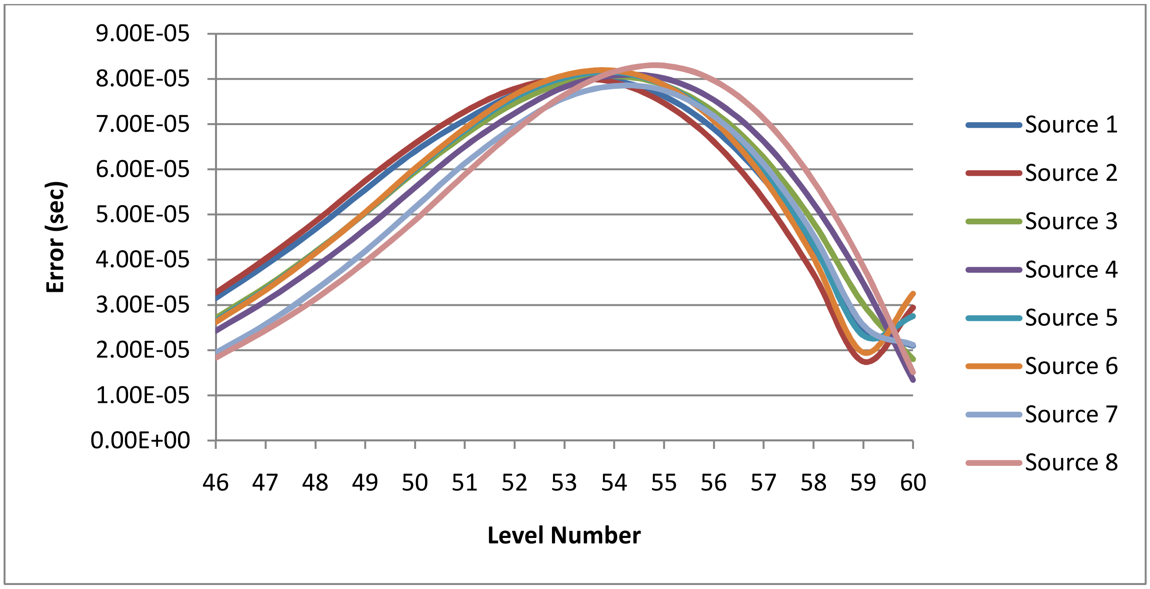


Figure (5.8): Difference between computed traveltimes using forward modelling algorithm and calculated traveltimes using equation (5.3.2) for the initial model at 15 levels within monitoring well 3.

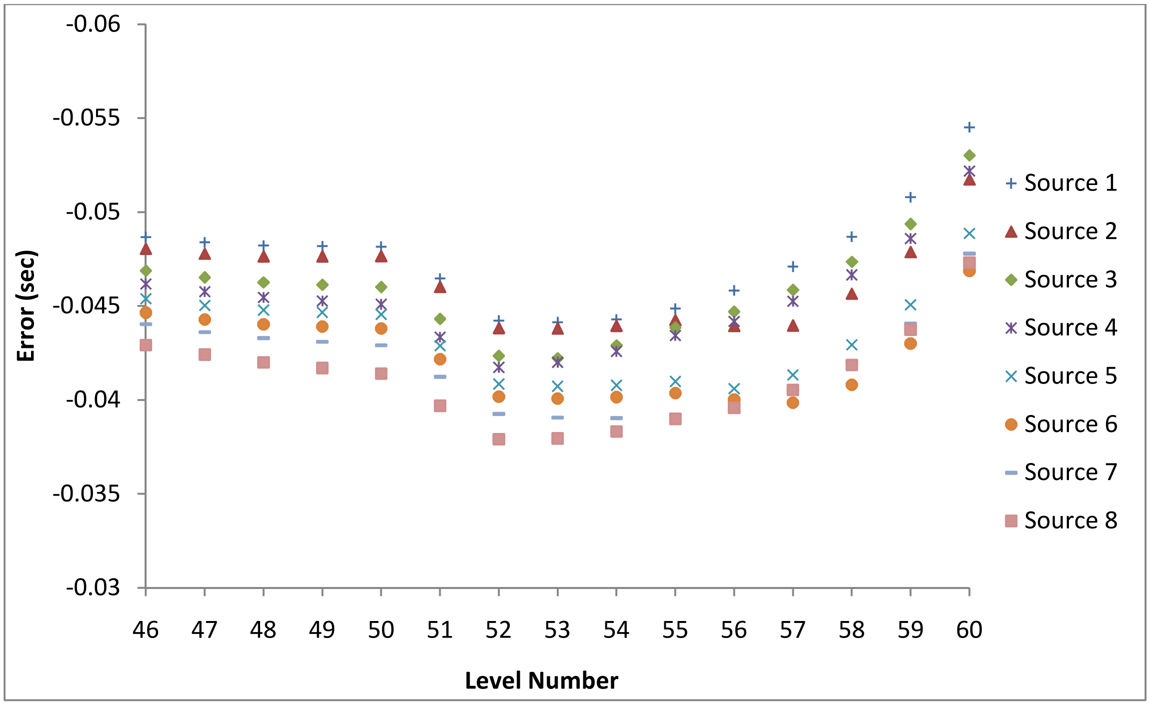


Figure (5.9): Difference between generated arrival times for the true model and computed traveltimes for the initial model at 15 levels within monitoring well 3.

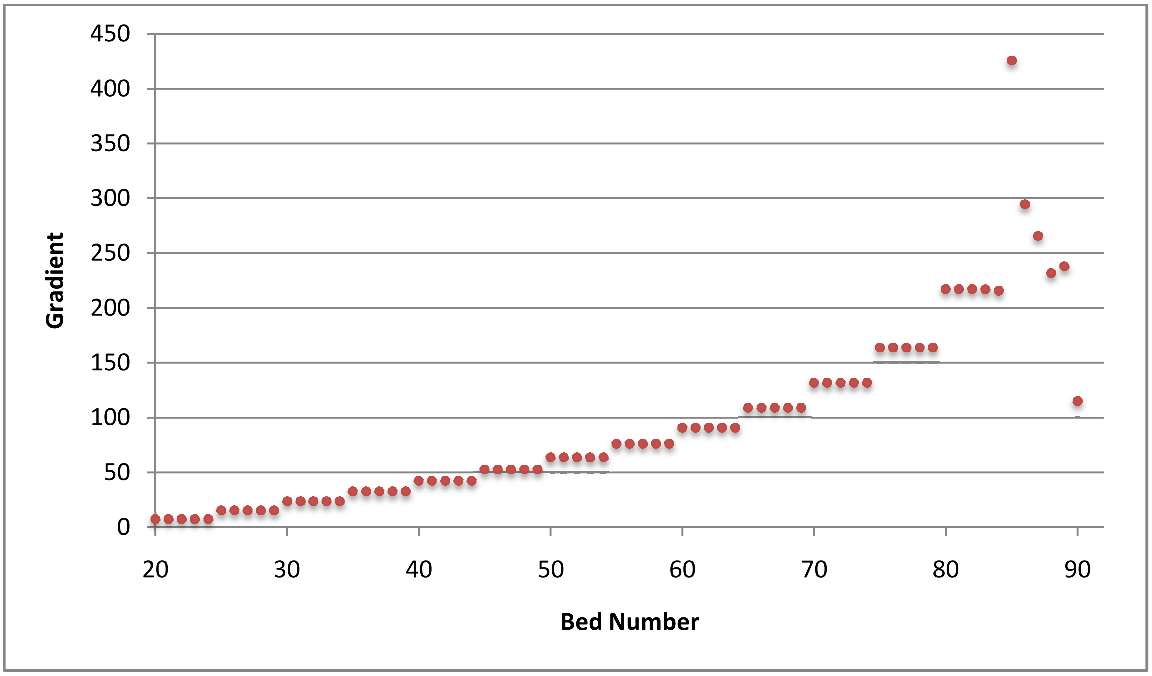


Figure (5.10): Gradients components at different beds computed for the initial model.

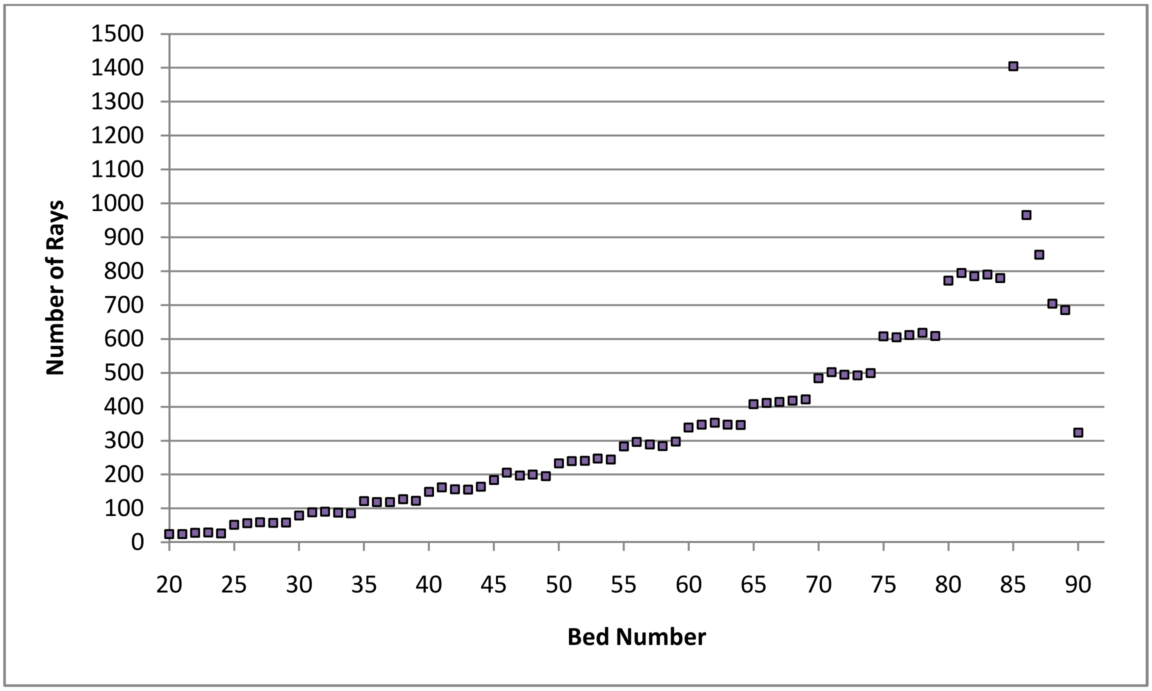


Figure (5.11): Number of rays passing different beds for the initial model.

Another option was to set the bedding exactly as the original model. This option is still practical, but simple to obtain with other methods. Besides, often the structure boundaries are not known. Still, it is straightforward to switch to this option with even less computation during the optimisation phase whenever the structure boundaries are known. Indeed, I started testing the algorithm for this setting and concluded that the algorithm is effective for such situations. I do not show any results for this case as they are trivial.

The algorithm is easily adjusted to fit any other structure by adjusting the grid size. This is very useful whenever the structure boundaries are known. However, when the grid size is made too small, this will result in huge computation cost and excessive demand for memory. On the other hand, larger grid size may result in less accurate traveltimes leading to inaccuracy in the reconstructed slowness model.

The 5m bedding results in one hundred horizontal beds. However, the algorithm reconstructs the slowness model parameters only for seventy beds because the remaining thirty beds provide no contribution to the objective function. This setting assumes that we have no knowledge of the structure boundaries though it is implied that the beds are not dipping. This can be justified when dealing with a single monitoring well. Since the micro-seismic events tend to cluster, the ray-paths connecting such sources and the receiver stations, fixed in a vertical monitoring well, cover small lateral area. The results that I will show are for the case when the three monitoring wells are used. Still, I tested the algorithm with only single monitoring well and the results were exactly the same. So, these results are not shown.

The algorithm accurately obtains the slowness model parameters for all the seventy beds. The reconstruction algorithm executed 21 times iteratively for updating the ray-paths components and thus rebuilding the objective function and its gradient. Following each reconstruction iteration, the optimisation algorithm executes 3 times for updating the slowness model parameters. The results are shown in Figure (5.12) to Figure (5.16).

Figure (5.12) shows the slowness model parameters for layer 2. This Figure has twenty curves with different colours corresponding to twenty beds (bed 20 to bed 39). Of course, the second layer thickness is 125 metres, but the first five beds do not contribute to the objective function as there are no rays passing through them. Careful analysis of the Figure shows that the plots are clustered into four groups such that each group is composed of five curves. This is due to the fact that every two consecutive receiver stations have a separation of 25m. Further, each separate group has a unique convergence to the true model. One can notice that the first has the lowest convergence rate because the corresponding beds have the least rays passing through them. The last group, on the other hand, has the highest convergence rate among its peers. The surprising result is the fact that after 21 iterations all the beds for this layer converge remarkably to the true slowness model parameters. In fact, after 15 iterations, three groups have already converged. Clearly, the convergence behaviour is nonlinear which is related to the problem being nonlinear.

Figure (5.13) shows the slowness model parameters for layer 3. This Figure has ten curves with different colours corresponding to ten beds (bed 40 to bed 49). Careful analysis of this Figure shows that the curves are clustered into two groups such that each group is composed of five plots. After a few iterations, the two groups become indistinguishable because they have enough rays passing through them with more rays passing through the latter. The convergence rate is fast for both of them. Notice the remarkable jump between the first iteration and the third one. Still, the convergence to the true slowness model parameters occurred at the last three iterations.

Figure (5.14) shows the slowness model parameters for layer 4. This Figure has fifteen curves with different colours corresponding to fifteen beds (bed 50 to bed 64). Careful analysis of this Figure shows no clustering at the start and this is so because we started with a homogenous slowness model. This layer is the middle layer with respect to the different sources and receiver stations. Therefore, the rays passing through this layer are much harmonised as shown in Figure (5.11). After 15 iterations, the three groups have become indistinguishable and converged to the true slowness model. Similar to the previous Figure, there is a remarkable jump between the first iteration and the second one and this is due to optimising the step length parameter.

Figure (5.15) shows the slowness model parameters for layer 5. This Figure has twenty curves with different colours corresponding to twenty beds (bed 65 to bed 84). Careful analysis of this Figure shows no clustering at the start and this is so because of the homogenous slowness model we used. After the second iteration the clustering starts to become clear. This layer is relatively closer to the sources and hence has narrow vertical apertures. Therefore, the rays passing through this layer have less curvature and thus the gradient is comparatively smaller. This is obvious for the last two groups and it exhibits itself in the convergence rate. Even after 21 iterations, the five groups, specially the last two, are easily distinguishable with relatively slow convergence rate to the true slowness model parameters. The jump at the first iteration is very noticeable for the first three groups due to the relatively large gradient.

Figure (5.16) shows the slowness model parameters for layer 6. This Figure has four curves with different colours corresponding to four beds (bed 85 to bed 88). This layer thickness is 25 metres, but the last bed does not contribute to the objective function as there are no rays passing through it. Careful analysis of this Figure shows no clustering except in the first few iterations and this is so because of the homogenous slowness model we used. The sources are imbedded within this layer. Thus, the gradient is small due to the narrow vertical apertures. Therefore, the rays passing through this layer have the least curvature. This is obvious for the group and it exhibits itself in the slow convergence rate. Even after 21 iterations, different curves are easily distinguishable with relatively slow convergence rate to the true slowness model parameters. The jump at the first iteration is very noticeable for the group due to the effect from the other layers.

Although I use only eight clustered sources, I think that the method have worked suitably. We are able to reconstruct the velocity model parameters almost exactly without any presumption about the boundaries between the different formations. In addition, the true boundaries between consecutive layers were defined clearly. Despite the fact that the initial slowness model parameters are far different from the actual model parameters and the existence of velocity inversions in the model, the method remains effective. For this example, it took only 21 iterations to converge to the true slowness model parameters though there are seventy model parameters, which would require the conjugate gradient method at least 69 iterations. The solutions are clustered according to their receiver stations for the upper three layers, but then shift toward the sources.

I have run many other tests on the same data, but with different starting slowness model parameters. The results of such tests suggest that the method is robust and the problem we have might not have severe nonlinearity as we have anticipated. It might be better to run the method separately for each monitoring well to sustain the assumption of the lateral variations. Further, it makes sense to have geophones planted above and below the target zone in order to have wider aperture.

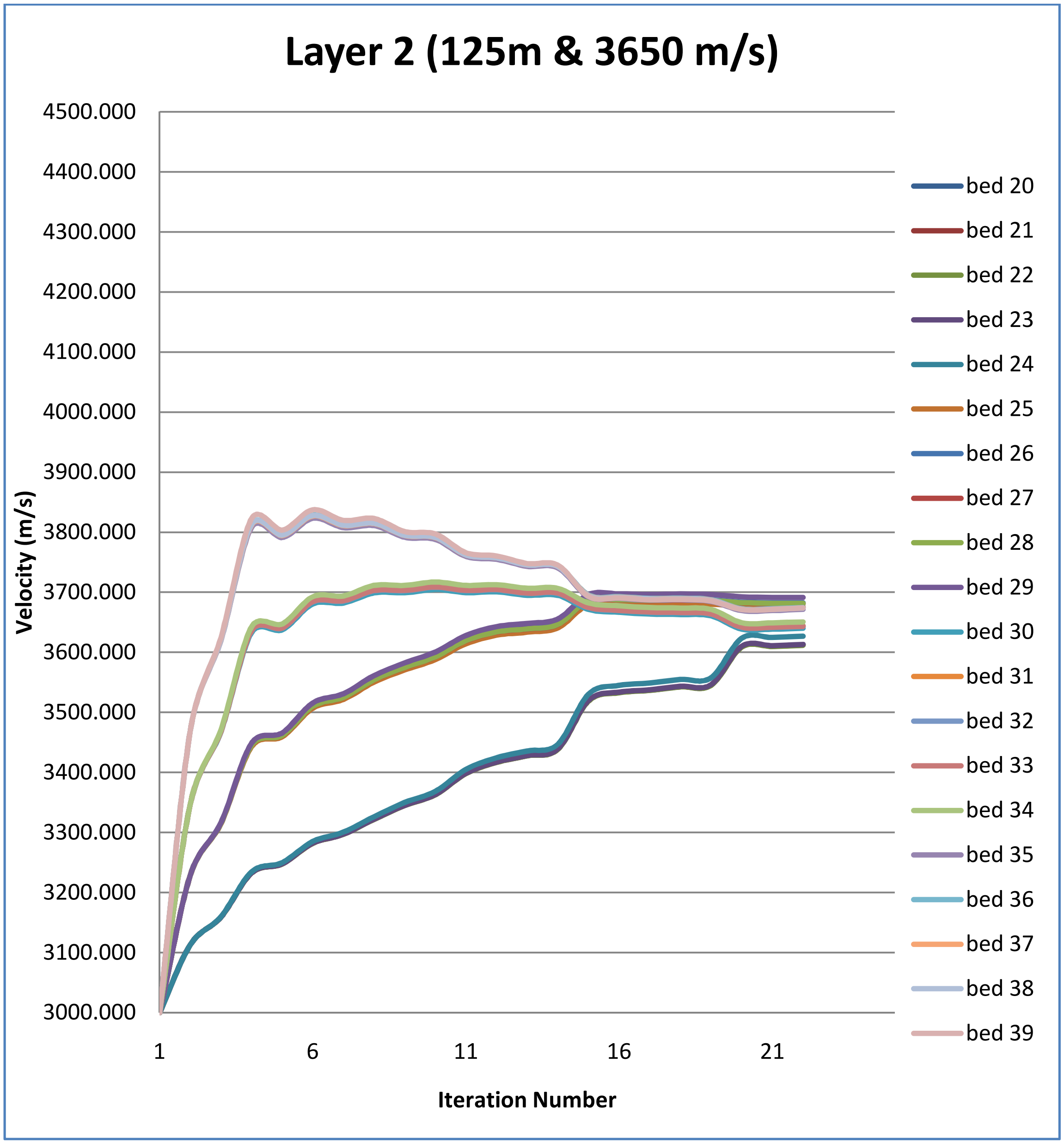


Figure (5.12): Second layer with its twenty components labelled bed 20 to bed 35. The first five beds are not shown as they provide no contribution. Different colours are given to different beds.

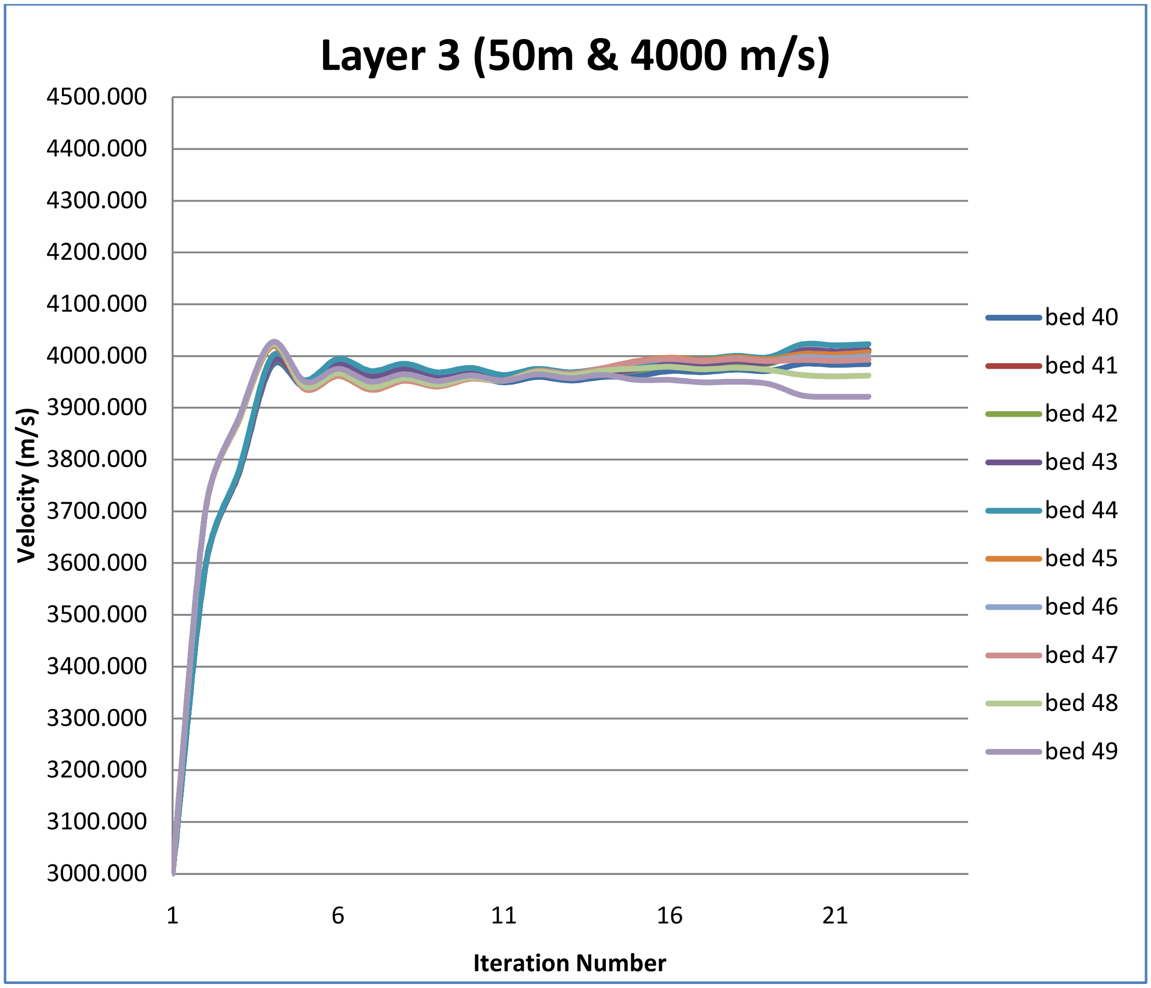


Figure (5.13): Third layer with its ten components labelled bed 40 to bed 49. Different colours are given to different beds.

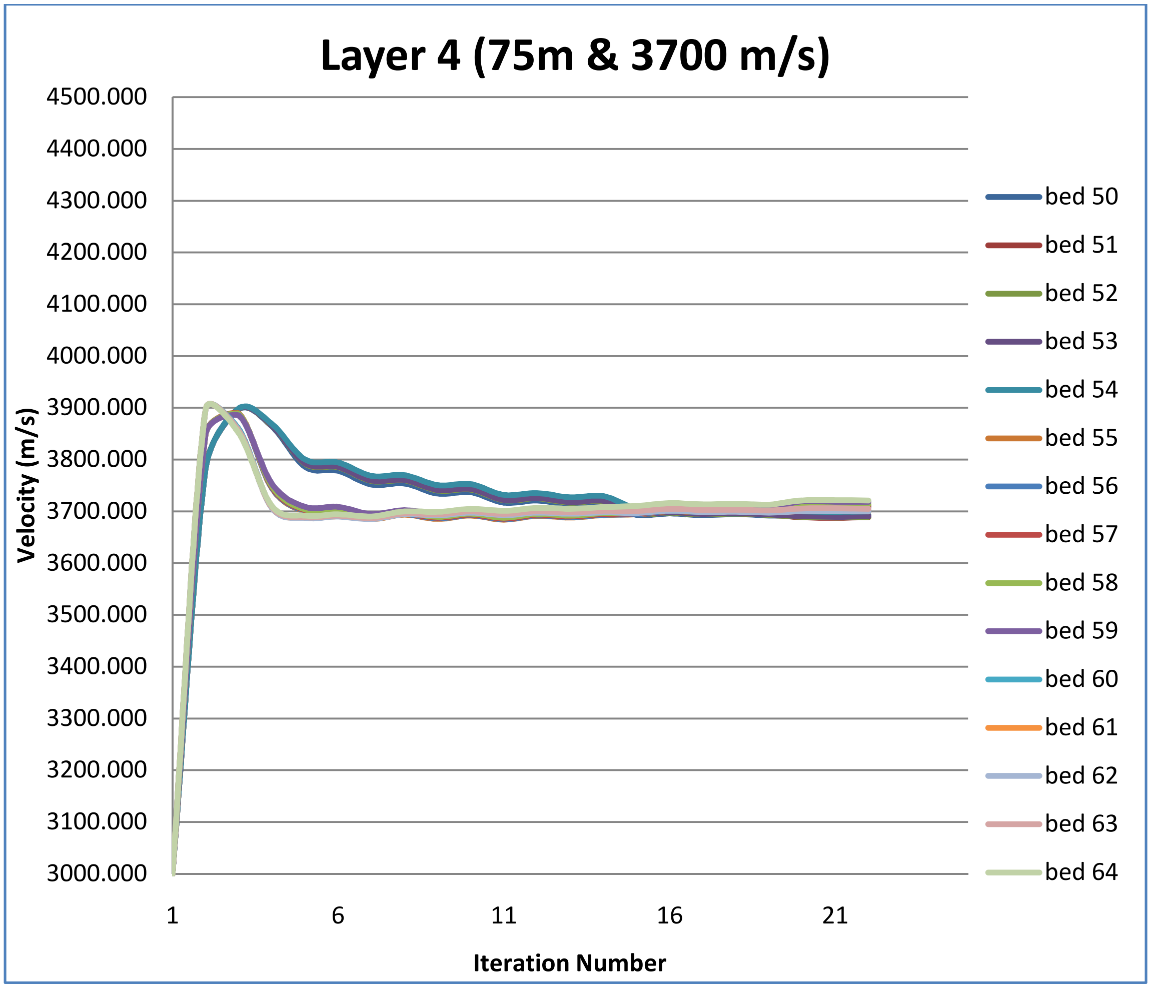


Figure (5.14): Fourth layer with its fifteen components labelled bed 50 to bed 64. Different colours are given to different beds.

