Summer 8-10-2016

3D Post-stack Seismic Inversion using Global Optimization Techniques: Gulf of Mexico Example

Elijah A. Adedeji
University of New Orleans, eadedeji@uno.edu

Recommended Citation

Adedeji, Elijah A., "3D Post-stack Seismic Inversion using Global Optimization Techniques: Gulf of Mexico Example" (2016). *University of New Orleans Theses and Dissertations*. 2231.
https://scholarworks.uno.edu/td/2231

3D Post-stack Seismic Inversion using Global Optimization Techniques: Gulf of Mexico Example

A Thesis

Submitted to the Graduate Faculty of the
University of New Orleans
in partial fulfillment of
the requirements for the degree of

Master of Science,
in
Applied Physics
with concentration Geophysics

By

Elijah Adedeji
B.S. Federal University of Technology, Akure 2008

August, 2016

Acknowledgment

We sincerely acknowledge WesternGeco for providing the seismic data used for this research. Special thanks also to IHS Kingdom for donating licenses of Kingdom and the newly developed seismic inversion plugin. I would like to appreciate my Co-Advisors Dr Abu Sarwar and Dr Juliette Ioup for their passion, love, patience, guidance, and support for me throughout this research. I would like to thank Drs Ashok Puri and Arslan Tashmukhambetov for their very helpful suggestions, steer and active participation in the research.

Special thanks to the 2015 AAPG Imperial Barrel Award team members and colleagues, Joshua Flathers, Trey Kramer, Joe Frank, Mark Leopold, Rachel Carter, and Rabin Haiju. It was fun working with you guys. I am also very grateful to the Physics department for the teaching assistantship, without which, I would not have been able to complete this research. I am greatly indebted to Dr George Ioup (late) for his impact on my life as a geophysics graduate students at UNO.

I want to thank all members of my family, especially my Mom (Elizabeth), and my brothers Olufemi and Tosin. You have been always my role models.

I would like to dedicate this work to my loving late father, Evangelist Lawrence O. Adedeji, who went to be with the Lord few months after I started the graduate studies.

Table of Contents

List of Figure

List of Tables

Nomenclature and Abbreviations

Mississippi Canyon	MC
Gulf of Mexico	GOM
Three Dimensional	3D
Basis Pursuit Inversion	BPI
Simulated Annealing	SA
Kirchhoff Pre-stack Time Migration	PSTM
Random Noise Attenuation	RNA

Abstract

Seismic inversion using a global optimization algorithm is a non-linear, model-driven process. It yields an optimal solution of the cost function – reflectivity/acoustic impedance, when prior information is sparse. The inversion result offers detailed interpretations of thin layers and internal stratigraphy as well as improved interpretations of lateral continuity and connectivity of sand bodies. This study compared two stable and robust global optimization techniques, Simulated Annealing (SA) inversion and Basis Pursuit Inversion (BPI), as applied to post-stack seismic data from the Gulf of Mexico.

Both methods use different routines and constraints to search for the minimum error energy function. Estimation of inversion parameters in SA is rigorous and more reliable because it depends on prior knowledge of subsurface geology. The BPI algorithm is a more robust deterministic process. It was developed as an alternative method to incorporating *a priori* information. Results for the Gulf of Mexico show that BPI gives a better stratigraphic and structural actualization due to its capacity to delineate layers thinner than the tuning thickness. The SA algorithm generates both absolute and relative impedances, which provide both qualitative and quantitative characterization of thin-bed reservoirs.

Keywords: Seismic Inversion, Reflectivity, Impedance, Thin-bed reservoirs, Global Optimization, Simulated Annealing Inversion, Basis Pursuit Inversion

1.0 INTRODUCTION

Seismic inversion has been used over the years by geoscientists as a critical tool in reducing risks associated with exploration, development, and production of oil and gas. It is used primarily for reservoir characterization and petro-physical studies because layer information at well locations is extended throughout the seismic volume, leading to the delineation of the true subsurface geology.

In seismic inversion, global optimization techniques are employed to combat the problem of solutions being trapped within local minima. This is done by estimating the smallest possible misfit between the model (objective function) and the seismic data. We state that the inversion process is model-driven because the observed seismic data can be considered as a forward model in which the seismic wavelet is convolved with the earth's reflectivity series. This process can be considered to place in either stochastic or deterministic space.

When these techniques are applied to post-stack seismic data, measurable true model parameters of the subsurface rock properties (earth models) such as impedance, compressional velocity (Vp), and reflectivity can be generated with all local minima being accounted for without any loss of information, thereby leading to an acceptable global optimum solution.

Generally, these procedures help remove the effects of the wavelet, solve the problem of tuning, and enhance quantitative characterization of sand bodies, fluids mapping, volumetric estimation, recovery of thin-bed stratigraphy, and seismic data de-noising. Inversion results from global methods can offer high resolvability and a better link between seismic data and lithology (Li, 2001; Sacchi and Ulrych, 1995).

1.1 Objectives and Motivation

In this study, we test the robustness and computational efficiency from two model-driven global optimization techniques, Simulated Annealing (SA) and Basis Pursuit Inversion (BPI) as applied to post-stack seismic data from the Gulf of Mexico.

The main objectives of the research are as follows:

1. Explore and test the robustness and computational efficiency of both algorithms.
2. Invert for reflectivity and impedance using both BPI and SA methods of inversion.
3. Use results from both inversion techniques to determine reservoir stratigraphy and reservoir connectivity.

In general, seismic inversion is a tedious and iterative process. It comes with some pitfalls that must to be overcome in order to achieve a reliable solution. These include but are not limited to: non-uniqueness, existence, stability, robustness, inversion interval, and absolute impedance. Our motivation, therefore, is to examine how the proposed techniques attempt to solve these problems and discuss the results in comparison with one another.

1.2 Study Area and Geological Setting of GOM

The study area (Figure 1) is a portion of the offshore Louisiana Mississippi canyon (MC 13-Q) survey acquired by WesternGeco in 2003. The survey contains 3D post-stack seismic data from the Viosca Knolls and Mississippi canyon area of the Northern part of offshore the Gulf of Mexico basin. The Viosca knoll is part of the northeastern Mississippi fan (upper fan) known as the channel-levee-overbank system (Bouma *et al*., 1985). Knolls (small natural hills up to 15-m high) are formed as a result of an increase in the deposition of limestones over the sandy bottom in the shallow marine areas. The area is characterized by the distribution of deep-sea coral and *Lophelia* species.

The Gulf of Mexico (GOM) is ranked the ninth largest water body in the world with area coverage of 1542985 square kilometers and maximum depth 4023 meters. GOM basin is nearly closed in by the United States, Mexico and the Island of Cuba to the North, West and Southeast, respectively (Figure 1).

GOM basin formation began in late Triassic as a result of a plate tectonic process called rifting. The interplay of high sedimentation and salt tectonics led to the establishment of the basin during late Triassic and Jurassic. Potential for hydrocarbon accumulation on the continental shelf is a consequence of sediment deposition on the outer shelf and upper slope of the northern GOM (Bormann *et al*., 2006). In general, the depositional patterns of the GOM are due to a combination of regional and local tectonics, sea level fluctuations, subsidence, and varying sediment supply (Posamentier *et al*., 2003).

The GOM experienced a large amount of subsidence during its relatively short geologic history. This results in massive accumulation of sediments, which in turn, led to huge salt diapirism in the basin.