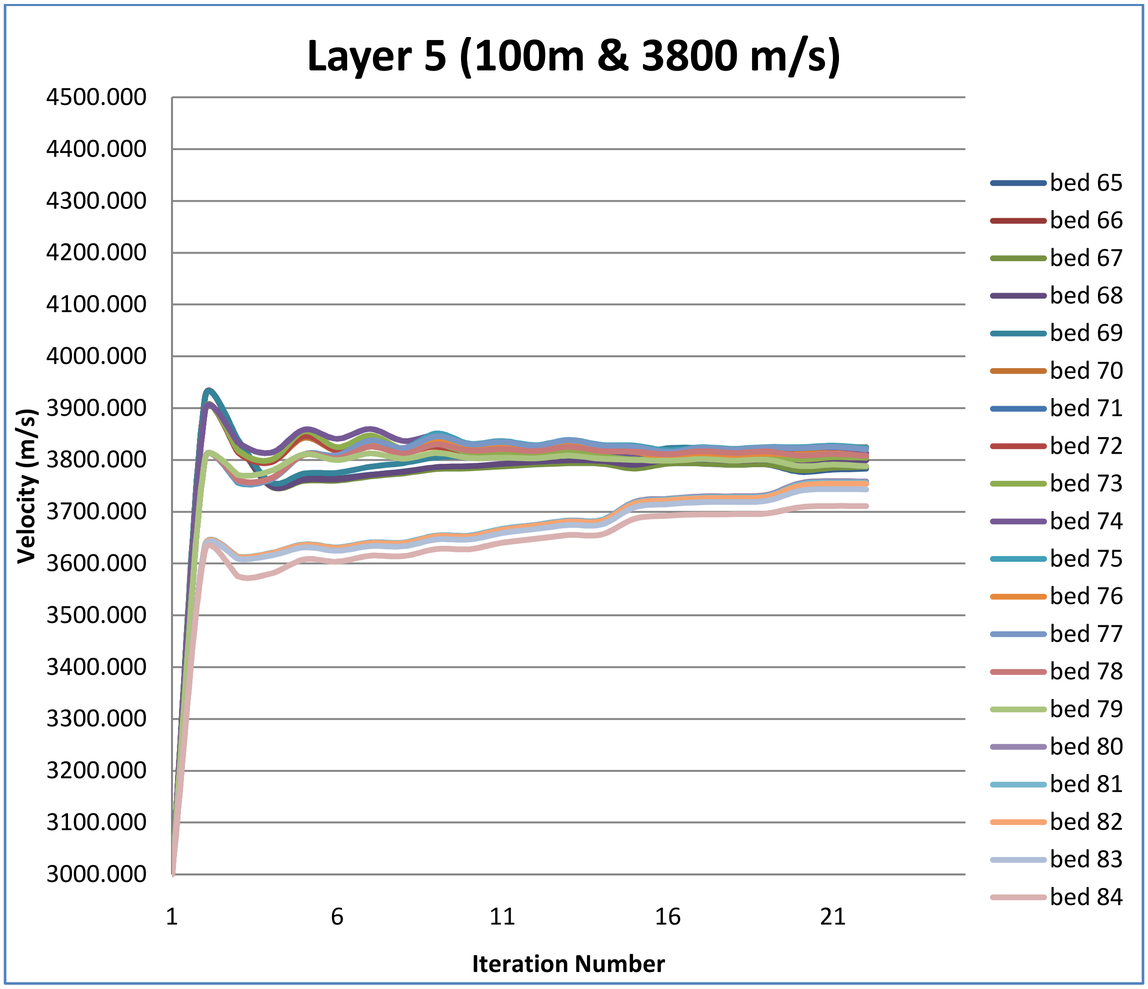


Figure (5.15): Fifth layer with its twenty components labelled bed 65 to bed 84. Different colours are given to different beds.

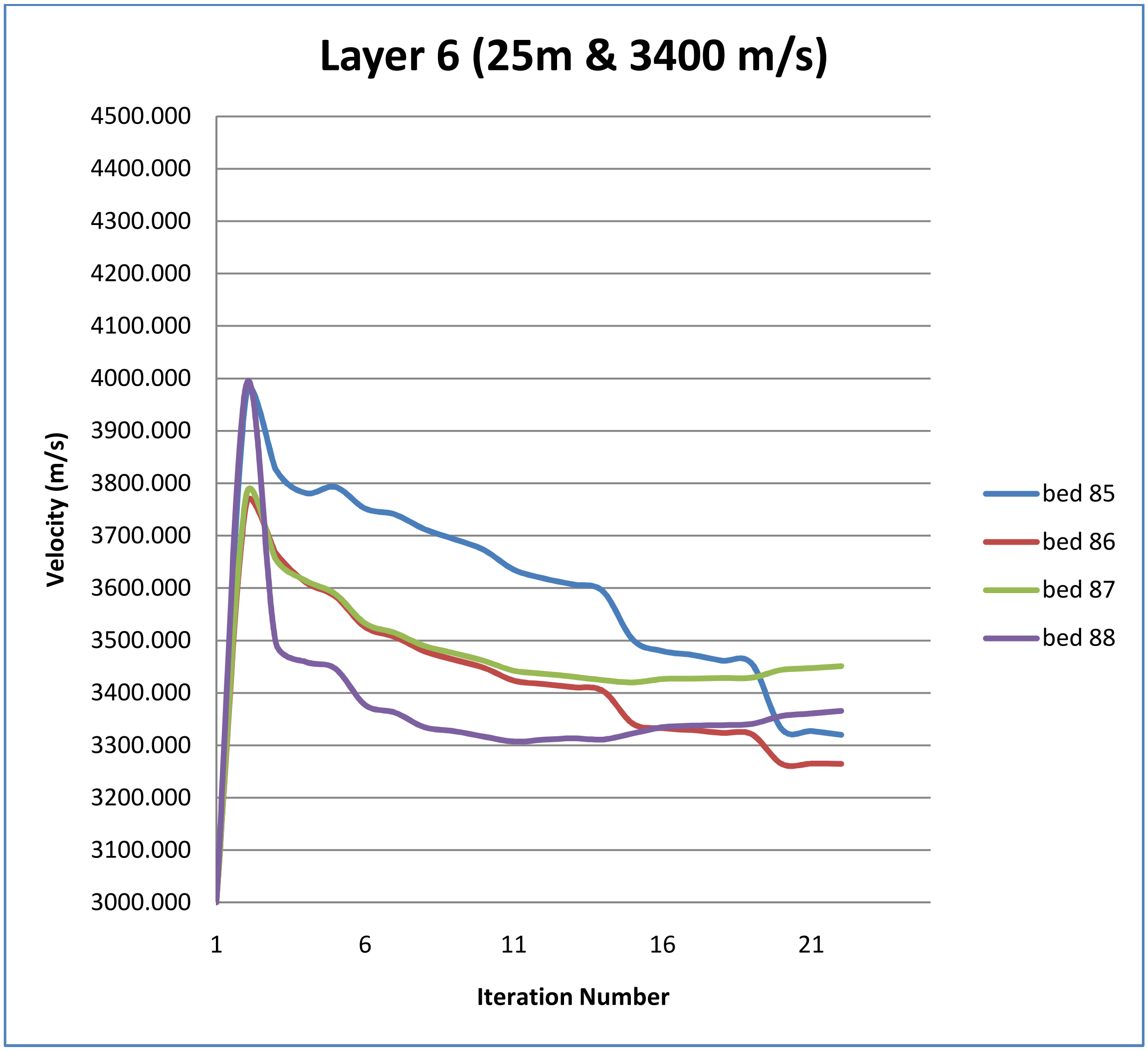


Figure (5.16): Sixth layer with its four components labelled bed 85 to bed 88. The last bed is not shown as it has no contribution. Different colours are given to different beds.

5.7 Error Analysis

There are three sources that could introduce errors into the results of the algorithm for estimating the velocity parameters. These sources are the event hypocentre and origin time, the picked arrival time and the receiver position. An error analysis is presented to address such error sources and their effects. The analysis is performed on a velocity model with seven independent random variables. The exact values of the seven random variables of such model are listed in Table (5.1). In this analysis, I have set the bedding exactly as the original model. Therefore, we will have six layers to deal with as there are no rays passing through the first layer.

Initially, the algorithm was used to obtain the velocity parameters using free error data and the result is shown in Figure (5.17) for 21 iterations. The plot shows that after four iterations the algorithm remarkably converges to the true values with less than 0.6% error except for the last two layers. For layer 6, the algorithm was able to recover the velocity parameter with 3.5% error. The reason for this relatively large error is the fact that this layer contains the microseismic events. Consequently, the rays that travel through this layer are nearly horizontal as shown in Figure (5.2). This causes the gradient to be less sensitive to the velocity within layer 6 irrespective of the number of rays. Layer 7, on the other hand, has the least number of rays passing through. Still, the algorithm was able to retrieve the velocity parameter within layer 7 with 1.2% error. In fact, similar behaviour is noticeable when random errors are introduced.

The velocity parameters obtained using free error data are set as a benchmark when I study the effects of the different random errors. Therefore, all the results are compared with and contrasted against such values. This would quantify the effects of error sources on the results both individually and collectively.

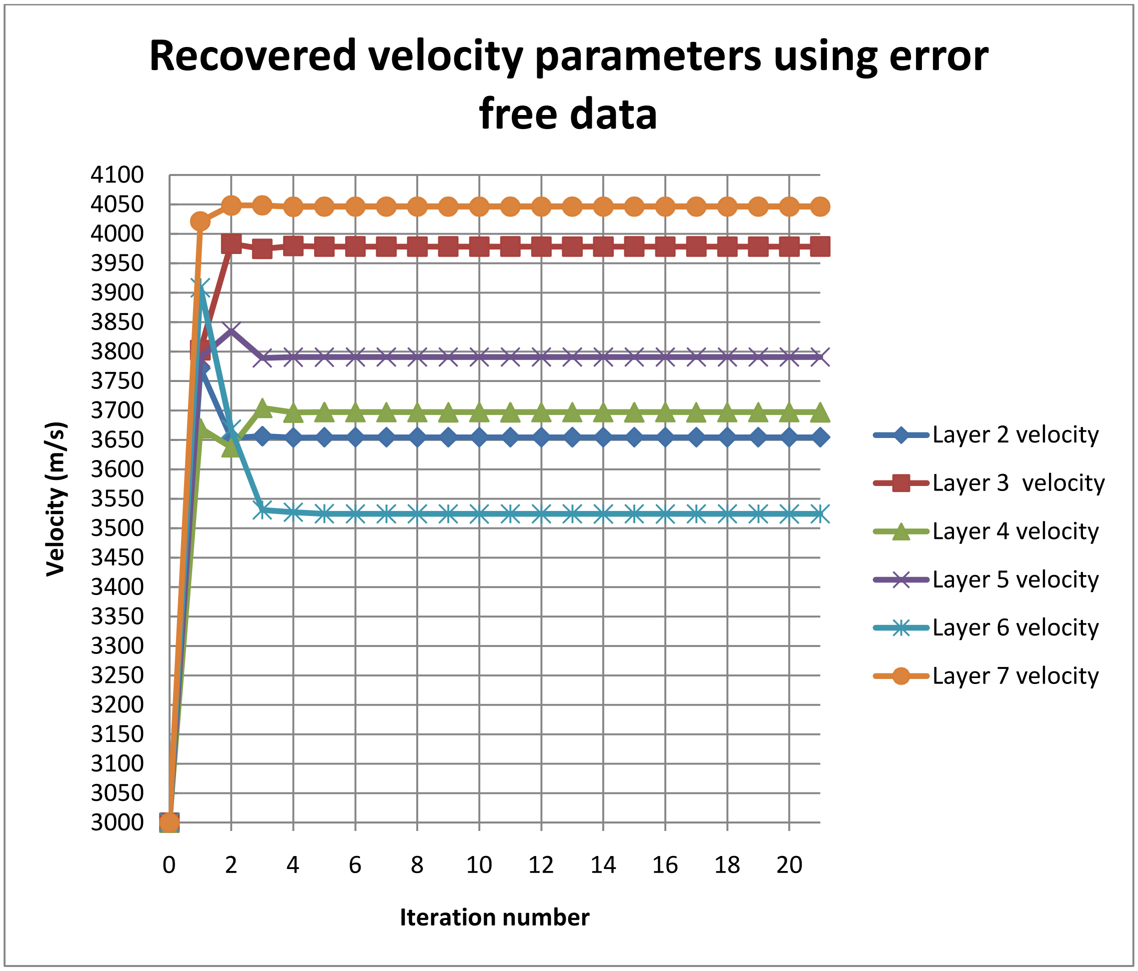


Figure (5.17): Plots showing the Velocity parameters obtained by the algorithm after 21 iteration using error free data for the six different layers.

There are a total of 527 variables that could introduce errors into the results; 32 variables belong to the eight events' four parameters including hypocentre coordinates and origin times; 360 variables belong to the picked arrival times coming from 45 receivers and 8 microseismic events; 135 variables belong to the receiver positions' coordinates. To do the error analysis, each variable belonging to a particular error source is treated independently. For example, the eight events parameters listed in Table (4.1) have 32 separate random errors. Initially, random errors are simulated for all variables that belong to one error source while maintaining the remaining variables belonging to the other two error sources fixed. The simulated random errors are added to each variable individually. This process provides three independent simulation experiments; one for each error source. Finally, random errors for the three error sources are simulated and introduced into all the 527 variables simultaneously. Random errors are simulated 100 times for each error experiment.

First, I contaminated the value of each parameter that belongs to the microseismic event with ±5m random errors for the coordinates and ±2ms for the origin time as suggested by the error analysis described in Chapter 4. The random errors for the origin time and the x-coordinate belonging to the first event are shown in Figure (5.18) and Figure (5.19). The random event errors for the other seven remaining events are not shown as they have similar trend. The random events' errors have induced errors in the recovered velocity parameters. Such induced errors are approximated with normal probability distributions. The second layer velocity errors have a displaced normal distribution with its expected mean μ_2=20m/s and standard deviation σ_2=24m/s as shown in Figure (5.20). The third layer velocity errors fit a normal distribution with its expected mean μ_3=-2m/s and standard deviation σ_3=20m/s as shown in Figure (5.21). The fourth layer velocity errors have a normal distribution with its expected mean μ_4=6m/s and standard deviation σ_4=20m/s as shown in Figure (5.22). The fifth layer velocity errors display a normal distribution with expected mean μ_5=-12m/s and standard deviation σ_5=28m/s as shown in Figure (5.23). The sixth layer velocity errors exhibit a shifted asymmetrical distribution with expected mean μ_6=178m/s and standard deviation σ_6=200m/s as shown in Figure (5.24). The seventh layer velocity errors show a displaced normal distribution with expected mean μ_7=52m/s and standard deviation σ_7=57m/s as shown in Figure (5.25). Generally, the standard deviations have relatively small variability except, as anticipated, for the last two layers. This suggests that when the events parameters have ±5m errors for the hypocentre and ±2ms errors for the origin time, the algorithm with a 95% confidence is capable of retrieving the exact values of the velocity parameters, excluding the sixth layer, with a maximum error of ±50m/s. It also confirms that the effect of the ±5m random position errors and ±2ms random origin time errors on the velocity parameters is minor. Layer 2 has 20m/s velocity shifts when compared to the error free results for the same layer. For layer 7, the results are consistent with the result obtained using error free data, in particular the ~50m/s shift (expected mean). The rays that pass through the sixth layer containing the eight events are almost horizontal. Consequently, the algorithm is unable to resolve the velocity of this layer and hence the large expected mean and standard deviation. Still, the results, for this layer, are consistent with the results obtained using error free data for the same layer. The introduced random errors have mostly affected the sixth layer exhibited as 200m/s standard deviation.

Second, I added ±5 (ms) random error into the observed arrival times. The random time errors for the observed arrival time belonging to the first receiver are shown in Figure (5.26). The remaining 359 picked arrival times have similar random time errors. Like the event random errors, the time random errors have introduced normally distributed errors to the velocity parameters. The second layer velocity errors have a normal distribution with its expected mean μ_2=12m/s and standard deviation σ_2=32m/s as shown in Figure (5.27). The third layer velocity errors fit a normal distribution with its expected mean μ_3=4m/s and standard deviation σ_3=37m/s as shown in Figure (5.28). The fourth layer velocity errors have a normal distribution with its expected mean μ_4=7m/s and standard deviation σ_4=28m/s as shown in Figure (5.29). The fifth layer velocity errors display a normal distribution with expected mean μ_5=-4m/s and standard deviation σ_5=15m/s as shown in Figure (5.30). The sixth layer velocity errors exhibit a skewed normal distribution with expected mean μ_6=137m/s and standard deviation σ_6=152m/s as shown in Figure (5.31). The seventh layer velocity errors show a displaced normal distribution with expected mean μ_7=57m/s and standard deviation σ_7=59m/s as shown in Figure (5.32). Compared to the results obtain in the first experiment, the standard deviations have comparatively wider variability for the first three layers and narrower variability for the last three layers. The expected means are still close to those obtained with error free data. To conclude, when the arrival observed times have ±5ms random errors, the algorithm with a 95% confidence is able to recover the true values of the velocity model parameters, except that of layer 6, with a maximum error of ±60m/s. The velocity of the sixth layer is unresolved by the algorithm as explained by the comparatively wide variability of the standard deviation of the errors distribution for this layer velocity parameter. Nonetheless, the results, for this layer, are consistent with the results obtained using error free data for the same layer. The effect of the ±5ms random time errors on the velocity parameters is very similar to the results obtained in the first experiment.

Third, I introduced ±5m random errors into the receiver position coordinates. The random position errors for the x-coordinate of the first receiver position are shown in Figure (5.33). The random position errors for the other 134 remaining coordinates are not shown as they have similar trend. The receiver position random errors have introduced errors to the velocity parameters. Such errors have probability normal distributions. The second layer velocity errors have a normal distribution with its expected mean μ_2=-2m/s and standard deviation σ_2=43m/s as shown in Figure (5.34). The third layer velocity errors fit a skewed distribution with its expected mean μ_3=54m/s and standard deviation σ_3=80m/s as shown in Figure (5.35). The fourth layer velocity errors have a normal distribution with its expected mean μ_4=10m/s and standard deviation σ_4=39m/s as shown in Figure (5.36). The fifth layer velocity errors display a displaced normal distribution with expected mean μ_5=45m/s and standard deviation

σ_5=48m/s as shown in Figure (5.37). The sixth layer velocity errors exhibit a shifted normal distribution with expected mean μ_6=-146m/s and standard deviation σ_6=166m/s as shown in Figure (5.38). The seventh layer velocity errors show a displaced normal distribution with expected mean μ_7=21m/s and standard deviation σ_7=25m/s as shown in Figure (5.39). Compared to the results obtain in the first two experiments, the standard deviations have comparatively the widest variability for the first three layers and the narrowest variability for the last layer. Layer 3 and layer 5 have 54m/s and 45m/s velocity shifts, respectively, when compared to the error free results for the same layers. Layer 6, on the other hand, has -270m/s velocity shift when compared to the error free results for the same layer. Accordingly, when the receiver positions have ±5ms errors, the algorithm with a 95% confidence is able to recover the exact values of velocity parameters, except for layer 6, with a maximum error of ±100m. The random receiver positions errors have introduced the most severe effect on the velocity parameters compared to the other experiments.

Last, random errors are generated as specified above and added into all random variables simultaneously. This process simulates the real situation. It has induced errors to the velocity parameters. The simulated random errors have produced errors in the velocity parameters which can be approximated with probability normal distributions. The second layer velocity errors have a shifted normal distribution with expected mean μ_2=-16m/s and standard deviation σ_2=65m/s as shown in Figure (5.40). The third layer velocity errors fit a displaced asymmetrical distribution with its expected mean μ_3=45m/s and standard deviation σ_3=97m/s as shown in Figure (5.41). The fourth layer velocity errors have a normal distribution with its expected mean μ_4=13m/s and standard deviation σ_4=53m/s as shown in Figure (5.42). The fifth layer velocity errors display a skewed normal distribution with expected mean μ_5=41m/s and standard deviation σ_5=59m/s as shown in Figure (5.43). The sixth layer velocity errors exhibit a normal distribution with expected mean μ_6=-131m/s and standard deviation σ_6=235m/s as shown in Figure (5.44). The seventh layer velocity errors show a displaced normal distribution with expected mean μ_7=19m/s and standard deviation σ_7=31m/s as shown in Figure (5.45).These results validate that the random errors have introduced velocity shifts to all the velocity parameters when compared to the error free data results. In addition, the standard deviations of such distributions have large variability resulted from the random errors. This suggests that the effect of random errors on the recovered velocity parameters is more severe compared to the retrieved event parameters dealt with in Chapter 4. For this experiment, the algorithm with a 95% confidence is capable of recovering the exact values of the velocity parameters with a maximum error of ±130m/s (a maximum of 3.5% error), except for layer 6. This confirms that the algorithm is robust in obtaining the event parameters even with the presence of moderate errors.

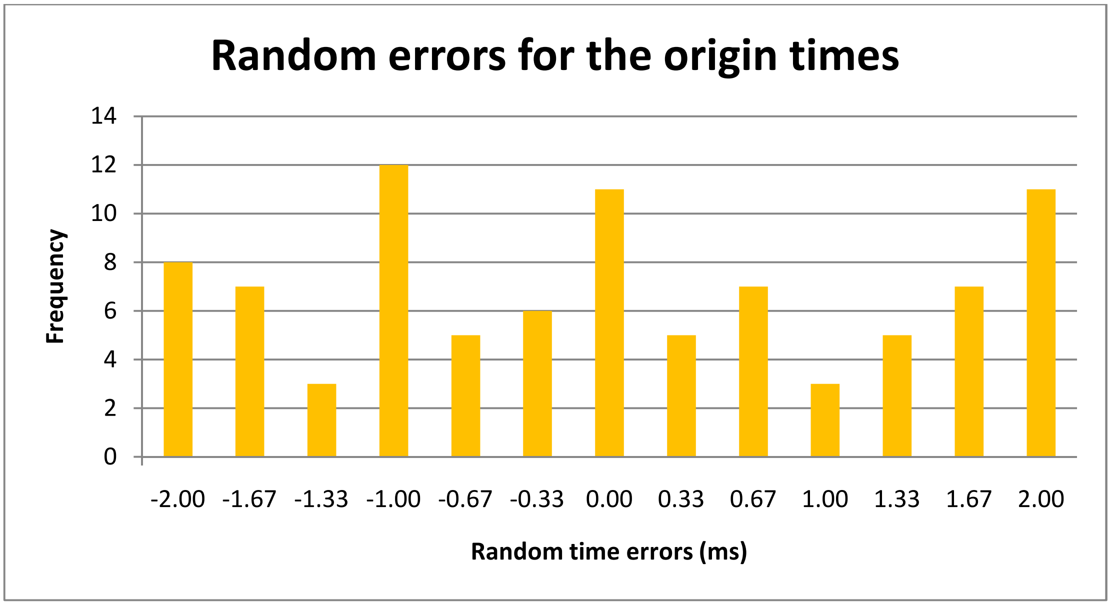


Figure (5.18): Simulation of ±2ms random errors for the origin time of the first event. The origin time random errors were generated 100 times.

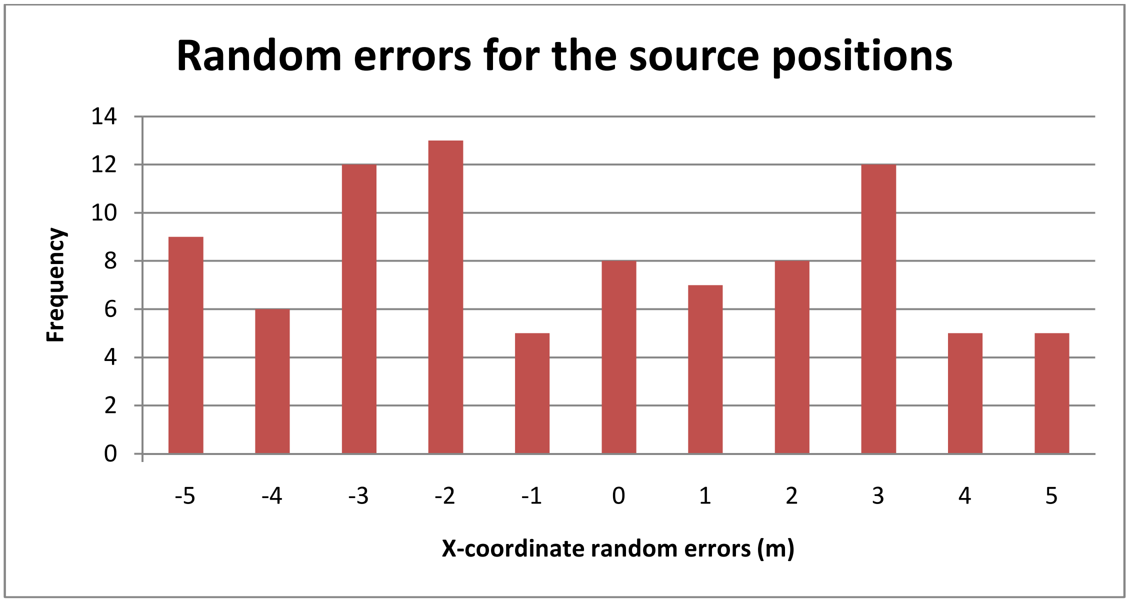


Figure (5.19): Simulation of ±5m random errors for the x-coordinate of the first event. The position random errors were generated 100 times.

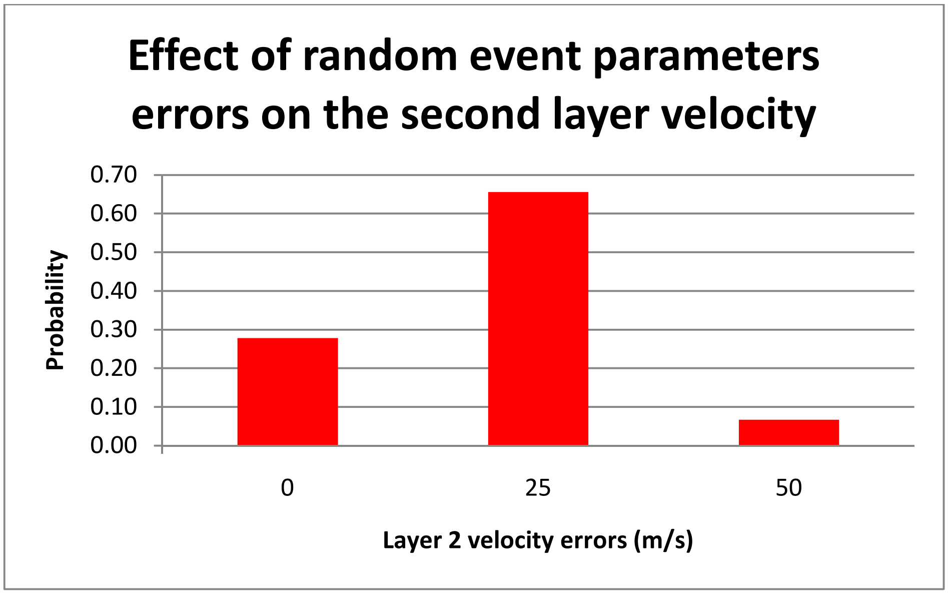


Figure (5.20): Histogram showing errors in the velocity of the second layer with their probabilities after introducing random errors to the event hypocentre and origin time.

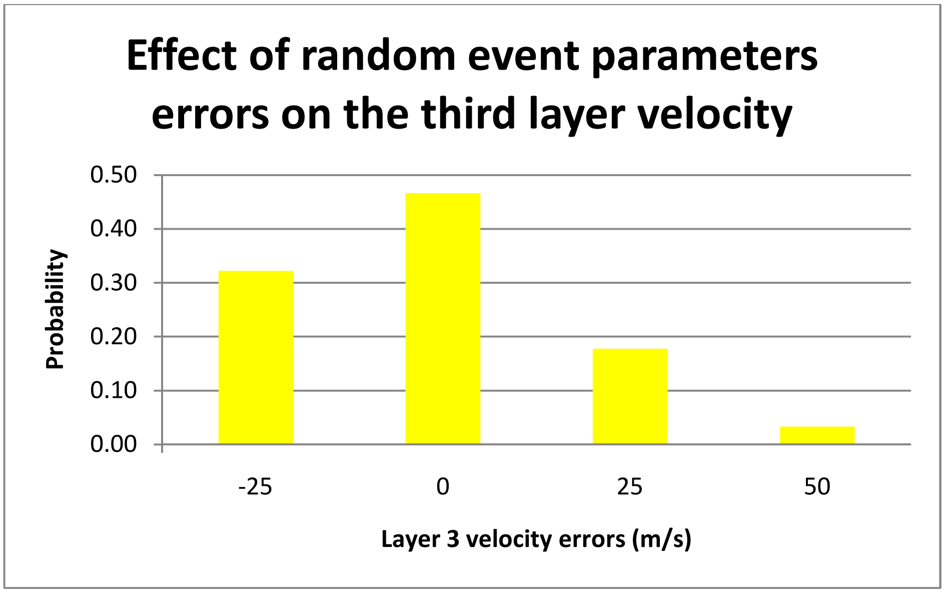


Figure (5.21): Histogram showing errors in the velocity of the third layer with their probabilities after introducing random errors to the event hypocentre and origin time.

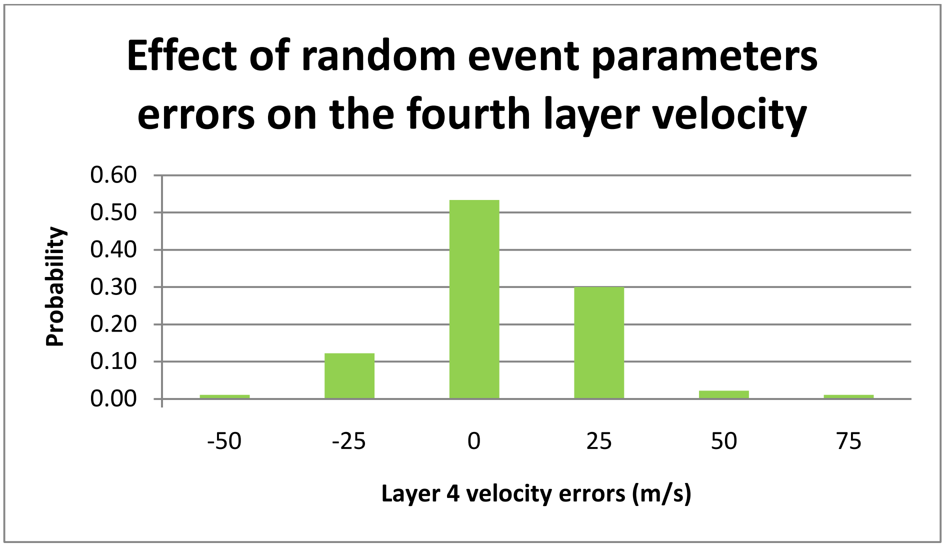


Figure (5.22): Histogram showing errors n the velocity of the fourth layer with their probabilities after introducing random errors to the event hypocentre and origin time.

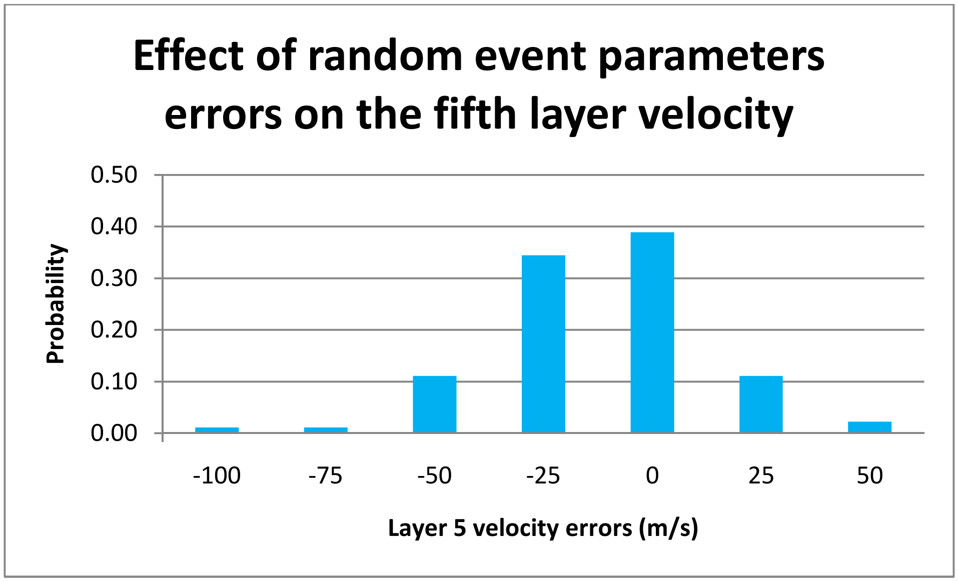


Figure (5.23): Histogram showing errors in the velocity of the fifth layer with their probabilities after introducing random errors to the event hypocentre and origin time.

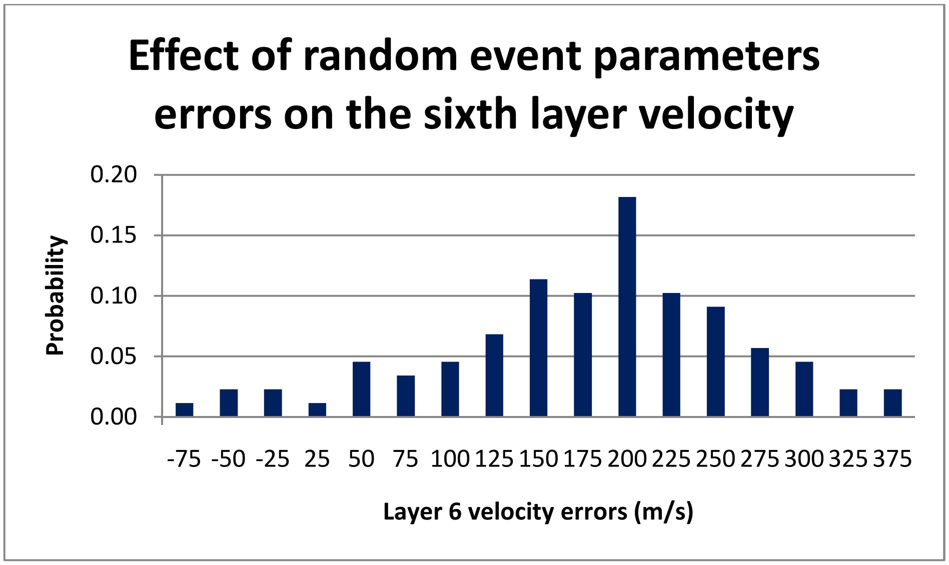


Figure (5.24): Histogram showing errors in the velocity of the sixth layer with their probabilities after introducing random errors to the event hypocentre and origin time.

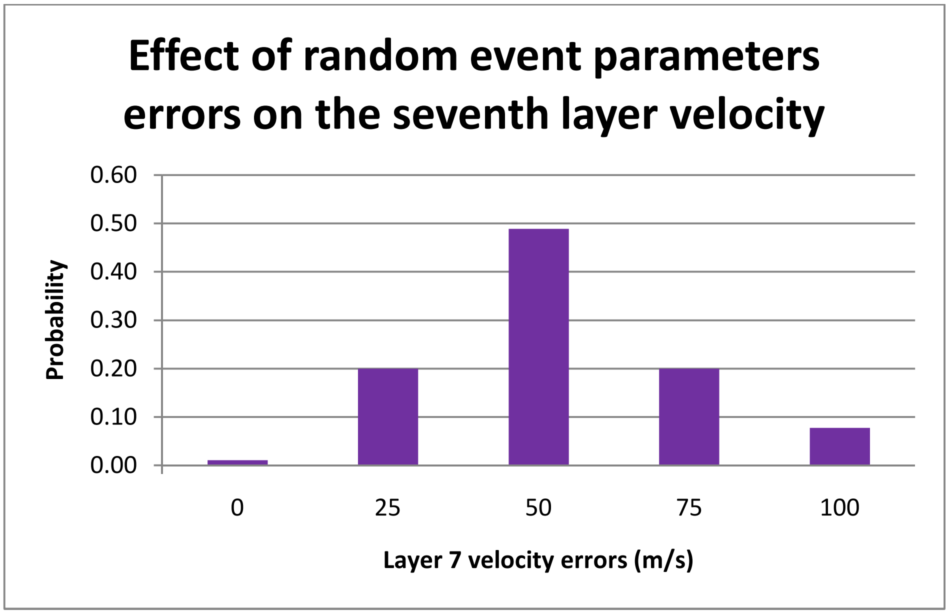


Figure (5.25): Histogram showing errors in the velocity of the seventh layer with their probabilities after introducing random errors to the event hypocentre and origin time.

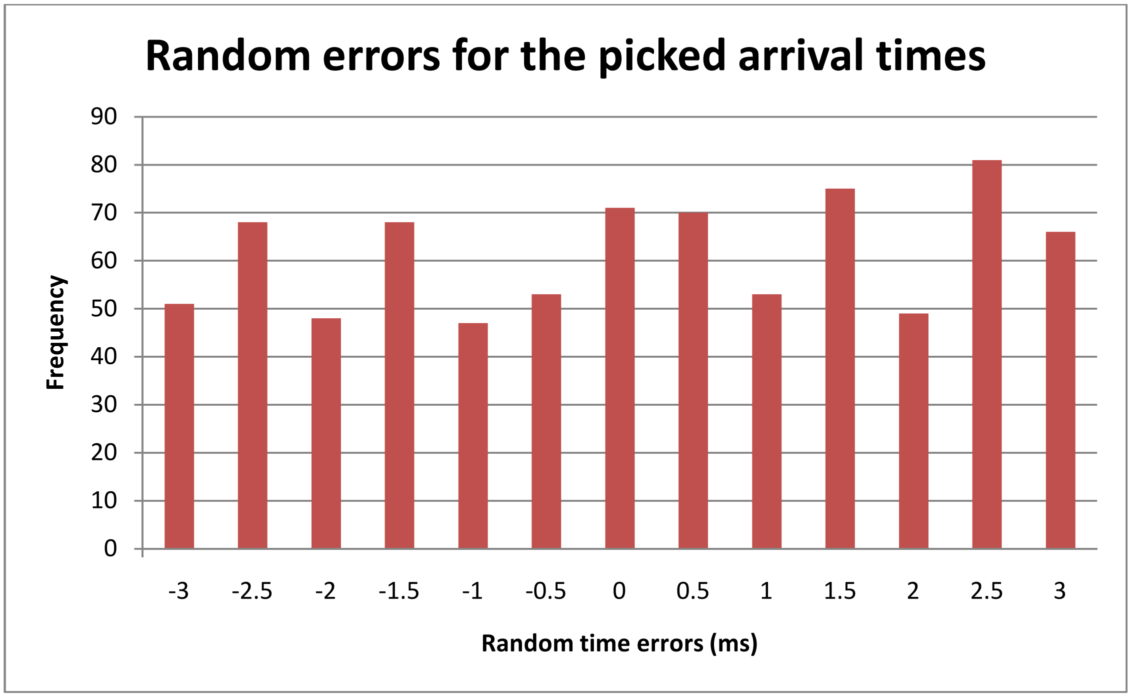


Figure (5.26): Simulation of ±5ms random errors for the picked arrival time related to the first receiver. The time random errors were generated 100 times.

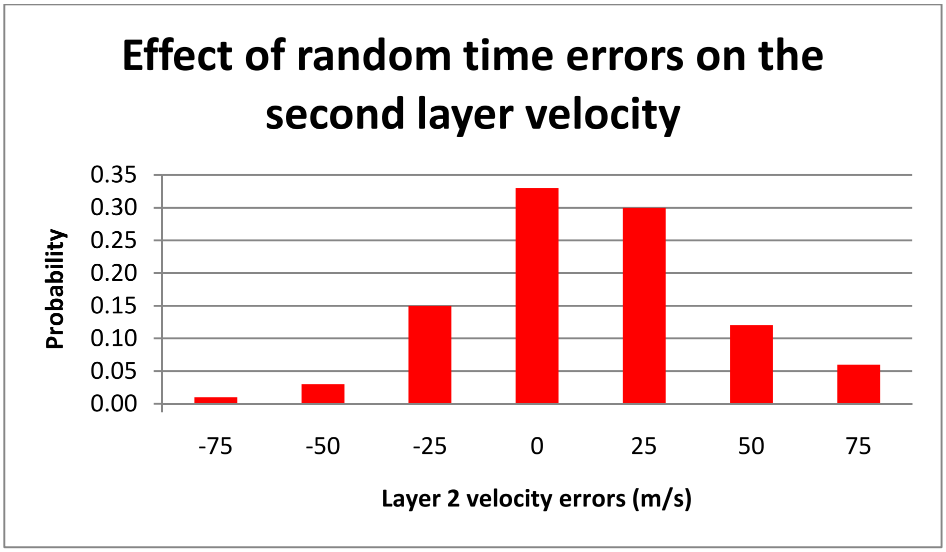


Figure (5.27): Histogram showing errors in the velocity of the second layer with their probabilities after introducing random errors to the picked arrival times.

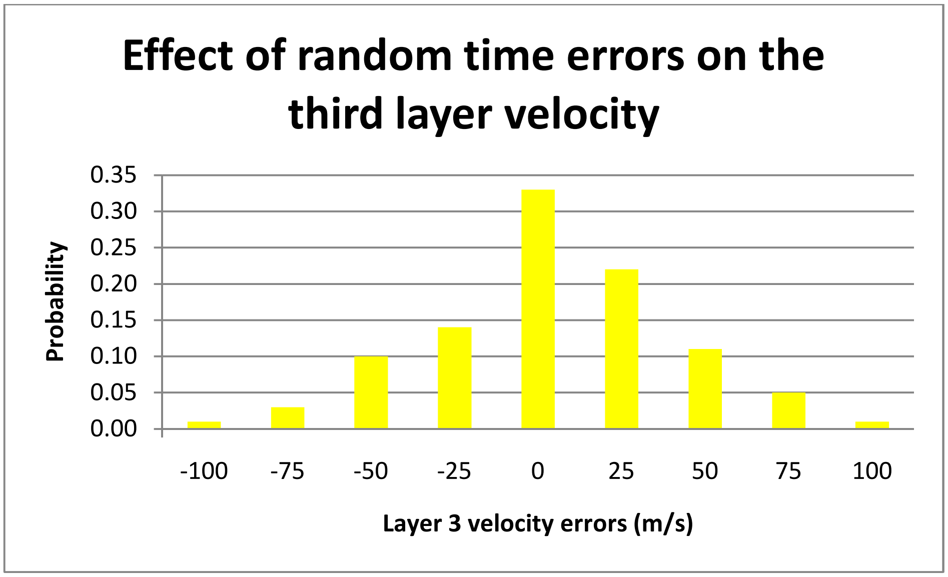


Figure (5.28): Histogram showing errors in the velocity of the third layer with their probabilities after introducing random errors to the picked arrival times.

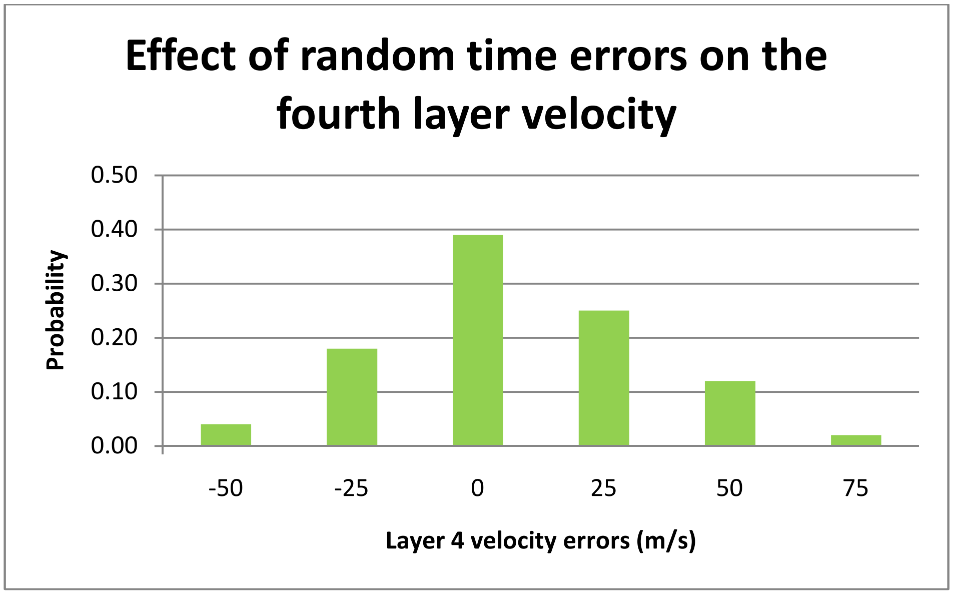


Figure (5.29): Histogram showing errors in the velocity of the fourth layer with their probabilities after introducing random errors to the picked arrival times.

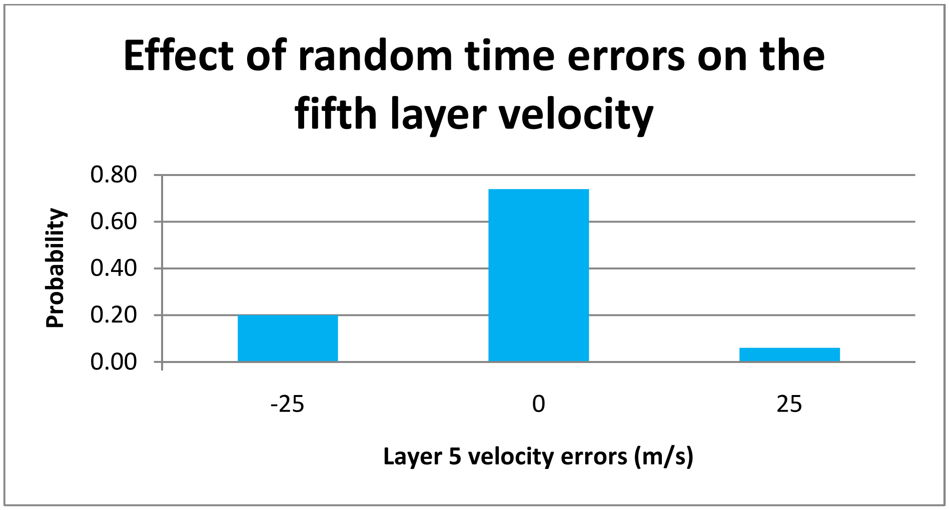


Figure (5.30): Histogram showing errors in the velocity of the fifth layer with their probabilities after introducing random errors to the picked arrival times.

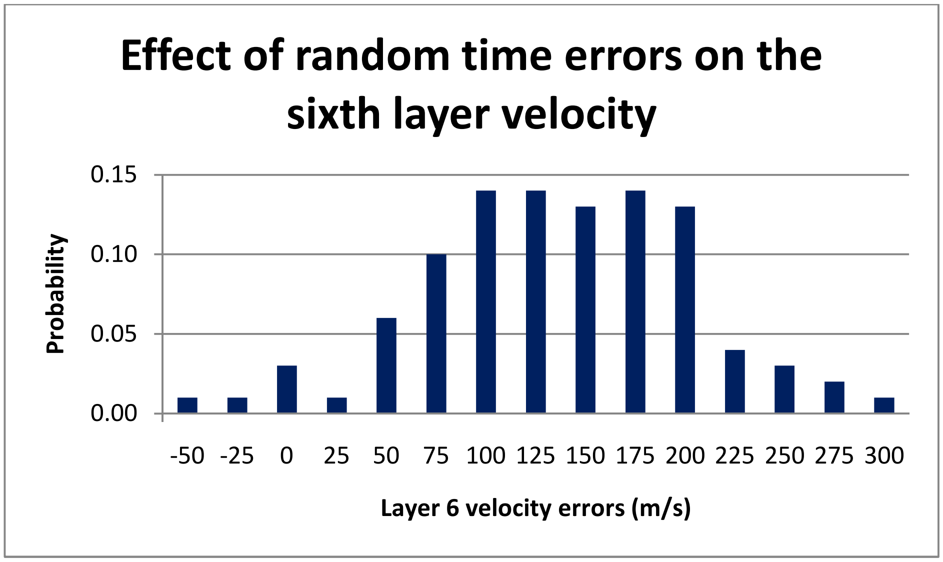


Figure (5.31): Histogram showing errors in the velocity of the sixth layer with their probabilities after introducing random errors to the picked arrival times.

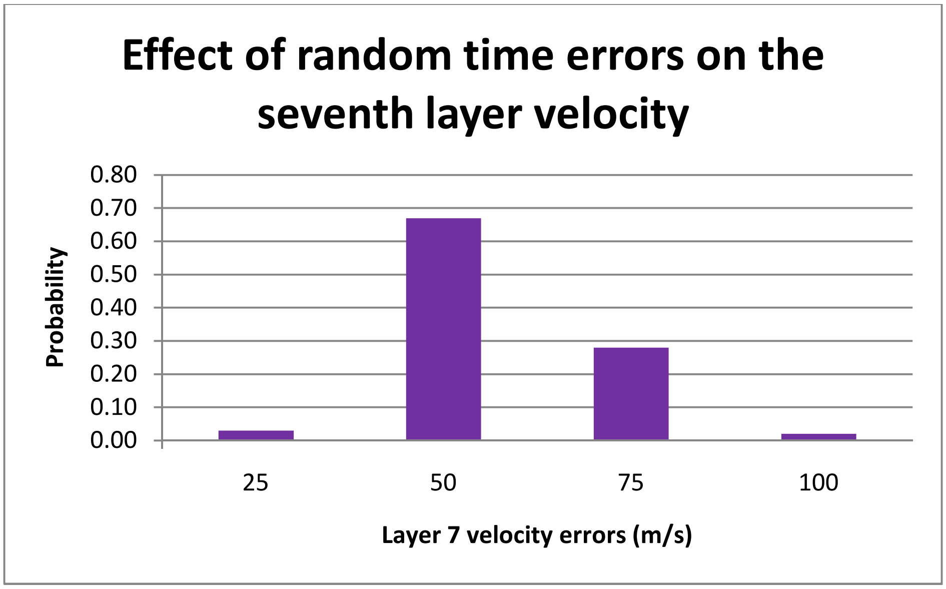


Figure (5.32): Histogram showing errors in the velocity of the seventh layer with their probabilities after introducing random errors to the picked arrival times.

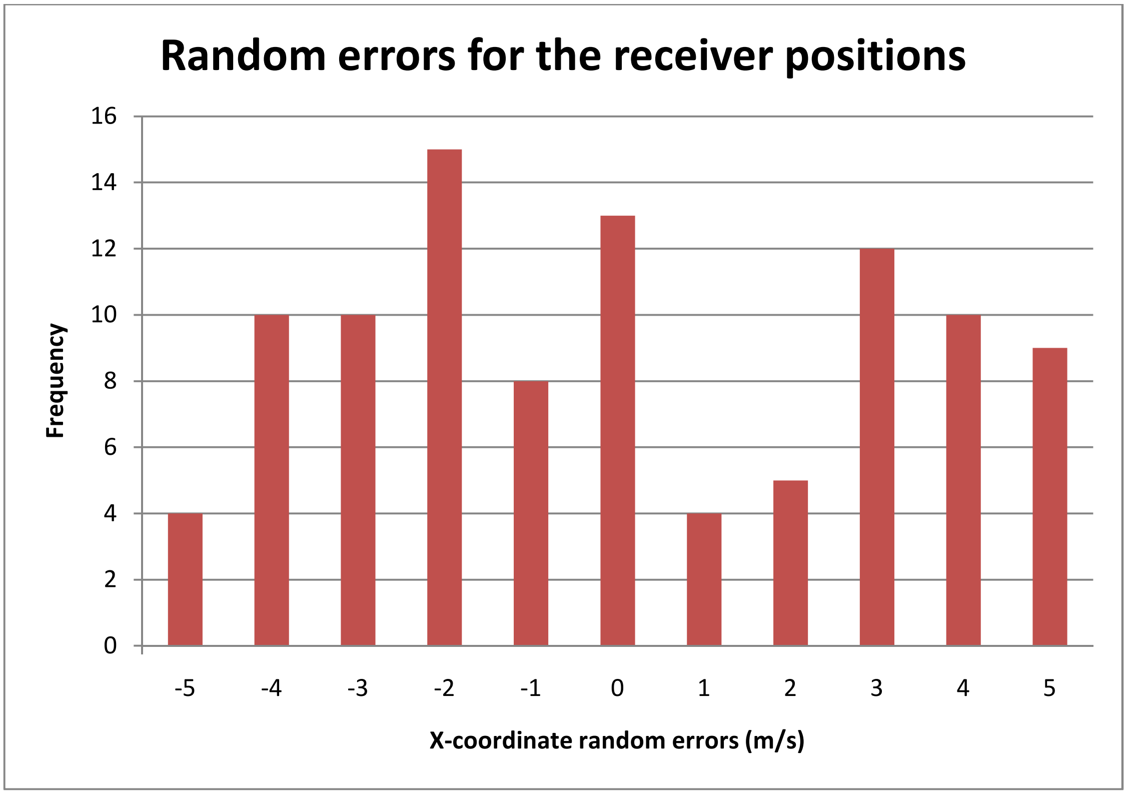


Figure (5.33): Simulation of ±5m random errors for the x-coordinate related to the first receiver. The position random errors were generated 100 times.

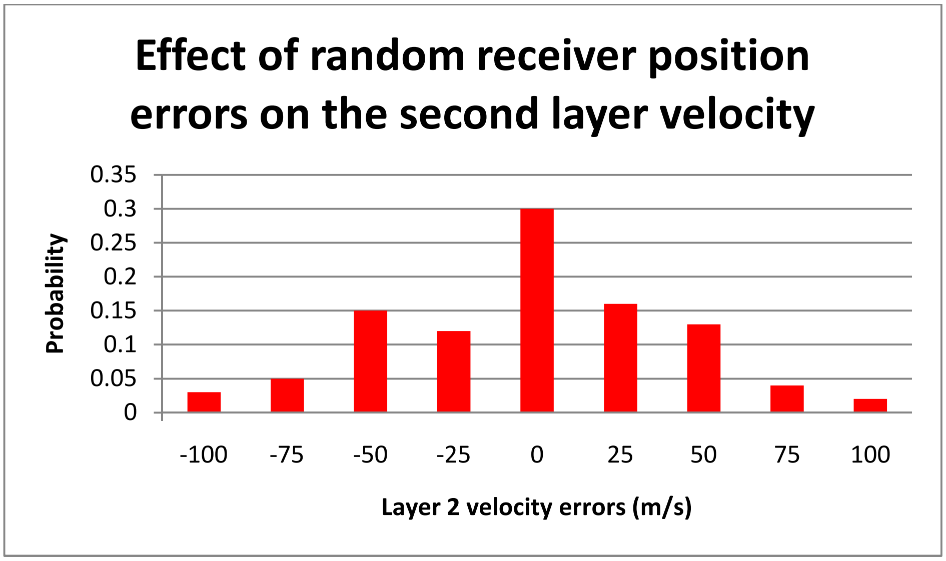


Figure (5.34): Histogram showing errors in the velocity of the second layer with their probabilities after introducing random errors to the receiver positions.

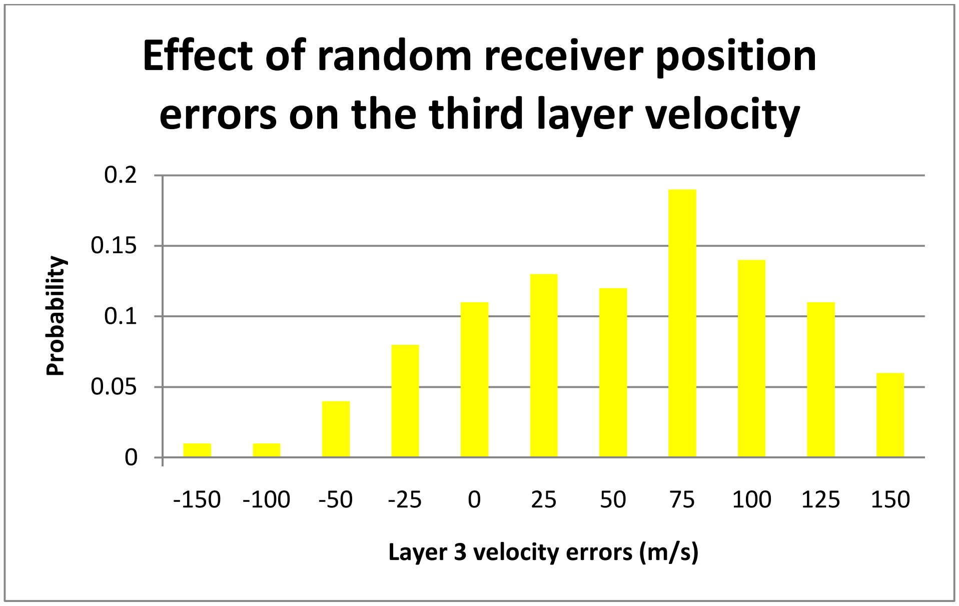


Figure (5.35): Histogram showing errors in the velocity of the third layer with their probabilities after introducing random errors to the receiver positions.

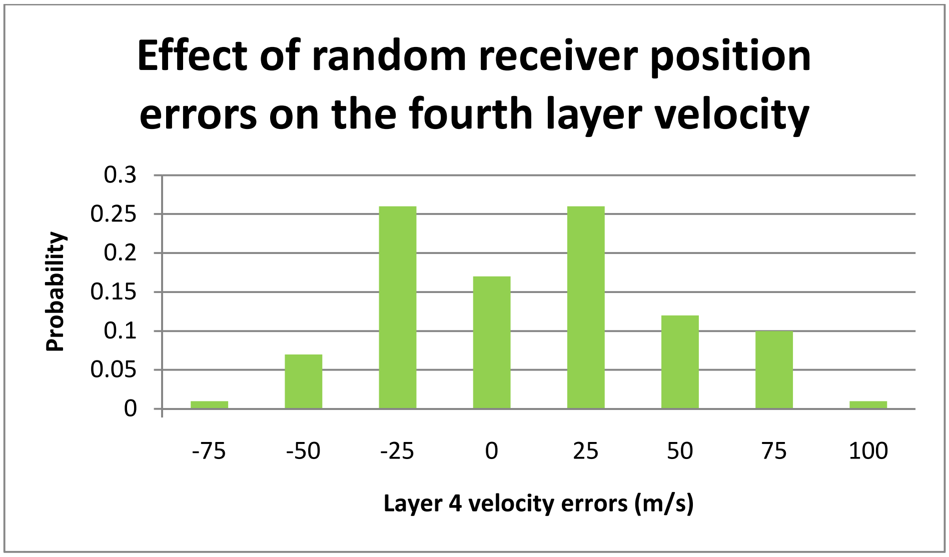


Figure (5.36): Histogram showing errors in the velocity of the fourth layer with their probabilities after introducing random errors to the receiver positions.

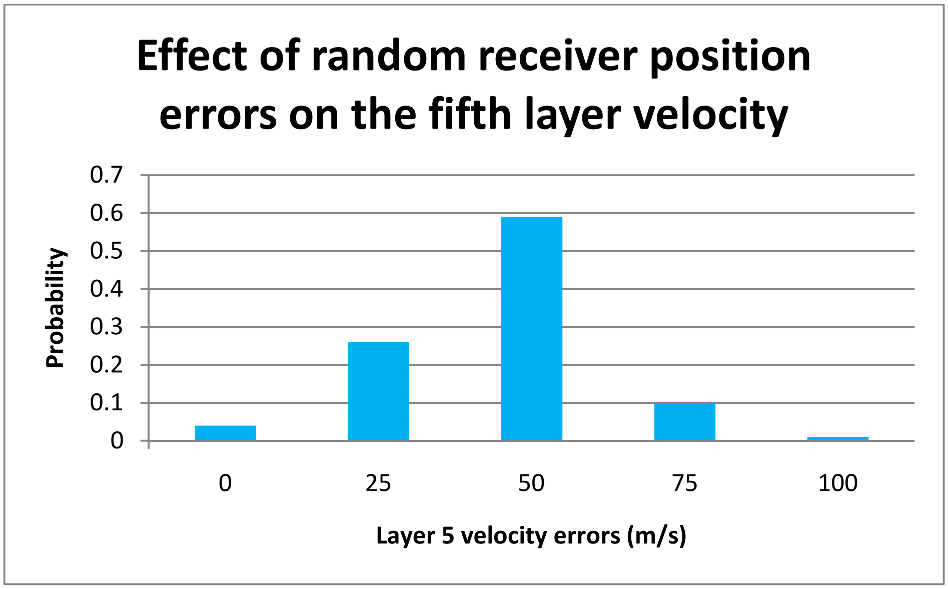


Figure (5.37): Histogram showing errors in the velocity of the fifth layer with their probabilities after introducing random errors to the receiver positions.

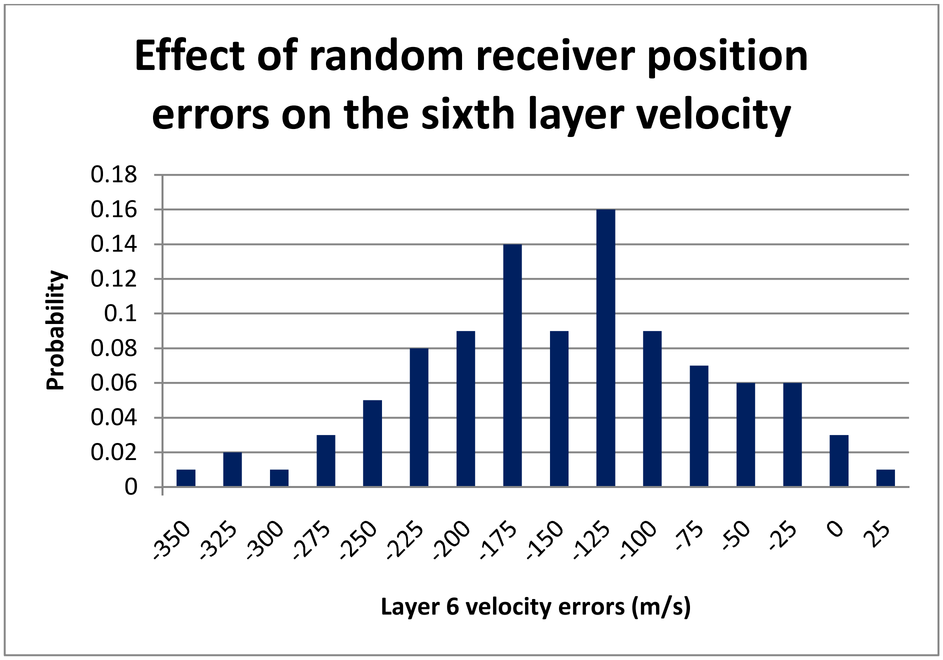


Figure (5.38): Histogram showing errors in the velocity of the sixth layer with their probabilities after introducing random errors to the receiver positions.

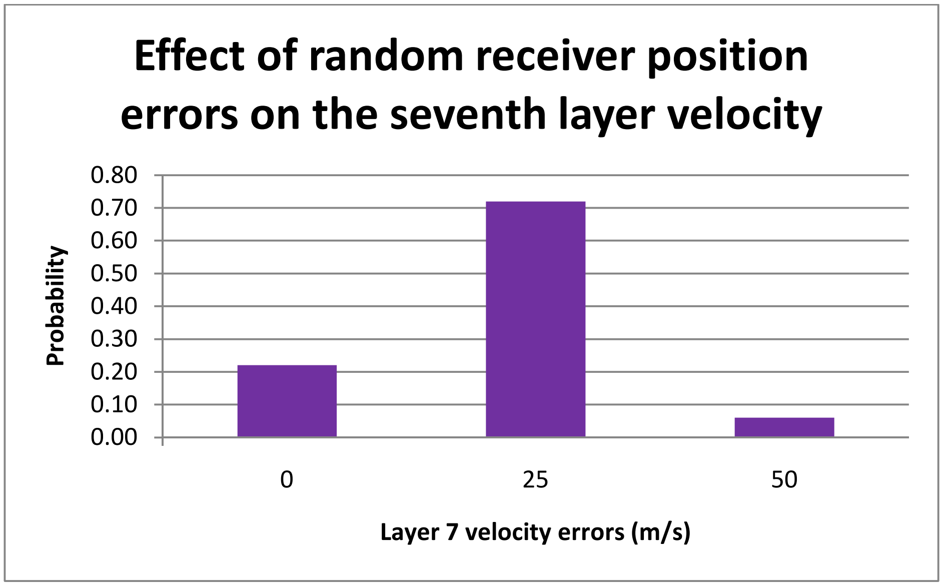


Figure (5.39): Histogram showing errors in the velocity of the seventh layer with their probabilities after introducing random errors to the receiver positions.

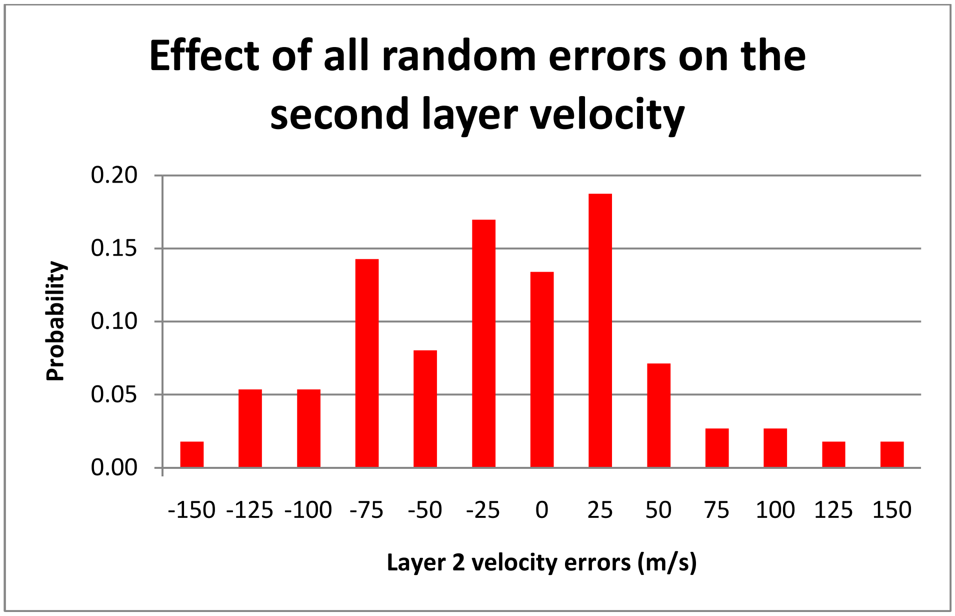


Figure (5.40): Histogram showing errors in origin time with their probabilities after introducing random errors to the 115 variables.

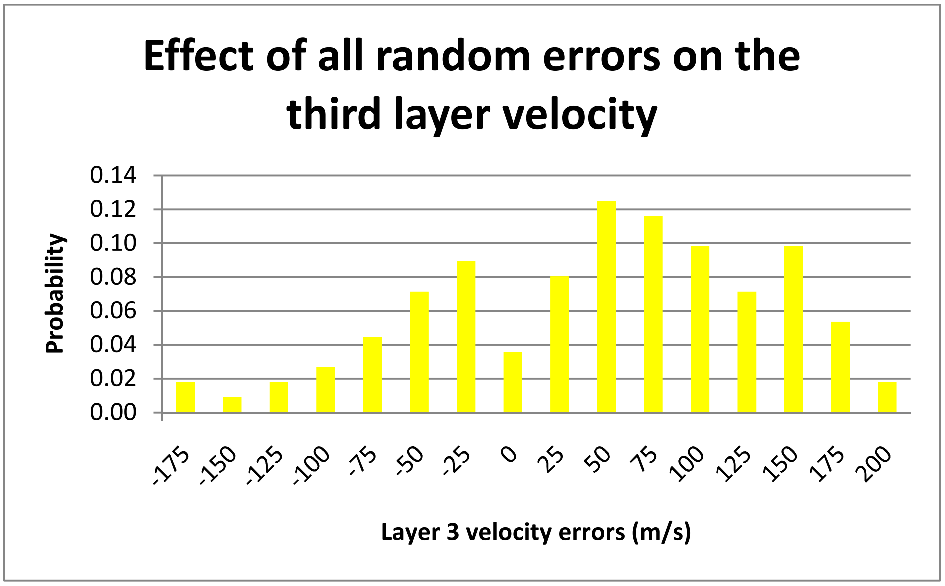


Figure (5.41): Histogram showing errors in hypocentre x-coordinate with their probabilities after introducing random errors to the 115 variables.

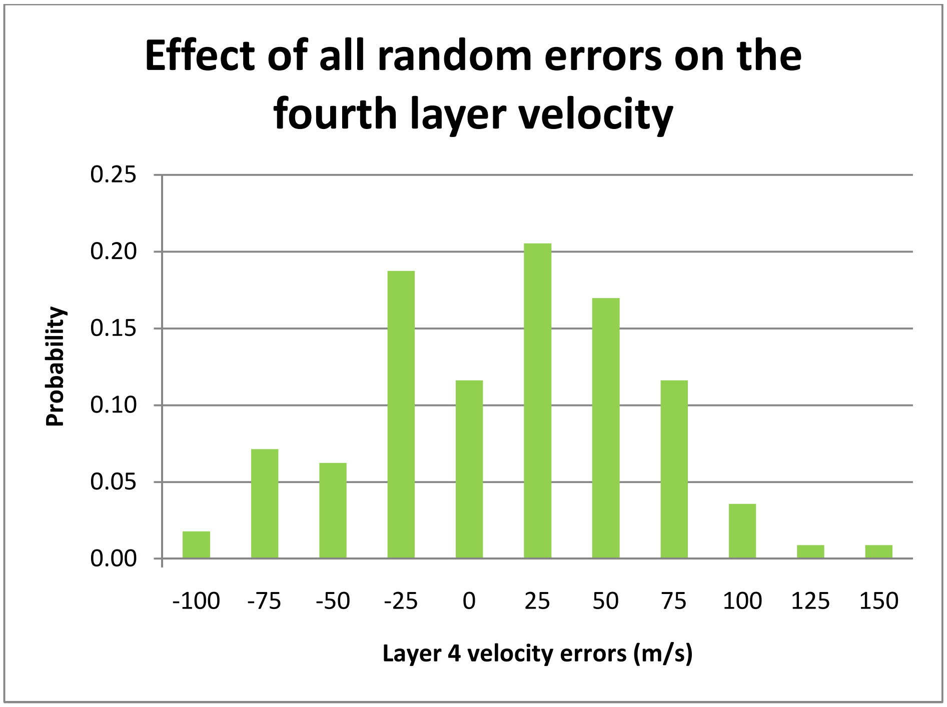


Figure (5.42): Histogram showing errors in hypocentre y-coordinate with their probabilities after introducing random errors to the 115 variables.

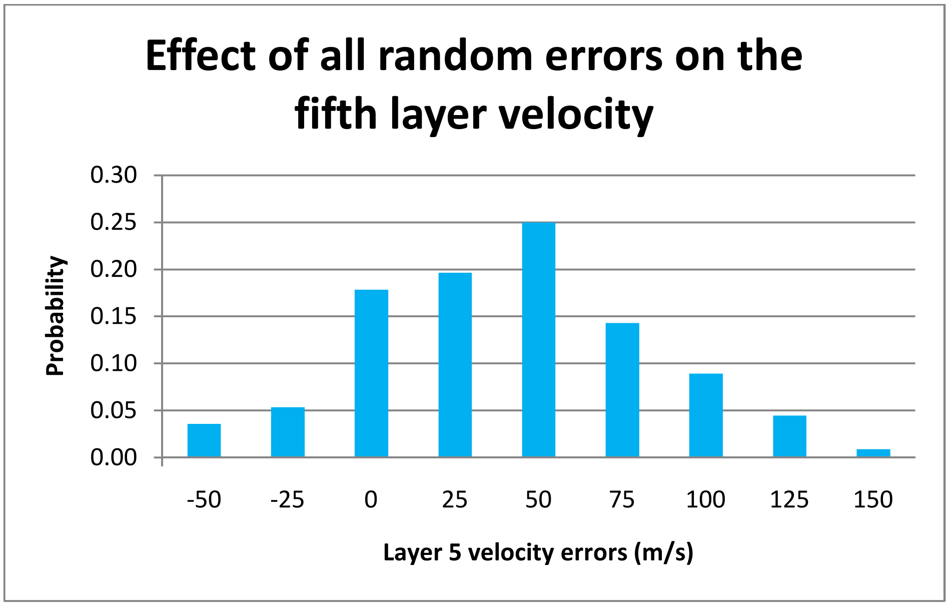


Figure (5.43): Histogram showing errors in hypocentre z-coordinate with their probabilities after introducing random errors to all 115 variables.

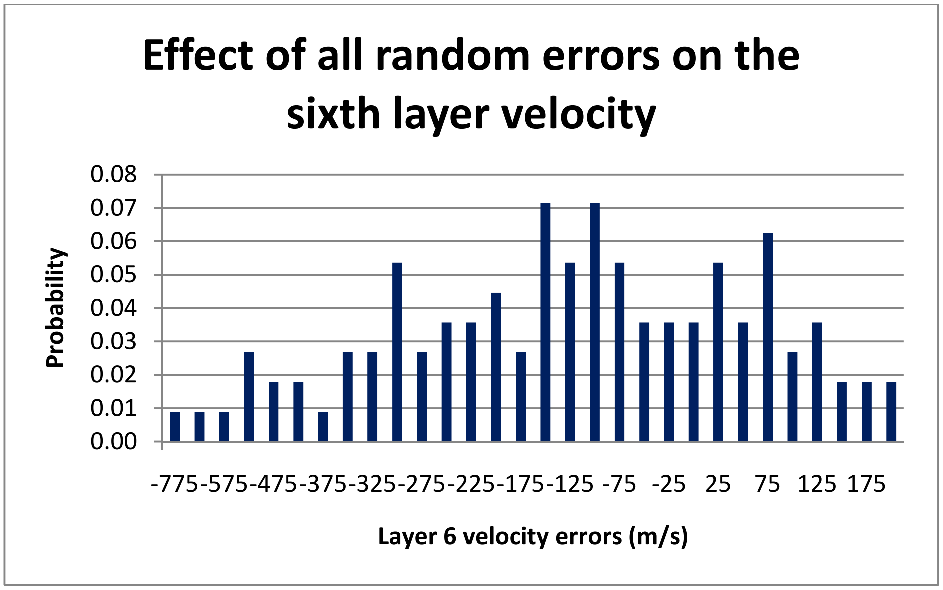


Figure (5.44): Histogram showing errors in hypocentre z-coordinate with their probabilities after introducing random errors to all 115 variables.

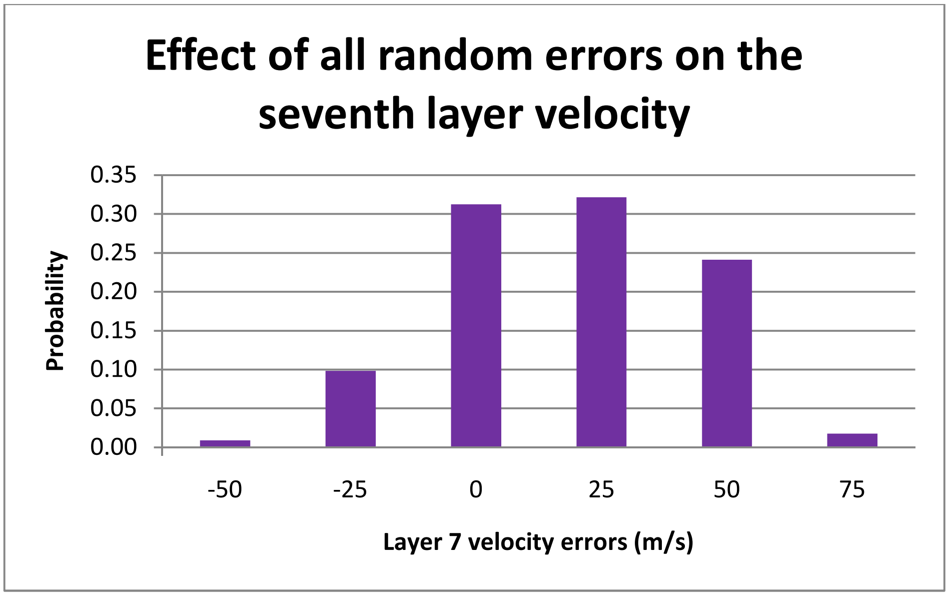


Figure (5.45): Histogram showing errors in hypocentre z-coordinate with their probabilities after introducing random errors to all 115 variables.

5.8 Conclusion

I have developed a set of algorithms written as object oriented C++ codes within Microsoft Visual Studio platform to use arrival times of induced micro-seismic events within a hydrocarbon reservoir for imaging. I assume that the micro-seismic events positions and origin times are known together with the receiver stations coordinates. I have shown through the results that the reconstructed velocity model is very accurate compared to the true model, for error free data. An error analysis is presented to quantify the sensitivities of the algorithm to such errors on the velocity parameters.

The methods and the algorithms presented in this chapter can be very practical when applied to VSP data as the coordinates for the sources and receiver stations are known. The accuracy of such algorithms is in a direct function of the sources and receiver position coordinates. Observed data – picked arrival traveltimes – is also equally important and related to the errors in the reconstructed slowness model. However, the initial slowness model parameters may not be important as I have shown. In Chapter 6, I will present the case when all the parameters of the equation (1.1) including the source location and origin time are unknown.

References

Berryman, J.G., 1990, Stable iterative reconstruction algorithm for nonlinear traveltime tomography: Inverse Problems, **6**, 2142.

—, 2000a, Analysis of Approximate Inverses in Tomography I. Resolution Analysis of Common Inverses: Optimization and Engineering, **1**, 87–115.

—, 2000b, Analysis of Approximate Inverses in Tomography II. Iterative Inverses: Optimization and Engineering, **1**, 437–473.

Lehmann, B., 2007, Seismic traveltime tomography for engineering and exploration applications: EAGE Publications bv

Menke, W., 1984, Geophysical Data Analysis: Discrete Inverse Theory: Academic Press, Inc.

Nadri, D., 2008, Joint non-linear inversion of amplitudes and travel times in a vertical transversely isotropic medium using compressional and converted shear waves, Curtin University of Technology.

Ravindran, A., K.M. Ragsdell, and G.V. Reklaitis, 2006, Engineering optimization: methods and applications: John Wiley & Sons, Inc.

Sambridge, M., 1998, Exploring multidimensional landscapes without a map: Inverse Problems, **14**, 427-440.

Sen, M.K., and P.L. Stoffa, 1995, Global optimization methods in geophysical Inversion: ELSEVIER.

Snieder, R., 1990, A perturbative analysis of non-linear inversion: Geophys. J. Int., **101**, 545-556.

—, 1998, The role of nonlinearity in inverse problems: Inverse Problems, **14**, 387–404.

Tarantola, A., 2005, Inverse problem Theory and Methods for Model Estimation: Society for Industrial and Applied Mathematics.

Trampert, J., 1998, Global seismic tomography: the inverse problem and beyond: Inverse Problems, **14**, 371–385.

Every reasonable effort has been made to acknowledge the owners of copyright material. I would be pleased to hear from any copyright owner who has been omitted or incorrectly acknowledged.

Chapter 6

Source Location, Origin Time, Ray-path and Slowness Model

6.1 Overview

The main topic of this chapter is to implement all the techniques and algorithms developed in the previous chapters to solve for the four unknowns in equation (1.1) for different settings. Given a 3D discretized medium and arrival times for the micro-seismic events recorded by receiver stations with their positions coordinates, I determine the unknowns in iterative two-stage methodology. First, I generate estimated locations for all the micro-seismic events together with their origin times for initial slowness model parameters. Second, I reconstruct new slowness model parameters from the estimated sources locations and the initial slowness model parameters for the hydrocarbon reservoir. These new slowness model parameters are used to generate new estimated location positions for all the micro-seismic events together with their origin times and then I switch to stage two to update the slowness model parameters. The algorithm is iterated until convergence is reached. The method is very robust when the initial model is close to the true solution.

6.2 Parameters Estimation

I started with a 3D medium representing a hydrocarbon reservoir. The medium is a cube with 500m on each side. I used a grid size of 5m to discretize the 3D medium into regular cells according to section 3.7.2. The resulting grid dimensions of the 3D medium are $N_x = N_y = N_z = 101$. The initial velocity model was set to 3750 m/s and was discretized with the same grid size to fit the discretized 3D medium. All the nodes were populated with their corresponding slowness parameters according to section 3.7.3.

I used three monitoring wells, each with 15 receivers, and eight micro-seismic events as shown in Figure (4.2). The geometry of such sources and receivers is described in section 4.2.4. The arrival times recorded at different receiver stations for different micro-seismic events are generated according to the micro-seismic event model, as described in section 4.2. Some 360 arrival times were obtained using the true velocity model listed in Table (4.1) and a two-point ray tracing algorithm.

The objective function, as described in section 4.2.2, for a micro-seismic event is constructed according to the algorithm I have developed in section 4.2.3. This algorithm makes extensive use of the forward modelling algorithm described in section 3.7.4. I then use a systematic grid search algorithm, described in section 4.2.3, to obtain the global minimiser of the objective function. This minimiser represents the micro-seismic event hypocentre. The origin time is obtained according to equation (4.2.9). The process is repeated for all the other micro-seismic events sequentially.

At this point, I have finished the first iteration for the first stage. I started with initial slowness model, observed arrival times and known positions of the receiver stations to generate estimates of the hypocentres and origin times for all the micro-seismic events.

Now, I switch to the second stage to update the slowness model. The objective function and its gradient for this stage are described in section 5.3. They are constructed according to the algorithm I have developed in section 5.4. This algorithm depends heavily on the forward modelling algorithm. The reconstructed slowness model is described in section 5.5. I use the variable metric method with the BFGS updating formula, as described in section 2.4.2.5, to compute the slowness update according to the algorithm I have developed in section 5.5. The optimisation algorithm executes enough times for updating the slowness model parameters. In addition, the slowness model parameters algorithm is iterated 5 times before switching to the first stage. This is required to induce sufficient update in the slowness model parameters. The new slowness model parameters replace the current one. This concludes the second stage and hence the first iteration.

The second iteration starts at the first stage starting with the current slowness model and the positions of the receiver stations together with the picked arrival times. A new set of hypocentres and origin times for all the micro-seismic events are obtained according to the current slowness model. New slowness model parameters are reconstructed in the second stage using the new hypocentres and origin times. This ends the second iteration. The procedure is repeated until an acceptable match between observed data and computed synthetics is achieved.

The above described integrated framework depends on incremental updates during each stage for each iteration. The first stage may produce more accurate results as it is designed to obtain the global minimiser. Further, it uses the l_1 norm which is relatively immune against the outliers. I have developed the framework in C++ as object oriented codes within Microsoft Visual Studio platform.

Figure (6.1) illustrates the classes that I have used in the first stage. There are three classes together with their required input and the relationships. The diamond head indicates a composition relationship between the classes while the arrow head indicates required input to the classes. The classes' attributes and operations are not shown. I refer to this package as source parameters optimisation package.

Figure (6.2), on the other hand, illustrates the classes that I have used in the second stage. There are four classes together with their required input and relationships. Similarly, the diamond head indicates a composition relationship between the classes while the arrow head indicates required input to the classes. The classes' attributes and operations are not shown. I refer to this package as slowness model parameters optimisation package. Figure (6.3) shows the complete framework.

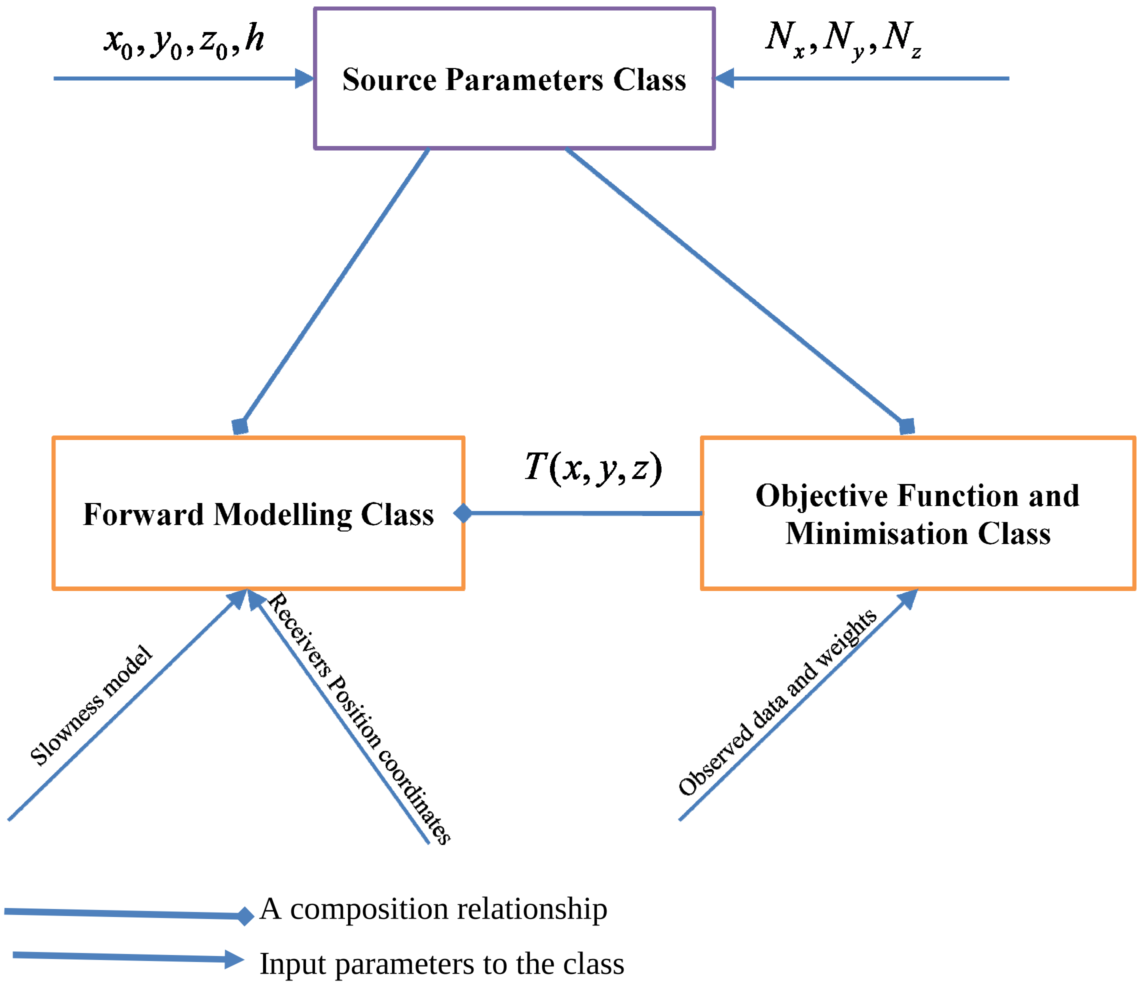

x_0, y_0, z_0, h
Source Parameters Class
N_x, N_y, N_z
Forward Modelling Class
$T(x, y, z)$
Objective Function and Minimisation Class
Slowness model
Receivers Position coordinates
Observed data and weights
A composition relationship
Input parameters to the class

Figure (6.1): A schematic diagram showing all the different classes of the source parameters optimisation package. Attributes and operations are not shown. The parameters x_0, y_0, z_0 refers to the original position of the 3D model while h is the grid spacing. The numbers N_x, N_y, N_z are the number of nodes along the Cartesian coordinate system.

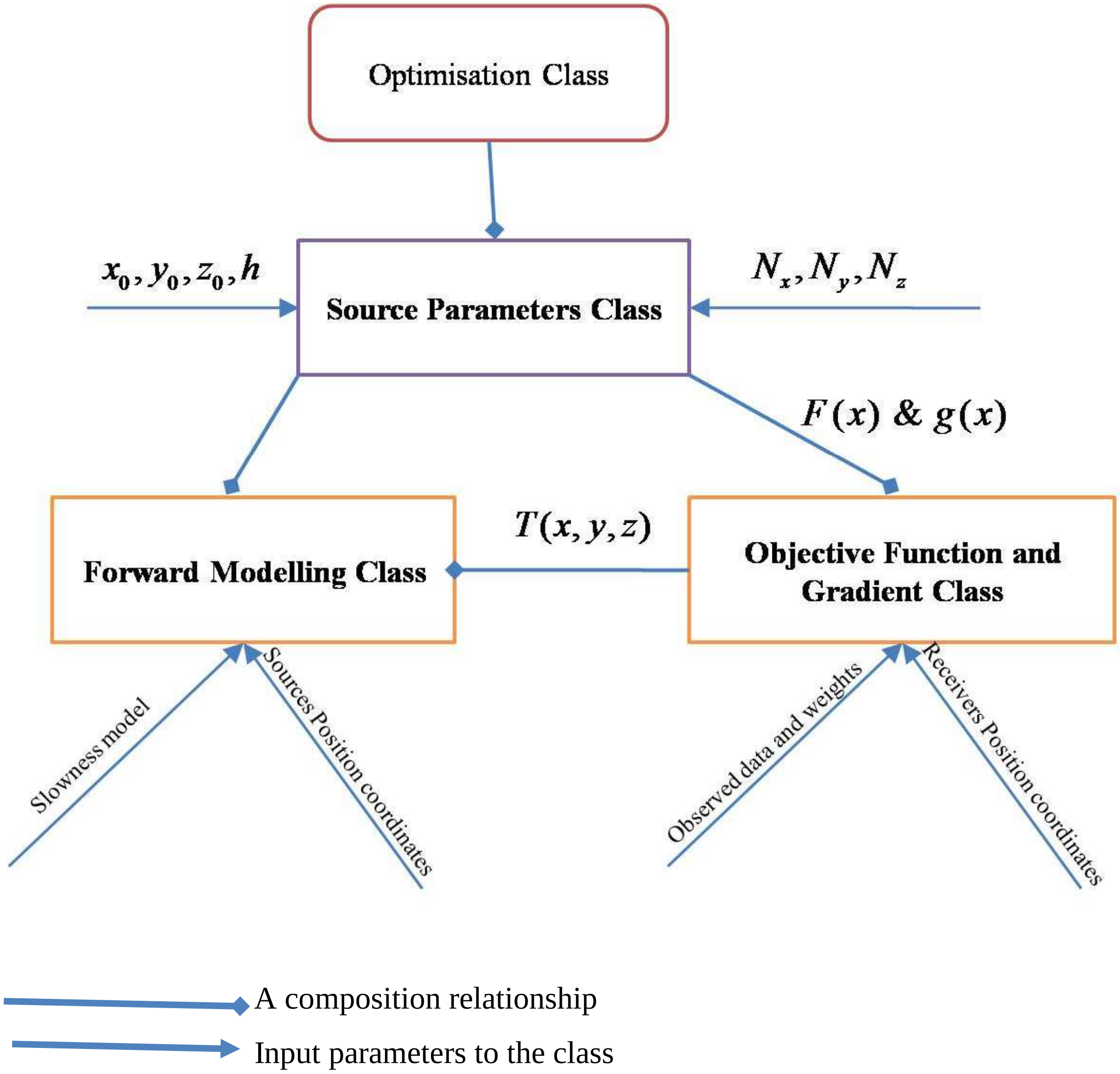

Optimisation Class
x_0, y_0, z_0, h
Source Parameters Class
N_x, N_y, N_z
$F(x)$ & $g(x)$
Forward Modelling Class
$T(x,y,z)$
Objective Function and Gradient Class
Slowness model
Sources Position coordinates
Observed data and weights
Receivers Position coordinates
A composition relationship
Input parameters to the class

Figure (6.2): A schematic diagram showing all the different classes of the slowness model parameters optimisation package.

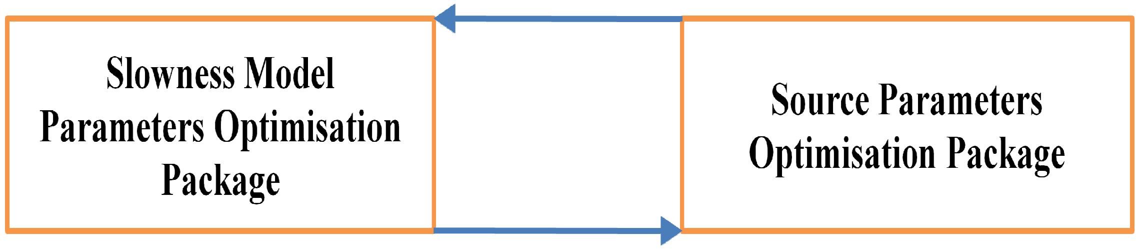

Slowness Model Parameters Optimisation Package
Source Parameters Optimisation Package

Figure (6.3): A schematic diagram showing the complete framework.

6.2.1 Parameter Estimation for Known Boundaries with *a priori* Velocity Model

In this section, I present the results when the formations' boundaries are known together with *a priori* velocity model. The objective function for this case is described in section 2.3.3. This will result in seven unknown parameters for the slowness model. Still, we need to obtain only six parameters as there are no traces passing through the first layer. The number of the events' parameters remains 32 parameters including 24 position coordinates and 8 origin times. I set the constraining velocity model to the initial velocity model. In general, a producing hydrocarbon reservoir is investigated thoroughly leading to a quite accurate initial velocity model.

The estimations of the hypocentre parameters for all the eight micro-seismic events are shown in Figures (6.4), (6.5) and (6.6). The hypocentre parameters were obtained accurately because the initial velocity was an average of the true velocity model parameters. The hypocentre parameters remain fixed despite the fluctuation in the velocity model parameters. The x-coordinate and the y-coordinate parameters are exact while the z-coordinate parameters have ±5m error. The origin time parameters for all the eight events were also obtained accurately and the results are shown in Figure (6.7). Still, the origin time parameters kept fluctuating until they reach the fourth iteration. That is, the cross-talk is between the origin time parameters and the velocity parameters.

The estimation of the velocity model parameters are shown in Figure (6.8). The analysis, described in section 5.6, is applied here with a minor change that the objective function consists of two parts: the data part and the model part as described in section 2.3. After 8 iterations, the algorithm has converged to the true velocity parameters with ±1% error, except for layer 6. The slowness model parameters optimization package generates five iterations after the source parameter optimization package has finished one iteration.

Therefore, there are 25 iterations corresponding to the slowness parameters optimization against 5 iterations corresponding to source parameters optimization.

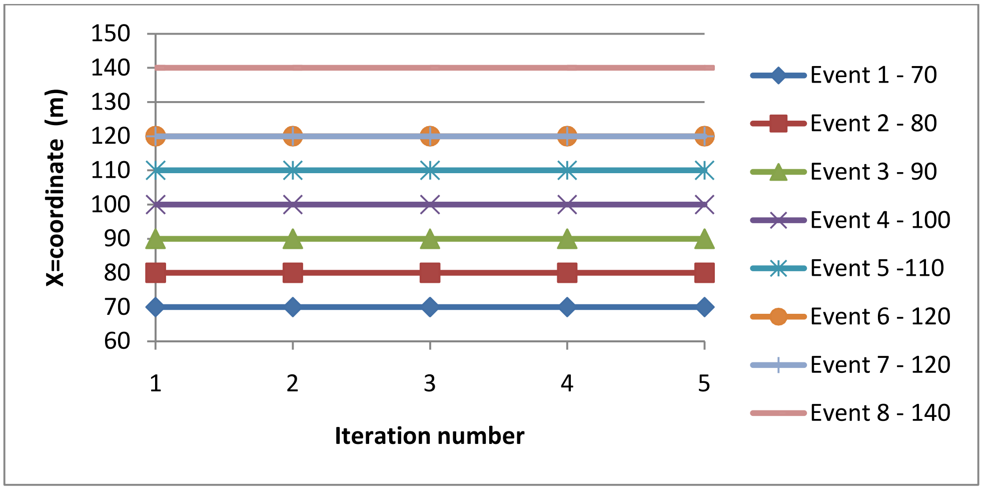


Figure (6.4): Plots representing the values of the x-coordinate parameters for the eight micro-seismic events for five iterations. The true values are listed to the right of the graph referenced with their corresponding microseismic event number.

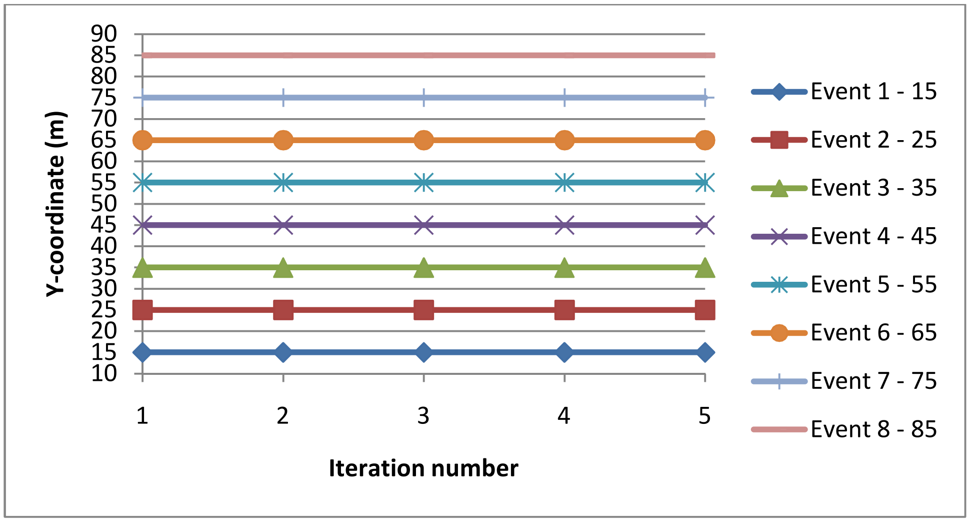


Figure (6.5): Plots representing the values of the y-coordinate parameters for the eight micro-seismic events for five iterations. The true values are listed to the right of the graph referenced with their corresponding microseismic event number.

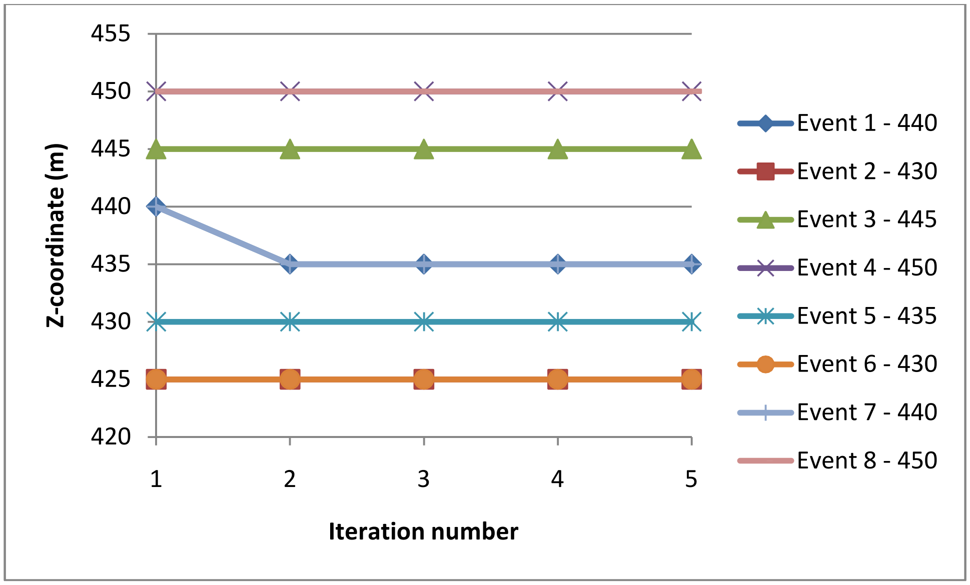


Figure (6.6): Plots representing the values of the z-coordinate parameters for the eight micro-seismic events for five iterations. The true values are listed to the right of the graph referenced with their corresponding microseismic event number.

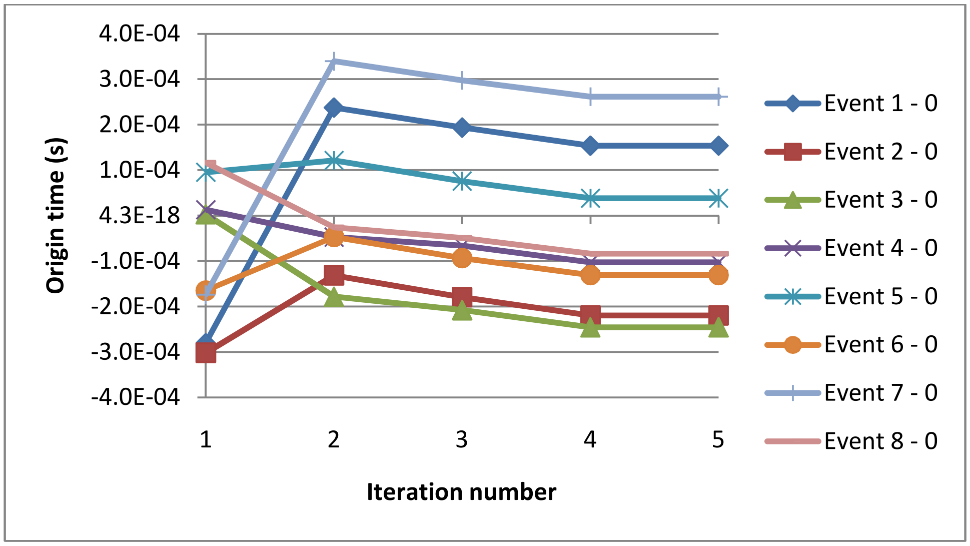


Figure (6.7): Plots representing the values of the origin time parameters for the eight micro-seismic events for five iterations. The true values are listed to the right of the graph referenced with their corresponding microseismic event number.

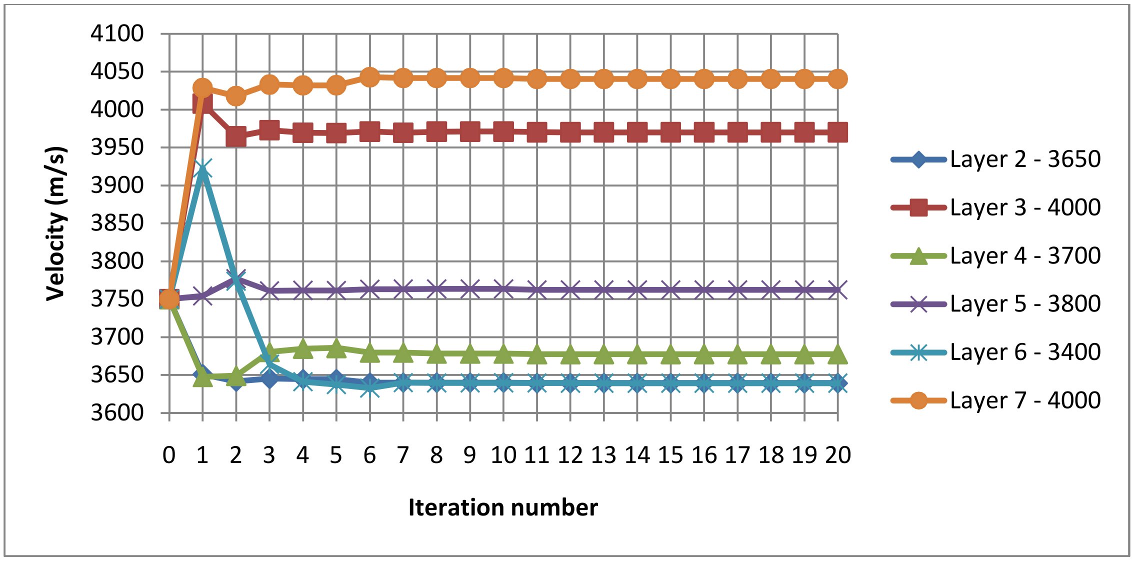


Figure (6.8): Plots representing the values of the velocity model parameters for the six formations for 20 iterations. The true values are listed to the right of the graph referenced with their corresponding layer number.

6.2.2 Parameter Estimation for Known Boundaries

In this section, I present the results when the formations' boundaries are known with no a *priori* velocity model. Like the previous section, this will result in seven unknown parameters for the slowness model. Still, we need to obtain only six parameters as there are no traces passing through the first layer.

The estimations of the hypocentre parameters for all the eight micro-seismic events are shown in Figures (6.9), (6.10) and (6.11). The hypocentre parameters were obtained accurately as anticipated by the error analysis described in section 5.7. The x-coordinate and the y-coordinate parameters are exact while the z-coordinate parameters have ±5m error. The origin time parameters for all the eight events were also obtained accurately and the results are shown in Figure (6.12). The estimation of the velocity model parameters are shown in Figure (6.13). After 5 iterations, the algorithm has converged to the true velocity parameters with ±1% error, except for layer 6. This suggests that the method is insensitive to the Bayesian constraints when the layer boundaries are known.

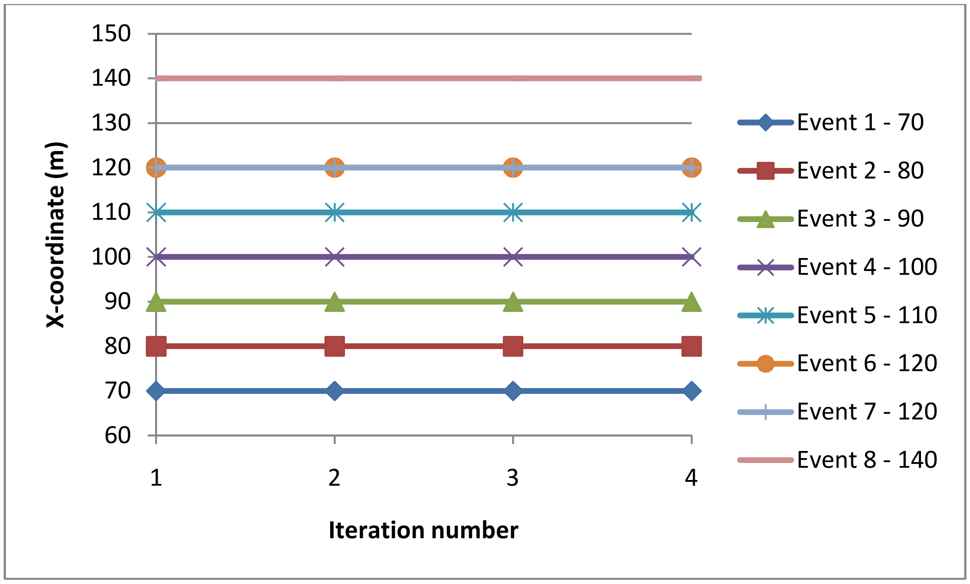


Figure (6.9): Plots representing the values of the x-coordinate parameters for the eight micro-seismic events for five iterations. The true values are listed to the right of the graph referenced with their corresponding microseismic event number.

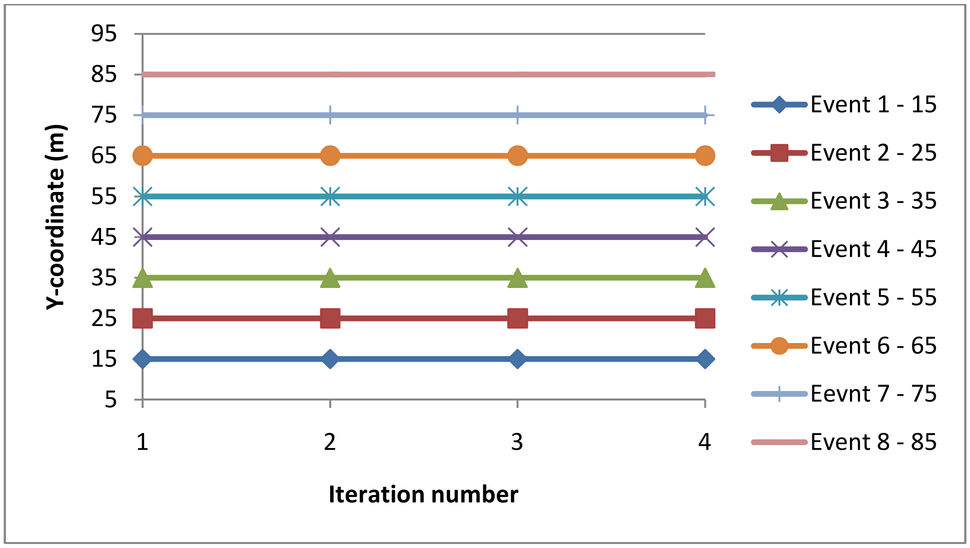


Figure (6.10): Plots representing the values of the y-coordinate parameters for the eight micro-seismic events for five iterations. The true values are listed to the right of the graph referenced with their corresponding microseismic event number.

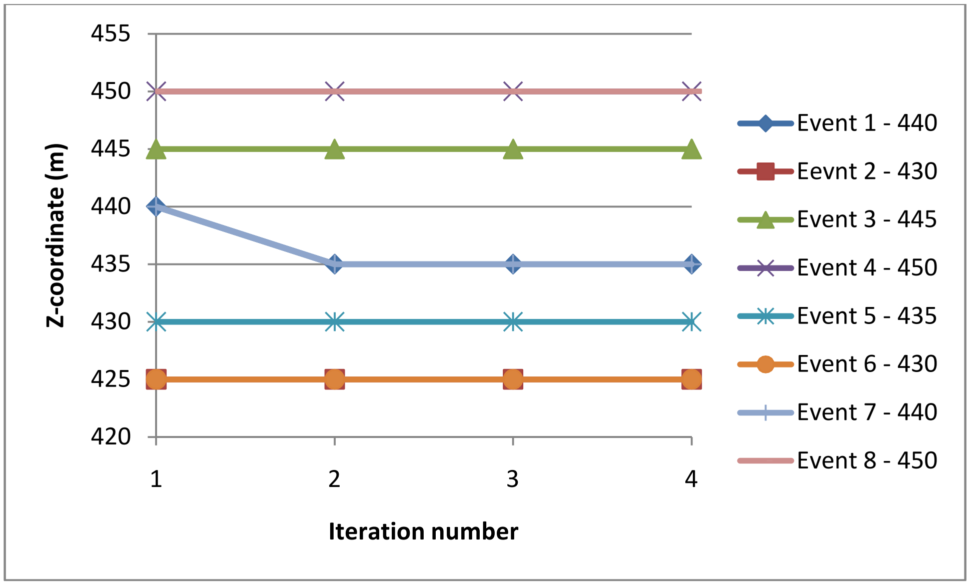


Figure (6.11): Plots representing the values of the z-coordinate parameters for the eight micro-seismic events for five iterations. The true values are listed to the right of the graph referenced with their corresponding microseismic event number.

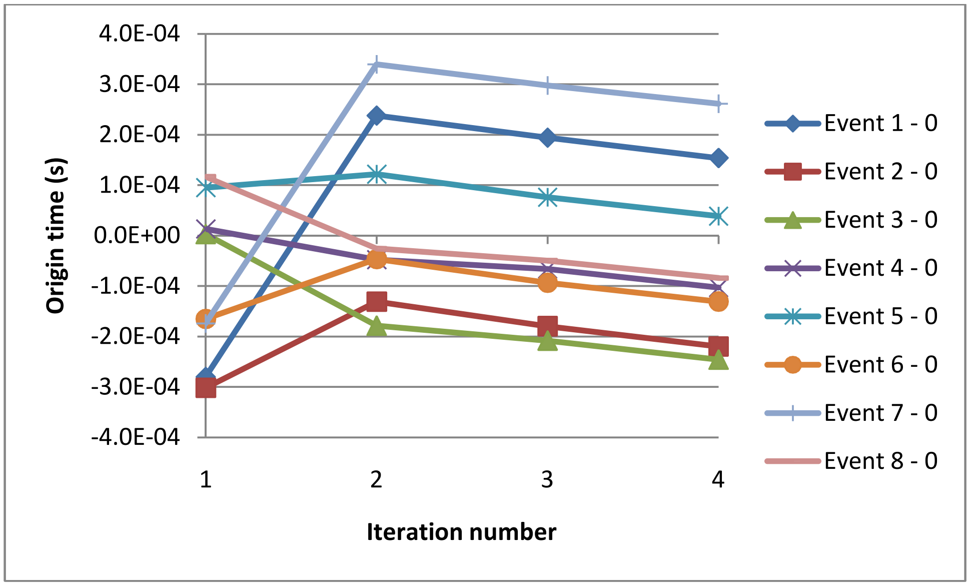


Figure (6.12): Plots representing the values of the origin time parameters for the eight micro-seismic events for five iterations. The true values are listed to the right of the graph referenced with their corresponding microseismic event number.

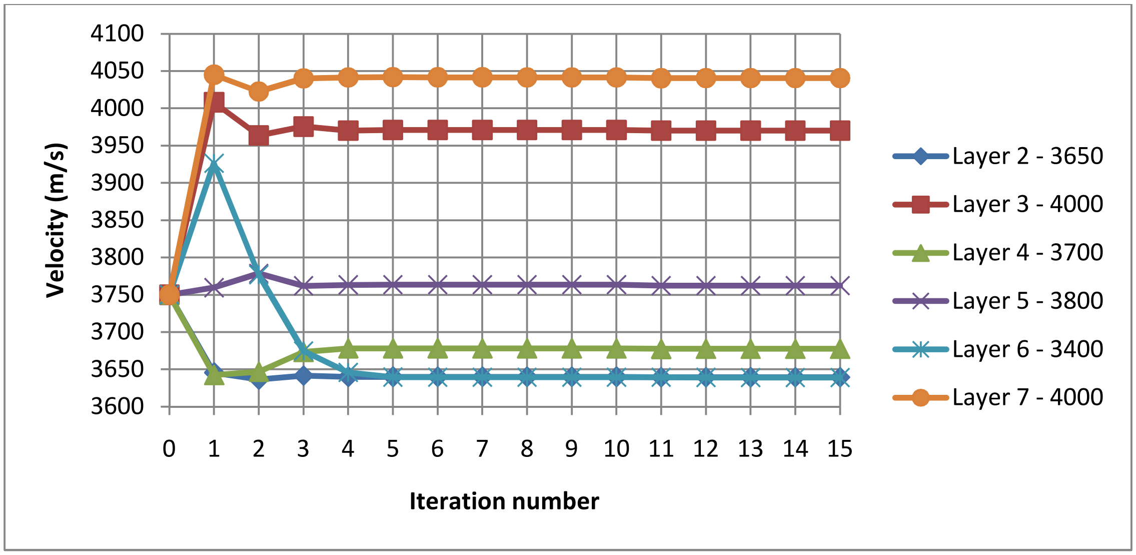


Figure (6.13): Plots representing the values of the velocity model parameters for the six formations for 20 iterations. The true values are listed to the right of the graph referenced with their corresponding layer number.

6.2.3 Parameter Estimation for Receiver Layer Dependent

In this section, I present the results as if the receivers were fixed at the boundaries such that neither the formations' boundaries nor the *a priori* velocity model is available. This will result in fifteen sub-layers with fifteen unknown parameters for the slowness model instead of six. Four parameters belong to the second layer, two parameters belong to the third layer, three parameters belong to the fourth layer, four parameters belong to the fifth layer, one parameter belongs to the sixth layer and one parameter belongs to the seventh layer. Each sub-layer has a height of 25m representing a separation between two consecutive receives.

The estimations of the hypocentre parameters for all the eight micro-seismic events are not shown as they are exactly the same as the previous results obtained in section 6.2.1 or section 6.2.2. The reason for this is the fact that the source parameters optimisation package uses the original discretization with a grid size of 5m. When I used the initial slowness model as a Bayesian constrain, I obtained similar results which are not shown.

Figure (6.14) shows the results for the second stage being the estimation of the velocity model parameters for layer 2. This figure has four plots with different colours corresponding to the four sub-layers within layer 2. After 8 iterations all the sub-layers' parameters belong to this layer converge satisfactorily to the true velocity value (3650) for the second layer. The last curve, representing sub-layer 4, has converged to a slightly different value. This behaviour is probably due to the effect from the next layer.

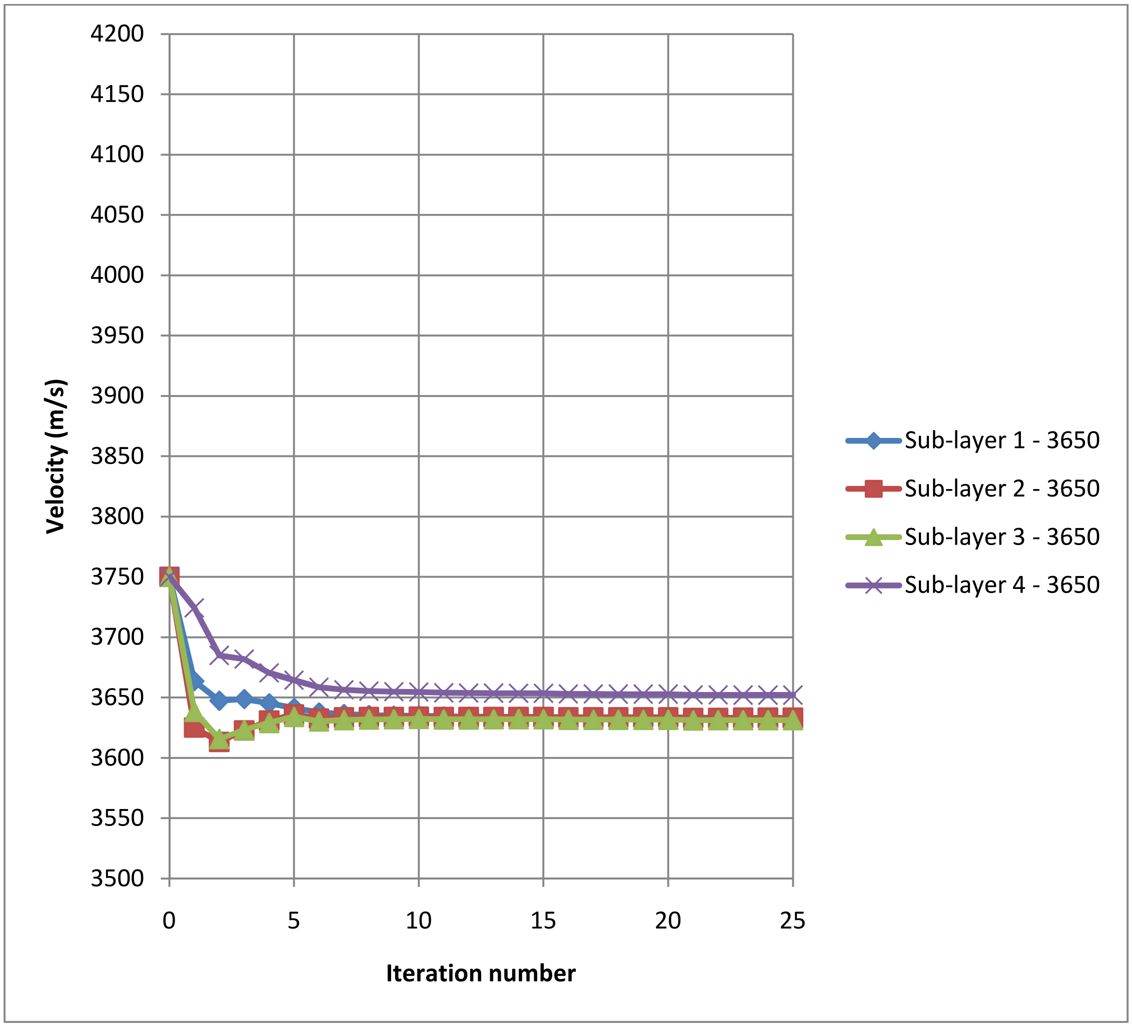


Figure (6.14): Plots representing the values of the velocity model parameters for the second layer. The true values are listed to the right of the graph referenced with their corresponding sub-layer number.

Figure (6.15) shows the results for the second stage being the estimation of the velocity model parameters for layer 3. This figure has two plots with different colours corresponding to the two sub-layers within layer 3. After 10 iterations all the velocity parameters belong to this layer converge closely to the true velocity parameters (4000).

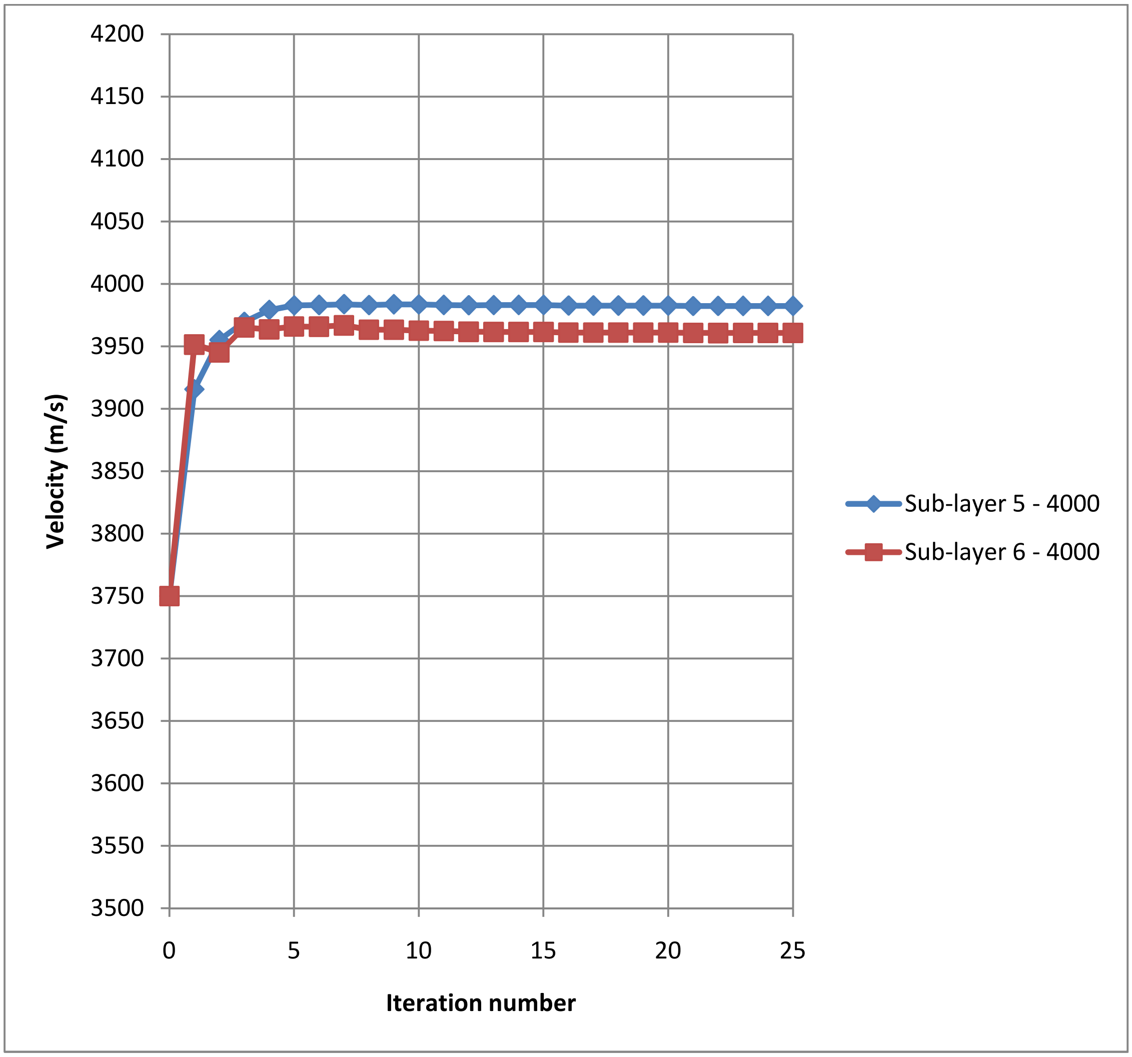


Figure (6.15): Plots representing the values of the velocity model parameters for the third layer. The true values are listed to the right of the graph referenced with their corresponding sub-layer number.

Figure (6.16) shows the results for the second stage being the estimation of the velocity model parameters for layer 4. This figure has three curves with different colours corresponding to the three sub-layers within layer 4. After 10 iterations all the parameters belong to this layer converge closely to the true velocity parameters (3700), except for sub-layer 9 which is affected slightly by layer 5.

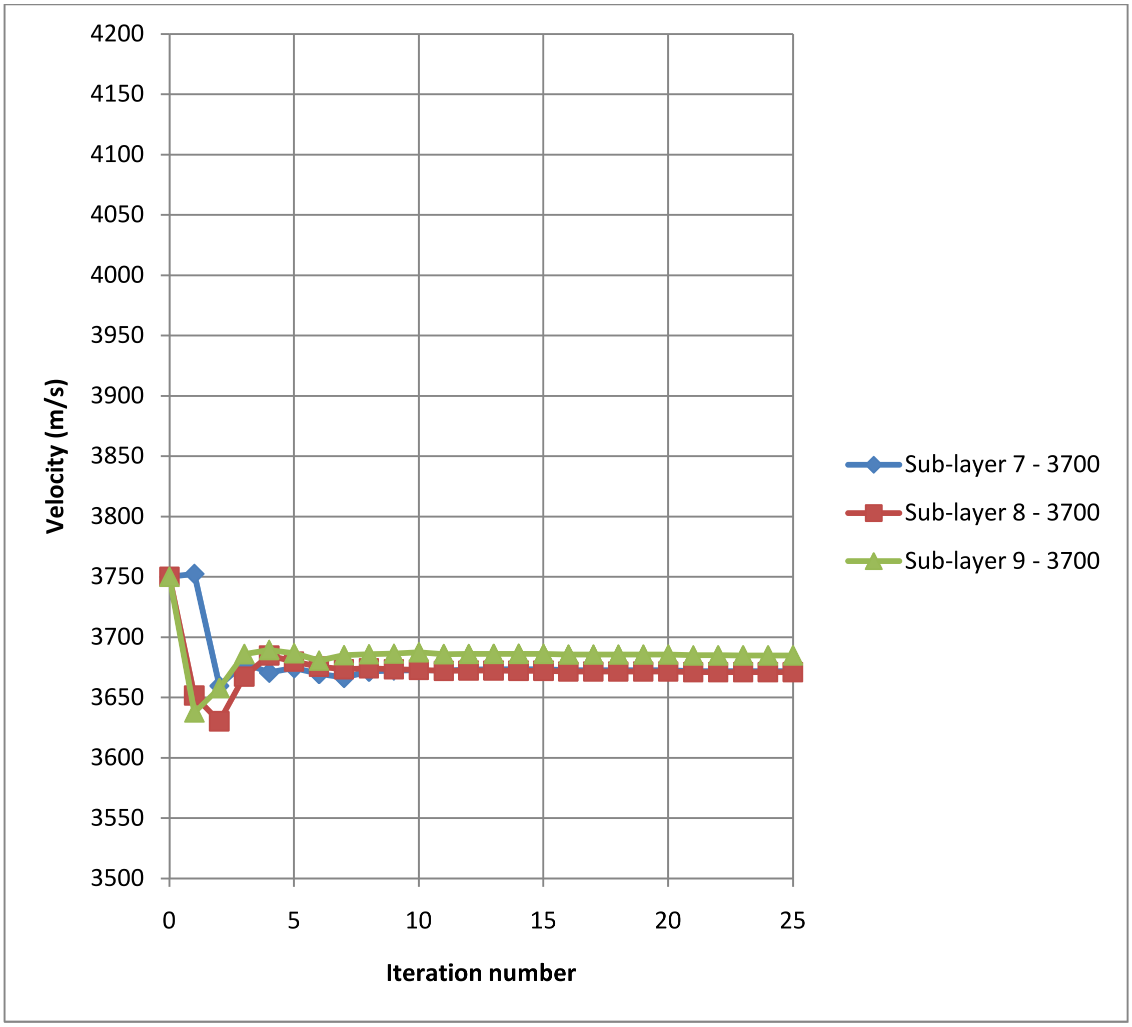


Figure (6.16): Plots representing the values of the velocity model parameters for the fourth layer. The true values are listed to the right of the graph referenced with their corresponding sub-layer number.

Figure (6.17) shows the results for the second stage being the estimation of the velocity model parameters for layer 5. This figure has four curves with different colours corresponding to the four sub-layers within layer 5. In general, after 10 iterations all the parameters belong to this layer converge coarsely to the true velocity parameter (3800), except the one that belongs to sub-layer 13. This strange behaviour is due to the effect from layer 6 resulted from the large velocity difference between layer 5 and layer 6.

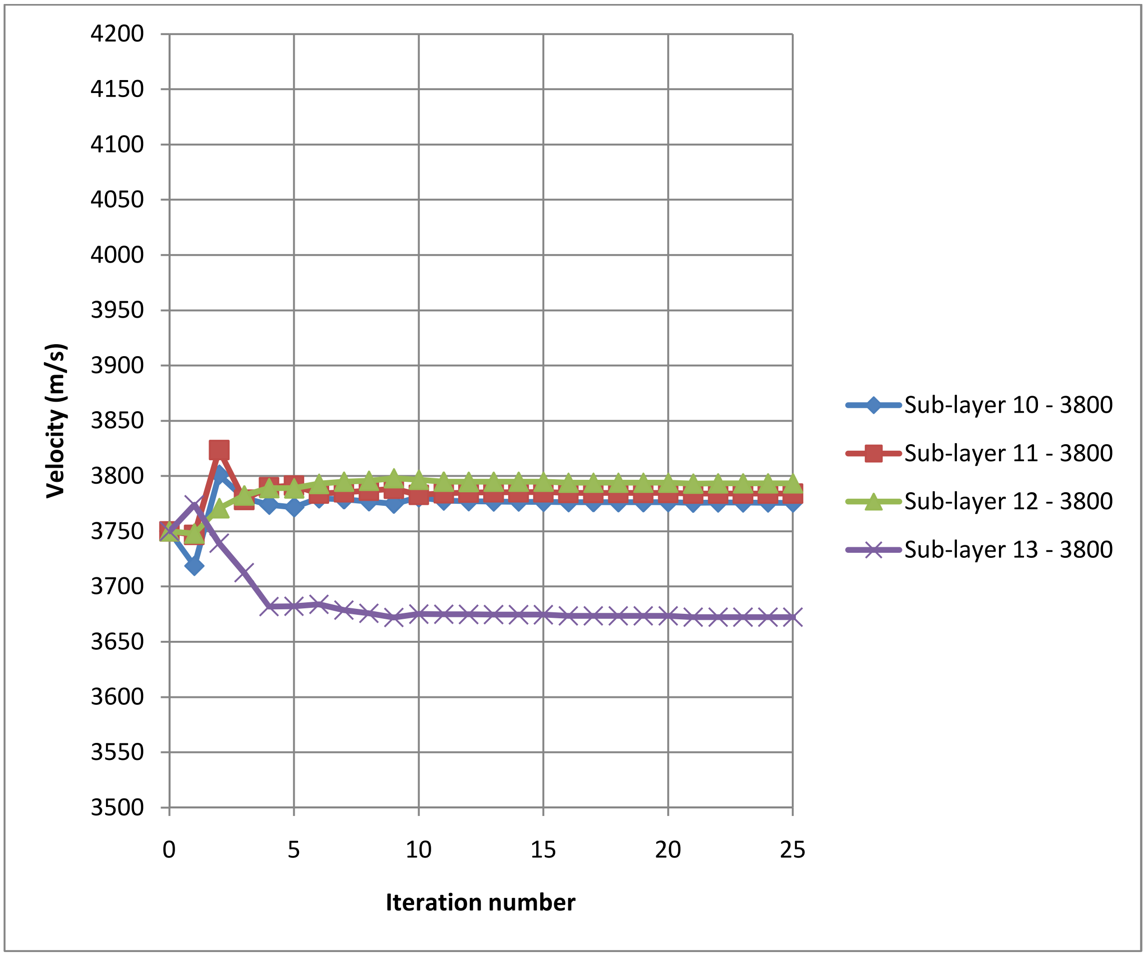


Figure (6.17): Plots representing the values of the velocity model parameters for the fifth layer. The true values are listed to the right of the graph referenced with their corresponding sub-layer number.

Figure (6.18) shows the results for the second stage being the estimation of the velocity model parameters for layer 6 and layer 7. This figure has two plots with different colours. The first curve corresponds to layer 6 while the second curve belongs to layer 7. After 10 iterations, the parameter that belongs to the sixth layer converges with 8% error. This large error is due to the fact that this layer contains the microseismic events and thus the rays that pass through this layer are nearly horizontal, as explained in Chapter 5. This causes the gradient to be insensitive to the velocity within layer 6. The second curve smoothly converges to the true velocity parameter (4000m/s) of layer 7 after 10 iterations.

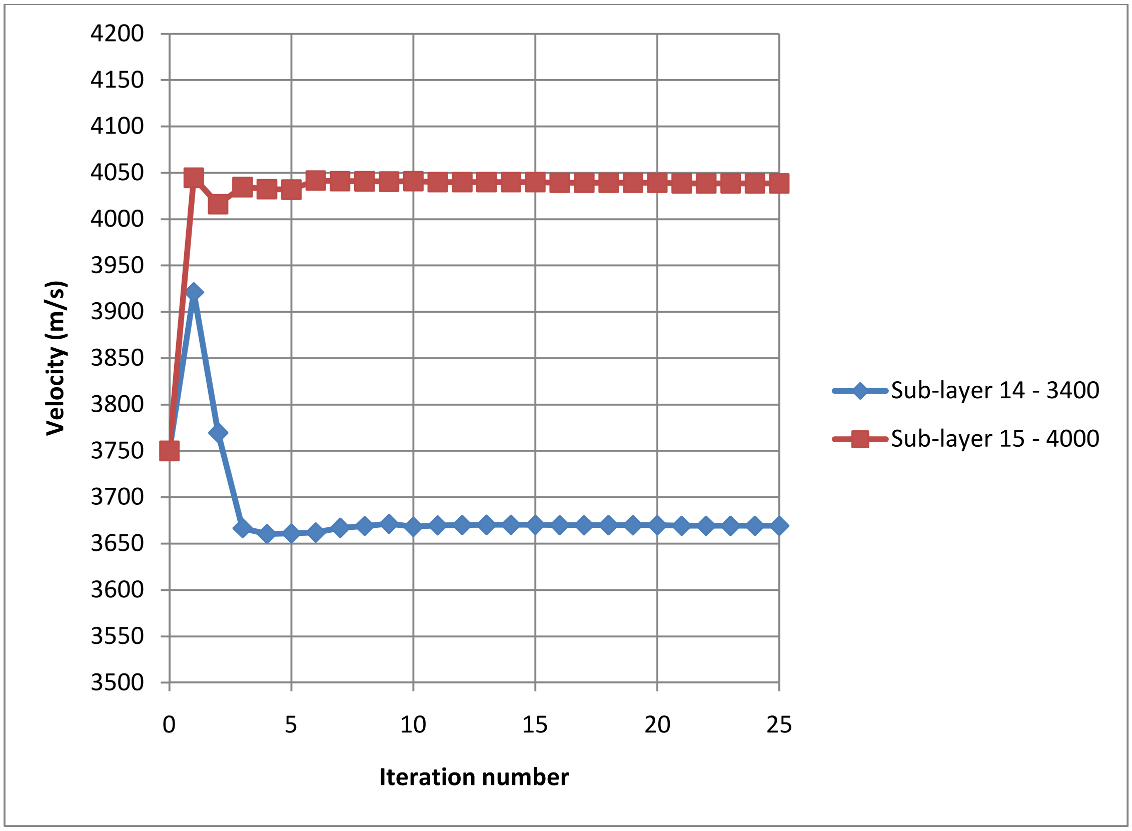


Figure (6.18): Plots representing the values of the velocity model parameters for both the fifth and the sixth layers. The true values are listed to the right of the graph referenced with their corresponding sub-layer number.

6.2.4 Parameter Estimation for Unknown Boundaries and *a priori* Velocity Model

In this section, I present the results when we have *a priori* velocity model, but the formations' boundaries are unknown. The model is divided into 101 horizontal beds such that each bed has a height of 5m representing cell height. The objective function for this case is described according to section 2.3.3. This configuration will result in 101 unknown parameters for the slowness model. Still, we need to obtain only 71 parameters as there are no traces passing through the remaining 30 beds. Layer 2 has 20 active beds, layer 3 has 10 beds, layer 4 has 15 beds, layer 5 has 20 beds, layer 6 has 6 beds and layer 7 has only one active bed. The number of the events' parameters remains 32 parameters including 24 position coordinates and 8 origin times. The *a priori* velocity model is set to the initial velocity model.

The estimations of the hypocentre parameters for all the eight micro-seismic events are shown in Figures (6.19), (6.20) and (6.21). The hypocentre parameters were obtained accurately as before since the optimization problem for the source parameters remains the same. The origin time parameters for all the eight events were also obtained accurately and the results are shown in Figure (6.22).

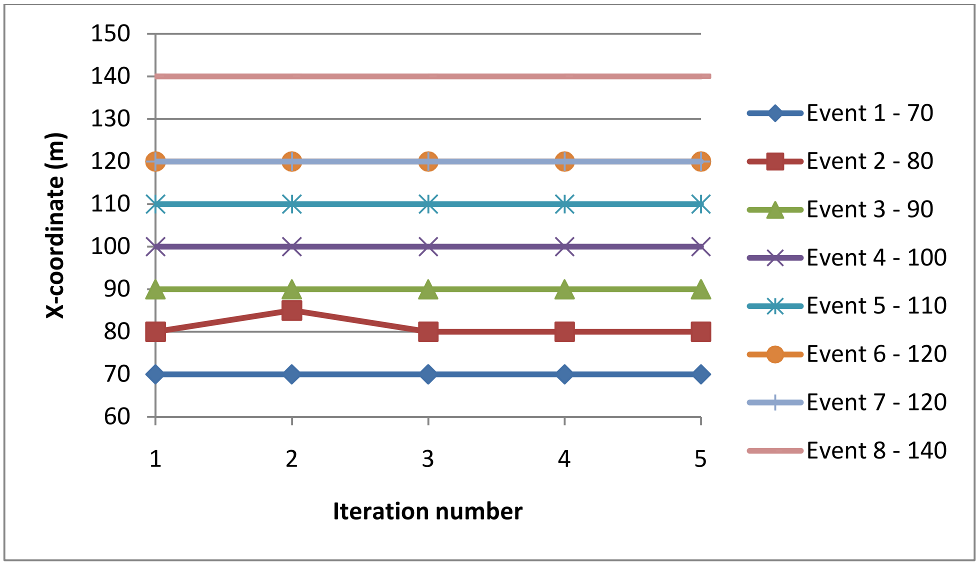


Figure (6.19): Plots representing the values of the x-coordinate parameters for the eight micro-seismic events for five iterations. The true values are listed to the right of the graph referenced with their corresponding microseismic event number.

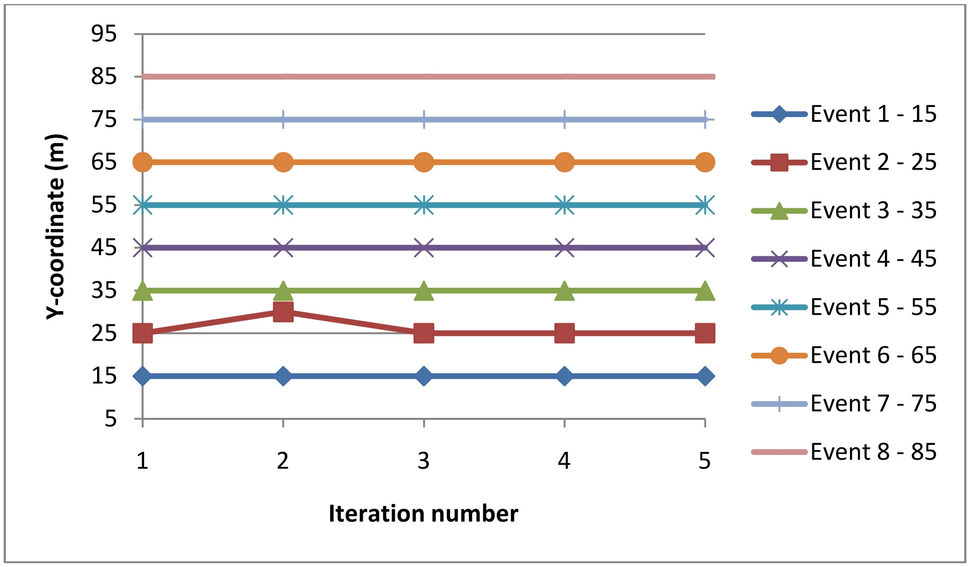


Figure (6.20): Plots representing the values of the y-coordinate parameters for the eight micro-seismic events for five iterations. The true values are listed to the right of the graph referenced with their corresponding microseismic event number.

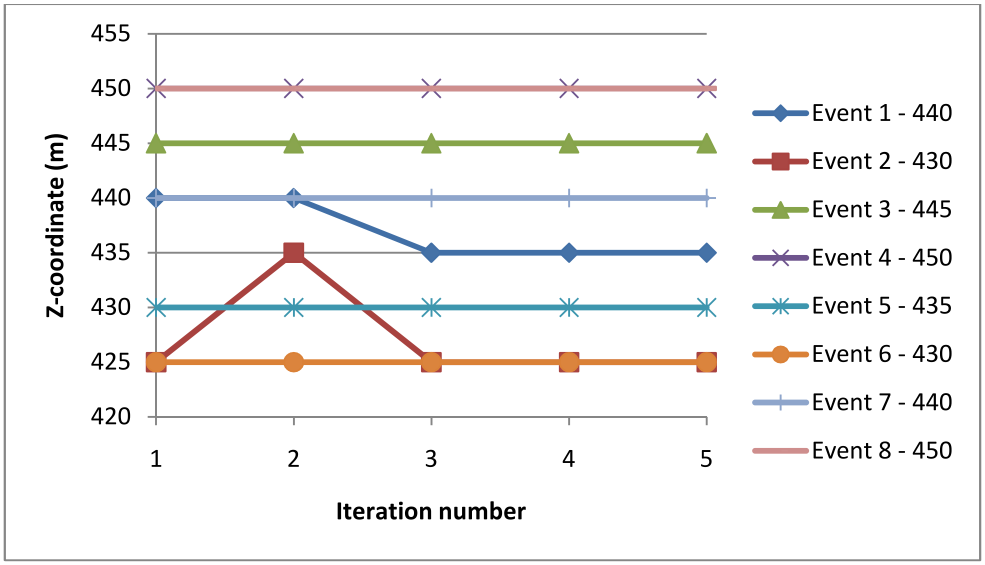


Figure (6.21): Plots representing the values of the z-coordinate parameters for the eight micro-seismic events for five iterations. The true values are listed to the right of the graph referenced with their corresponding microseismic event number.

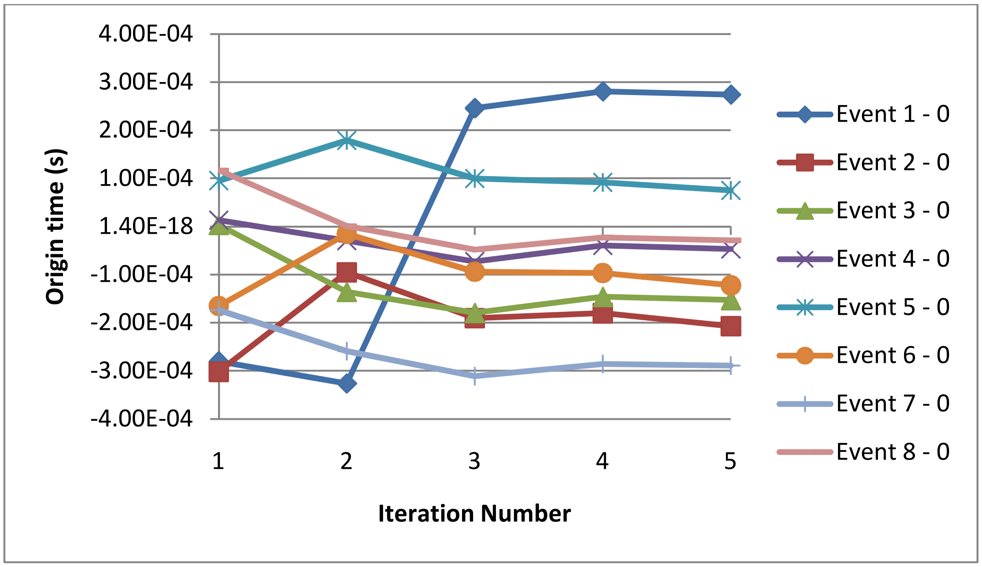


Figure (6.22): Plots representing the values of the origin time parameters for the eight micro-seismic events for five iterations. The true values are listed to the right of the graph referenced with their corresponding microseismic event number.

Figure (6.23) shows the results for the second stage being the estimation of the velocity model parameters for layer 2. This figure has twenty plots with different colours corresponding to twenty active beds (bed 20 to bed 39) within layer 2. After 15 iterations all the active beds' parameters belong to this layer converge satisfactorily to the true velocity of the second layer. The last five curves, representing beds 35, 36, 37, 38 and 39, converge to a slightly different value. This behaviour is probably due to the effect from the next layer as the velocity difference (350m/s) is reasonably high. Further, the horizontal resolution is 25m due to the receiver separation which has affected 5 beds.

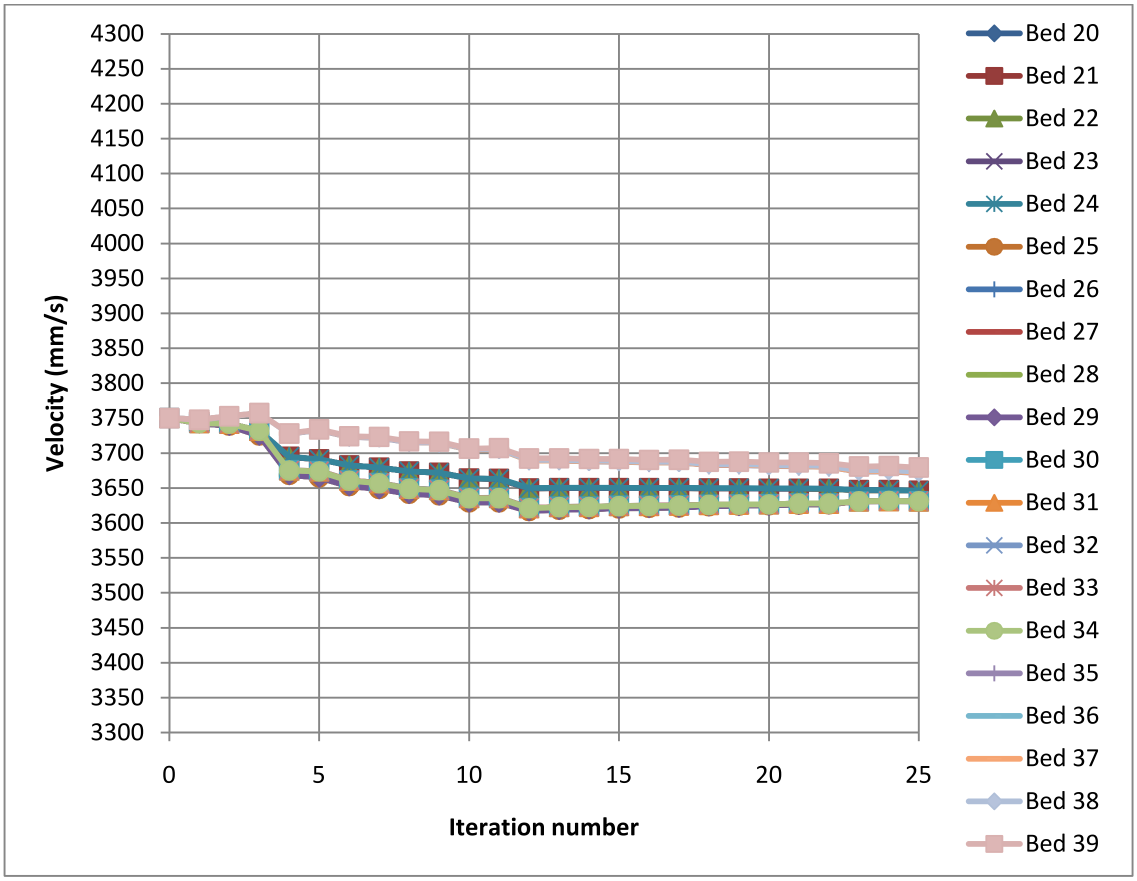


Figure (6.23): Plots representing the values of the velocity model parameters for the second layer. The twenty components that belong to the second layer are listed to the right of the graph referenced with their corresponding bed numbers.

Figure (6.24) shows the results for the second stage being the estimation of the velocity model parameters for layer 3. This figure has ten plots with different colours corresponding to ten active beds (bed 40 to bed 49) within layer 3. After 12 iterations all the active beds' parameters belong to this layer converge satisfactorily to the true velocity of the third layer. The last curve, representing bed 49, diverges to a different value. This behaviour is probably due to the fact that the velocity difference (300m/s) between bed 49 and bed 50 is comparatively large.

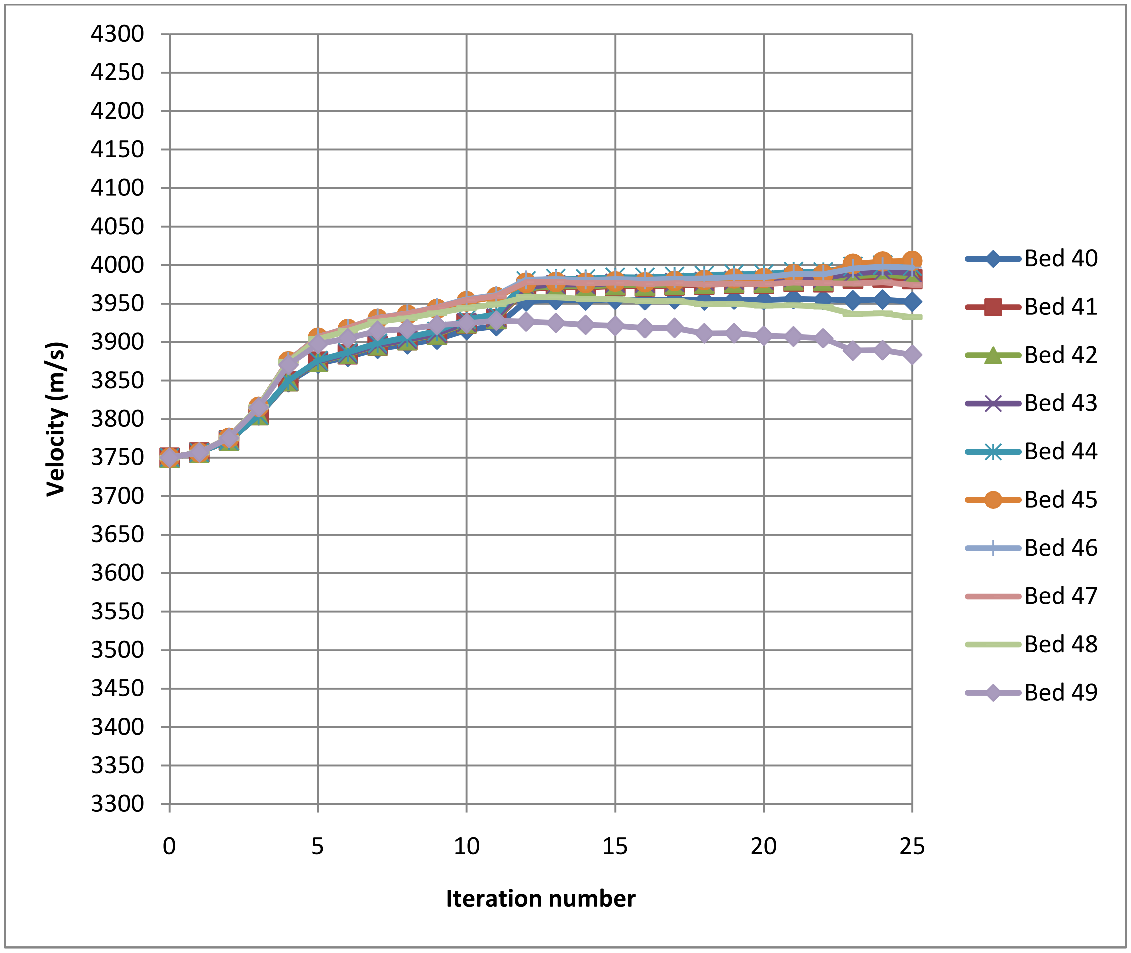


Figure (6.24): Plots representing the values of the velocity model parameters for the third layer. The ten components that belong to the second layer are listed to the right of the graph referenced with their corresponding bed numbers.

Figure (6.25) shows the results for the second stage being the estimation of the velocity model parameters for layer 4. This figure has fifteen plots with different colours corresponding to fifteen active beds (bed 50 to bed 64) within layer 4. After 13 iterations all the active beds' parameters belong to this layer converge satisfactorily to the true velocity of the fourth layer as the velocity difference (100m/s) between layer 4 and layer 5 is relatively small.

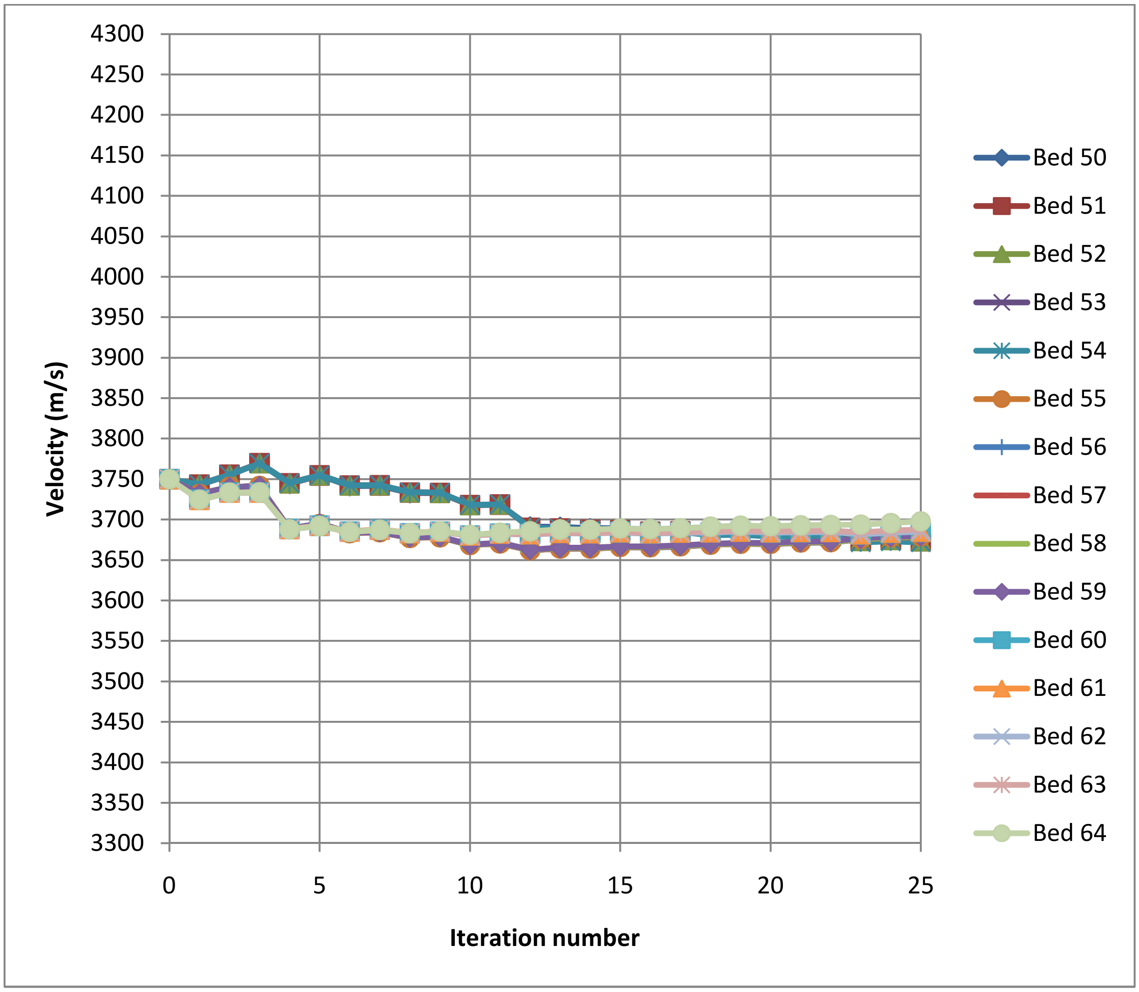


Figure (6.25): Plots representing the values of the velocity model parameters for the fourth layer. The fifteen components that belong to the second layer are listed to the right of the graph referenced with their corresponding bed numbers.

Figure (6.26) shows the results for the second stage being the estimation of the velocity model parameters for layer 5. This figure has twenty plots with different colours corresponding to twenty active beds (bed 65 to bed 84) within layer 5. After 10 iterations all the active beds' parameters belong to this layer converge satisfactorily to the true velocity of the fifth layer, except for three parameters. The corresponding three curves, representing beds 80, 81 and 82, converge to a lower value. This strange behaviour is probably due to the fact that the velocity difference (400m/s) between the fifth and sixth layer is reasonably large.

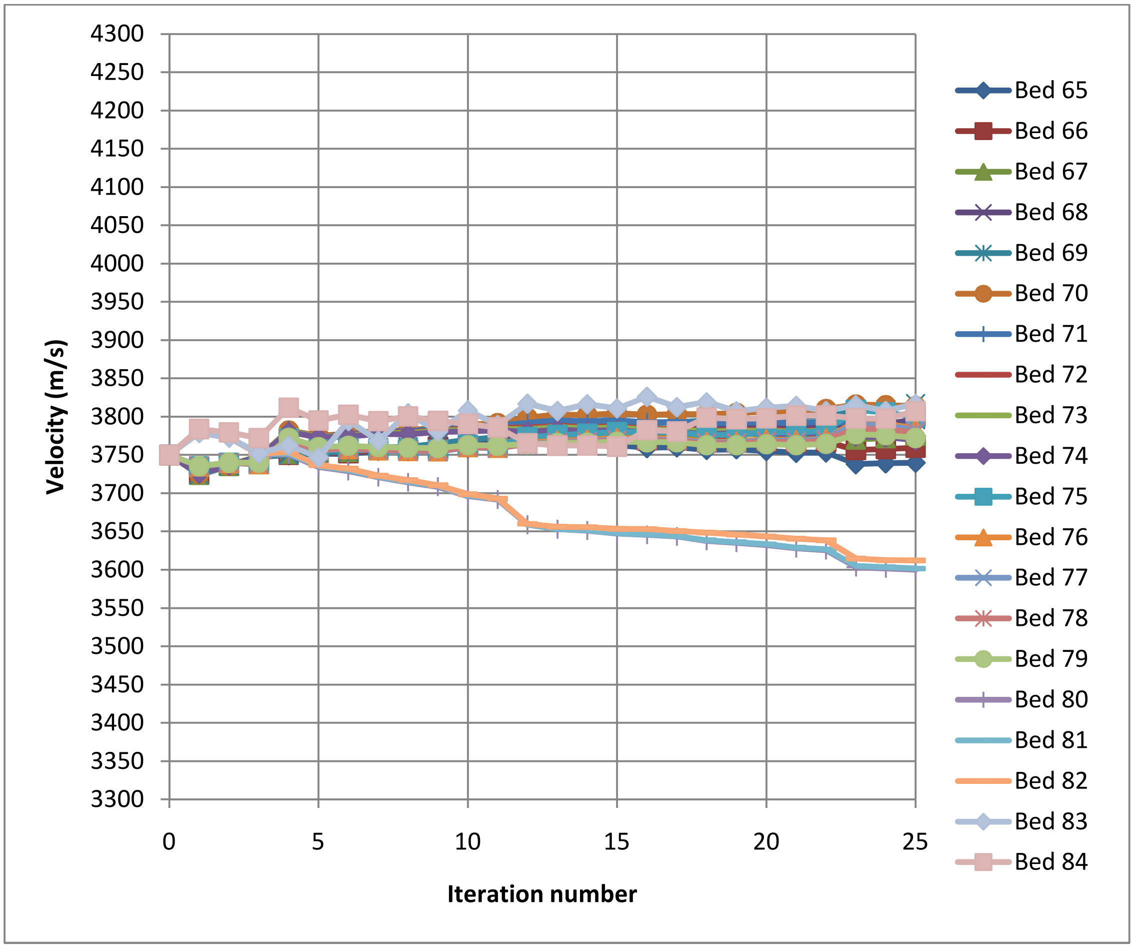


Figure (6.26): Plots representing the values of the velocity model parameters for the fifth layer. The twenty components that belong to the fifth layer are listed to the right of the graph referenced with their corresponding bed numbers.

Figure (6.27) shows the results for the second stage being the estimation of the velocity model parameters for layer 6 and layer 7. This figure has six plots with different colours corresponding to five active beds (bed 85 to bed 89) within layer 6 and one active bed (bed 90) within layer 7. After 25 iterations, the beds' parameters belonging to layer 6 converge unsatisfactorily to different values. This behaviour is due to the fact that this layer contains the microseismic events and thus the rays that pass through this layer are nearly horizontal, as explained in Chapter 5. This causes the gradient to be insensitive to the velocity within layer 6. The last curve smoothly converges to the true velocity parameter of layer 7 after 5 iterations.

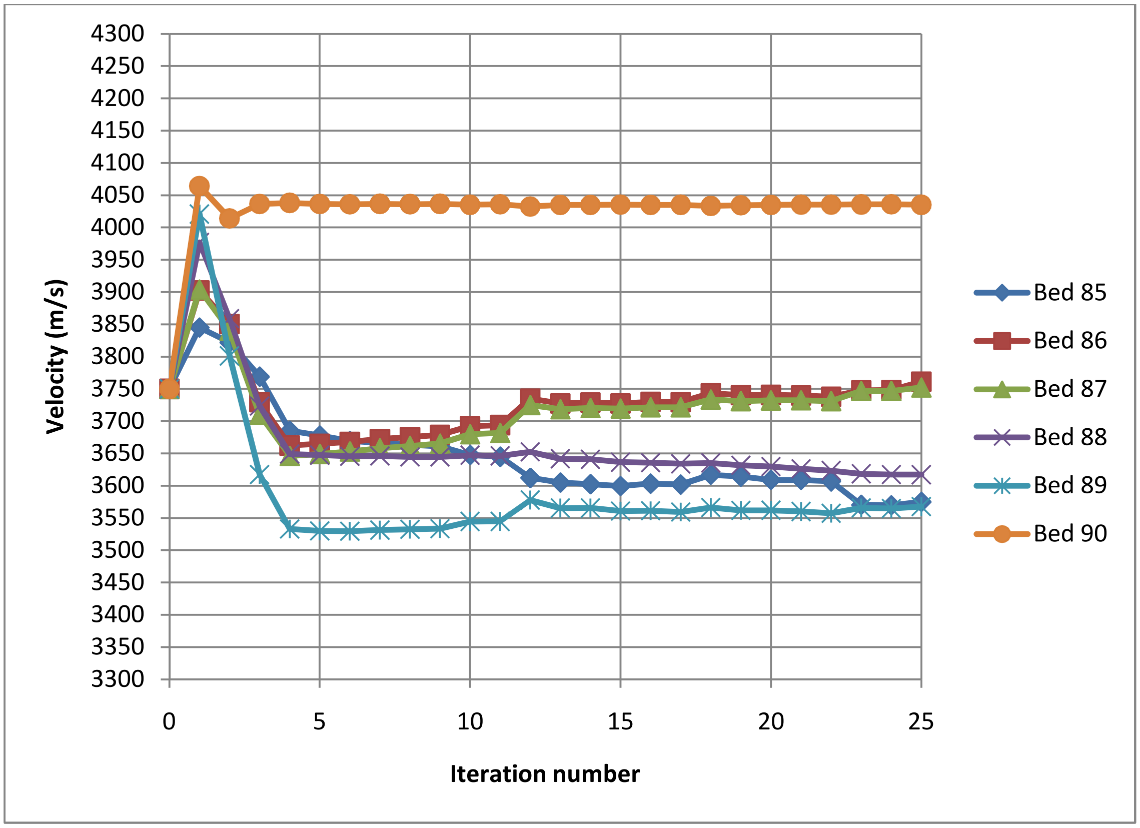


Figure (6.27): Plots representing the values of the velocity model parameters for the sixth and seventh layers. The five components that belong to the sixth layer besides the last component that belong to the seventh layer are listed to the right of the graph referenced with their corresponding bed number.

6.2.5 Parameter Estimation for Unknown Boundaries

In this section, I present the results when the formations' boundaries are unknown and we have no *a priori* velocity model. The model is divided into 101 horizontal beds such that each bed has a height of 5m representing cell height as we did in section 6.2.4. This configuration will result in 101 unknown parameters for the slowness model where we need to obtain only 71 parameters, as explained in section 6.2.4.

The estimations of the hypocentre and origin time parameters for all the eight micro-seismic events are very similar to the results obtained in the previous section despite the differences between velocity parameters for the two scenarios. Therefore, only the results related to estimating the velocity parameters are shown.

Figure (6.28) shows the results for the second stage being the estimation of the velocity model parameters for layer 2. This figure has twenty plots with different colours corresponding to twenty active beds (bed 20 to bed 39) within layer 2. After 15 iterations all the active beds' parameters belong to this layer converge satisfactorily to the true velocity of the second layer, except for the last bed's parameter. The last curve converges to a slightly different value. This behaviour is probably due to the velocity difference between bed 39 and bed 40.

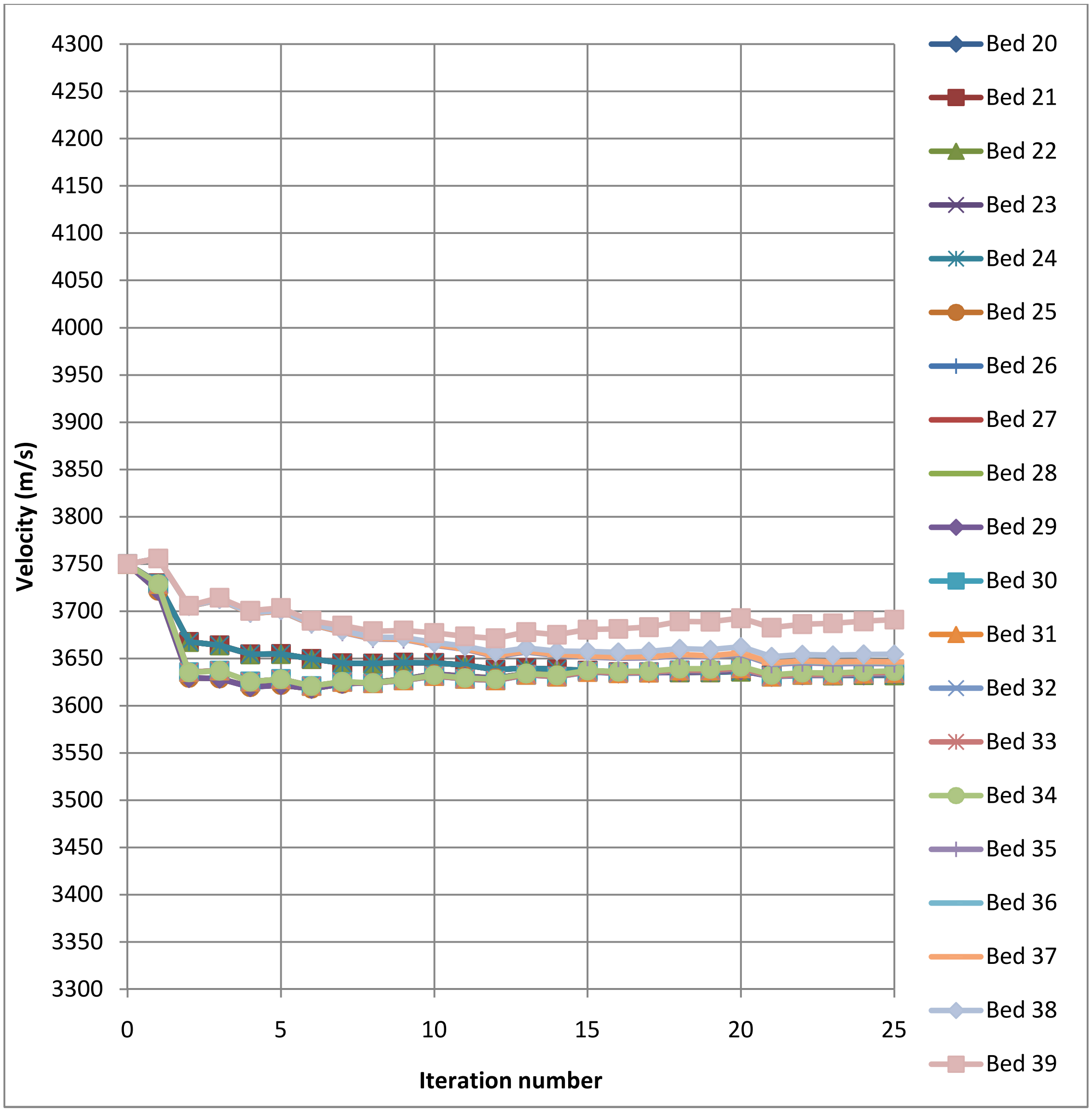


Figure (6.28): Plots representing the values of the velocity model parameters for the second layer. The twenty components that belong to the second layer are listed to the right of the graph referenced with their corresponding bed numbers.

Figure (6.29) shows the results for the second stage being the estimation of the velocity model parameters for layer 3. This figure has ten plots with different colours corresponding to ten active beds (bed 40 to bed 49) within layer 3. After 25 iterations all the active beds' parameters belong to this layer diverge away from the true velocity of the third layer.

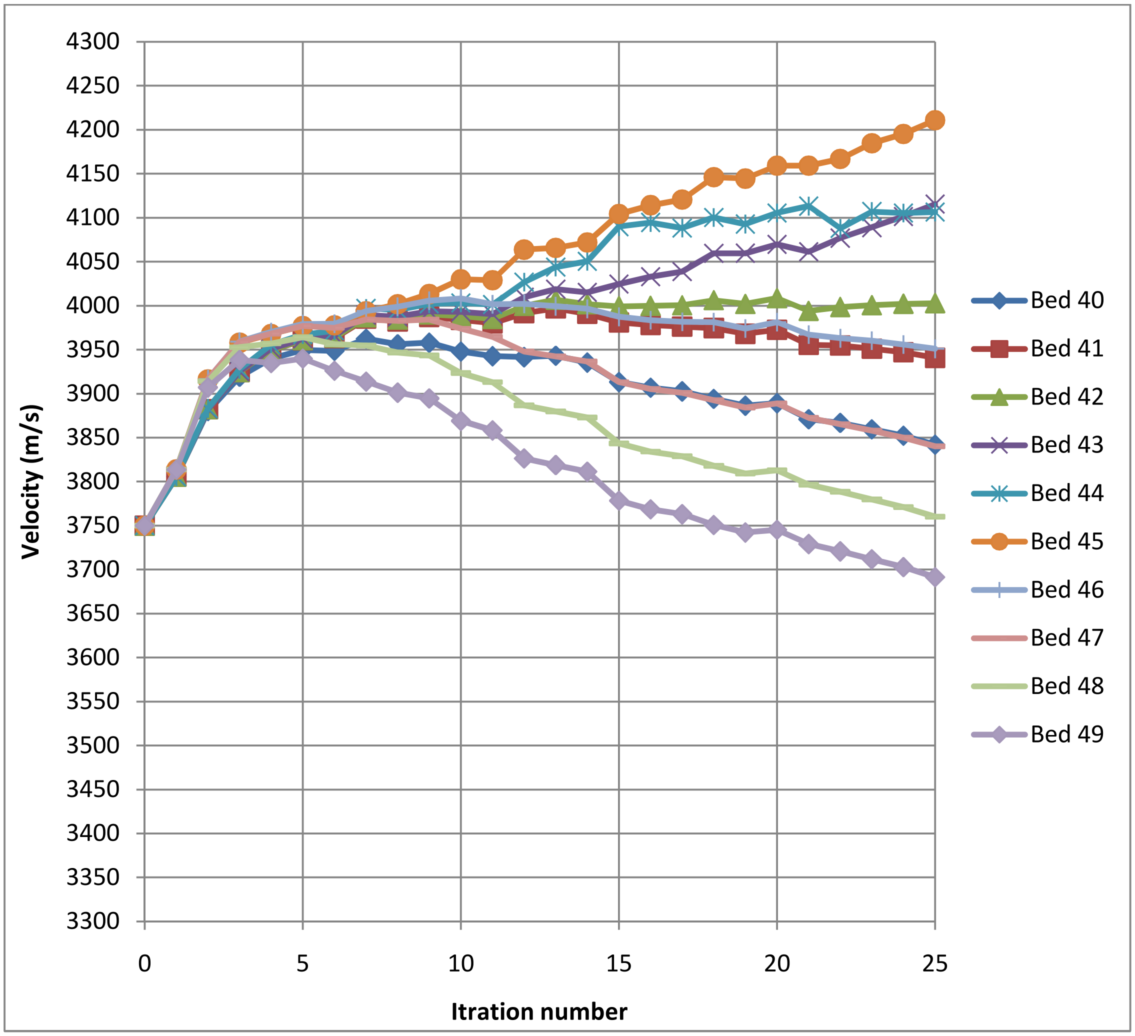


Figure (6.29): Plots representing the values of the velocity model parameters for the third layer. The ten components that belong to the second layer are listed to the right of the graph referenced with their corresponding bed numbers.

Figure (6.30) shows the results for the second stage being the estimation of the velocity model parameters for layer 4. This figure has fifteen plots with different colours corresponding to fifteen active beds (bed 50 to bed 64) within layer 4. After 13 iterations all the active beds' parameters belong to this layer converge satisfactorily to the true velocity of the fourth layer, except for the first bed's parameter. This is so because the velocity difference (300m/s) between bed 49 and bed 50 is relatively large. Still, even this parameter converges after 24 iterations.

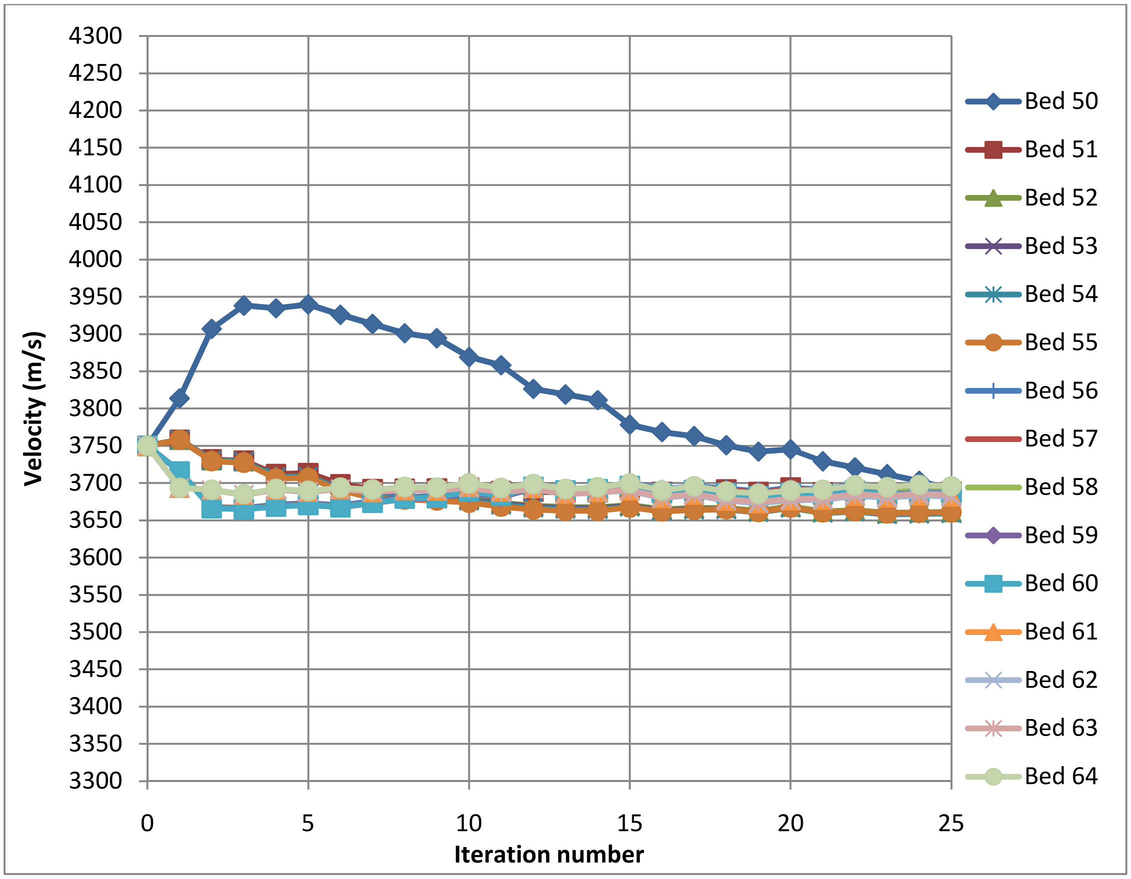


Figure (6.30): Plots representing the values of the velocity model parameters for the fourth layer. The fifteen components that belong to the second layer are listed to the right of the graph referenced with their corresponding bed numbers.

Figure (6.31) shows the results for the second stage being the estimation of the velocity model parameters for layer 5. This figure has twenty plots with different colours corresponding to twenty active beds (bed 65 to bed 84) within layer 5. After 10 iterations all the active beds' parameters belong to this layer starts to diverge away from the true velocity of the fifth layer. The three curves, representing beds 80, 81 and 82, diverges severely to a lower value. This strange behaviour is probably due to the fact that the velocity difference (400m/s) between the fifth and sixth layer is relatively large.

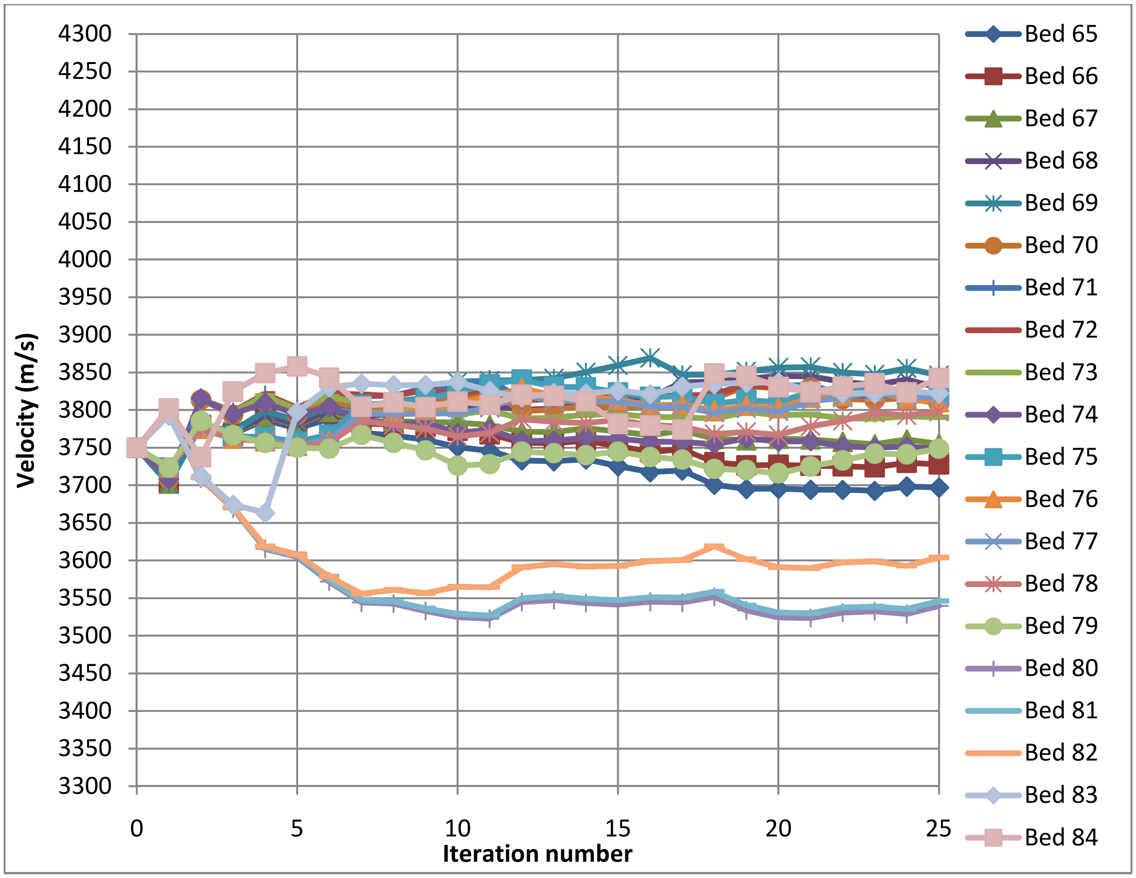


Figure (6.31): Plots representing the values of the velocity model parameters for the fifth layer. The twenty components that belong to the fifth layer are listed to the right of the graph referenced with their corresponding bed numbers.

Figure (6.32) shows the results for the second stage being the estimation of the velocity model parameters for layer 6 and layer 7. This figure has six plots with different colours corresponding to five active beds (bed 85 to bed 89) within layer 6 and one active bed (bed 90) within layer 7. Four beds' parameters belonging to layer 6 kept fluctuating dramatically. This behaviour is due to the fact that this layer contains the microseismic events and thus the rays that pass through this layer are nearly horizontal, as explained in Chapter 5. This causes the gradient to be insensitive to the velocity within layer 6. The last two curves smoothly converge to the true velocity parameter of layer 7 after 7 iterations.

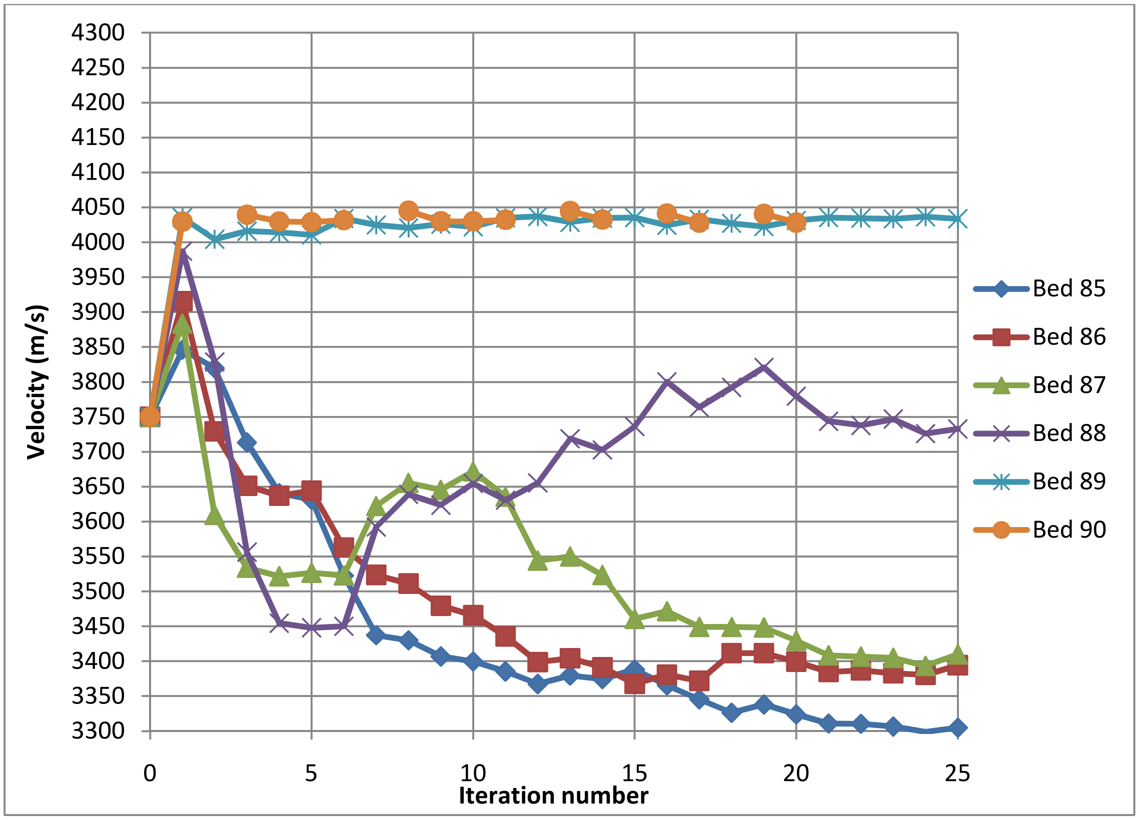


Figure (6.32): Plots representing the values of the velocity model parameters for the sixth and seventh layers. The five components that belong to the sixth layer besides the last component that belong to the seventh layer are listed to the right of the graph referenced with their corresponding bed number.

6.3 Conclusion

Generally, we can conclude that the source parameters optimisation package works satisfactorily well in estimating the position model parameters when we start with good initial velocity model. This is because the first stage is implemented to obtain the global minimiser. Further, it uses the l_1 norm which is immune against the outliers and the errors introduced by the velocity model parameters. This is in accordance with the results I obtained in Chapter 4. In regard to reconstructing the velocity model parameters, the slowness parameters optimisation package produces accurate results when the formations' boundaries are known even when the *a priori* model is not available. The Bayesian constraints are crucial when the formations' boundaries are unknown. It is, also, helpful to set the formations' boundaries according to the receivers' positions when the formations' boundaries are unknown if the case allows. The slowness parameters optimisation package is very sensitive to errors. This conclusion is based on the results obtained in Chapter 5.

I have implemented all the algorithms I have developed within the previous chapters as an iterative two-stage integrated framework on the problem stated in Chapter 1. It has worked very well in estimating the sources position parameters, but not as well in estimating the velocity model parameters due to the nonlinearity caused by the inaccuracies in micro-seismic events position parameters. The convergence to the correct solutions is insensitive to the Bayesian constrains when the formations boundaries are known. However, the Bayesian constrains have improved the convergence to the correct solutions when formations' boundaries are unknown. I have programmed the framework as a package consisting of object oriented C++ codes within Microsoft Visual Studio platform.

Chapter 7

Conclusions, Recommendations and Future Outlooks

Conclusions and discussion

This thesis deals with the use of arrival times from micro-seismic events, induced within a hydrocarbon reservoir, to monitor, characterise and/or image the reservoir. Such micro-seismic events are caused by the production activity, such as extracting oil or injecting fluid, causes perturbation to the local stresses and the pore pressure. The induced micro-seismic events produce high frequency waves, but they are too small in magnitude to be detected on the surface due to the seismic wave attenuation through the overburden rocks. Therefore, it is crucial to deploy many sensors positioned within the reservoir. This has lead to the development of iterative two-stage integrated framework.

The forward modelling is implemented through the use of an efficient and robust eikonal equation solver to generate the traveltime function at all nodes. This is sufficient because only the first arrival times are used. The accuracy of the eikonal solver depends inversely on the grid size with a corresponding increase to the computation time. Also, a trilinear interpolator is used to compute the traveltimes between grid points. Algebraic formulas have been derived for back tracing a ray-path through the traveltime function. The forward modelling and back tracing algorithms are essential for constructing the objective function together with its gradient, needed for the optimisation.

The first major contribution of this thesis is the development of an efficient and robust optimisation algorithm to effectively obtain the global minimiser of the objective function to attain the hypocentres and origin times. The driving tool within this optimisation algorithm is the systematic grid search method. Arrival times from micro-seismic events recorded at different receiver stations have been used to estimate the micro-seismic events position parameters and origin times such that the objective function is nonlinear and possibly multimodal. I have shown that the hypocentres and origin times of micro-seismic events are accurately estimated using arrival times and fewer receiver stations within two monitoring wells or more. The effect of the inaccuracies in the velocity model parameters on the estimation of the position parameters is minor.

The second major contribution is the development of a robust and efficient gradient optimisation algorithm to reconstruct the velocity model parameters through minimising the objective function. The driving force within this optimisation algorithm is the variable metric method employing the BFGS formula for updating the Hessian inverse which is an essential tool for ensuring the reconstructed solution. The reason for this choice is because of its superiority and robustness over the other methods in addition to its fast convergence rate. Arrival times from micro-seismic events recorded at different receiver stations have been used to estimate the velocity model parameters. The objective function is nonlinear and possibly multimodal. I have shown that the velocity model parameters are accurately estimated using arrival times and receiver stations within one monitoring well or more. The effect of the inaccuracies in the position parameters on the estimation of the slowness model parameters is very harsh. This is due to the strong nonlinearity between the events positions and traveltimes.

The gradient based algorithm uses the partial derivative of the objective function. This is obtained through the use of a five point stencil which has a local accuracy of fourth order. To reduce computation time, the gradient components of traveltimes are only computed at the corners of the cubes through which the ray has passed. The central difference, the forward and the backward difference have also been used and compared to the five point stencil which is more accurate.

Recommendations

The algorithms developed in this thesis can equally be used on seismic-while-drilling data. The drilling bit produces seismic waves propagating through the 3D medium and can be recorded at receiver stations planted on the surface close to the well. Picked arrival times are used to locate the drilling bit and/or reconstruct a 3D velocity model within the neighbourhood of the well for both P- and S-wave.

The algorithms also could be applied to Vertical seismic profile data. This is an ideal situation for generating a velocity model from the first arrival times. This situation provides the positions parameters for the sources and the receivers. Besides, the original time is known exactly. The sources (receivers) are used at many vertical levels. This is even better than the situation described in chapter 5.

Future Research

As explained in chapter 1, Saudi Aramco has recently conducted a field pilot study over a carbonate oil field. Many receiver stations were used on the surface and within three monitoring wells. The objective of such experiment is to track the flow anisotropy. The initial velocity model is very accurate. Therefore, if the micro-seismic arrival times can be picked confidently, the algorithms developed in this thesis can, as shown in chapter 4, produce reliable results. This will provide a good opportunity to challenge the robustness of such algorithms when dealing with real data.

The near surface weathering layer can cause severe damage to the quality of surface seismic leading to a deteriorated image of the subsurface. This is a prominent problem in Saudi Arabia. There have been extensive research directed to such a problem yet there has not been any effective solution. This problem can be investigated if the depth of a relatively shallow and strong reflector is known. The pre-stack time migrated gathers can be used such that the common depth points are used as sources while the surface positions for both the sources and the receivers are used as receiver stations, as described in this thesis. The traveltime is shared equally between its original source and receiver. It is important to use only near offset to ensure that the reconstructed velocity model parameters of the near surface is valid. This is a very different situation from those described in this thesis. This is because the sources and the receivers are essential positioned on two different levels. Thus, there is no vertical move out though there is a wide aperture and azimuth for 3D data.

The algorithms developed within the scope of this thesis are for waves propagating within elastic medium which can be extended to transversely isotropic medium. This can be achieved by transforming the forward modelling algorithm to account for the anisotropy. The algorithms can also be extended for finding a 3D varying velocity model through the use of 3D auto-adaptive gridding.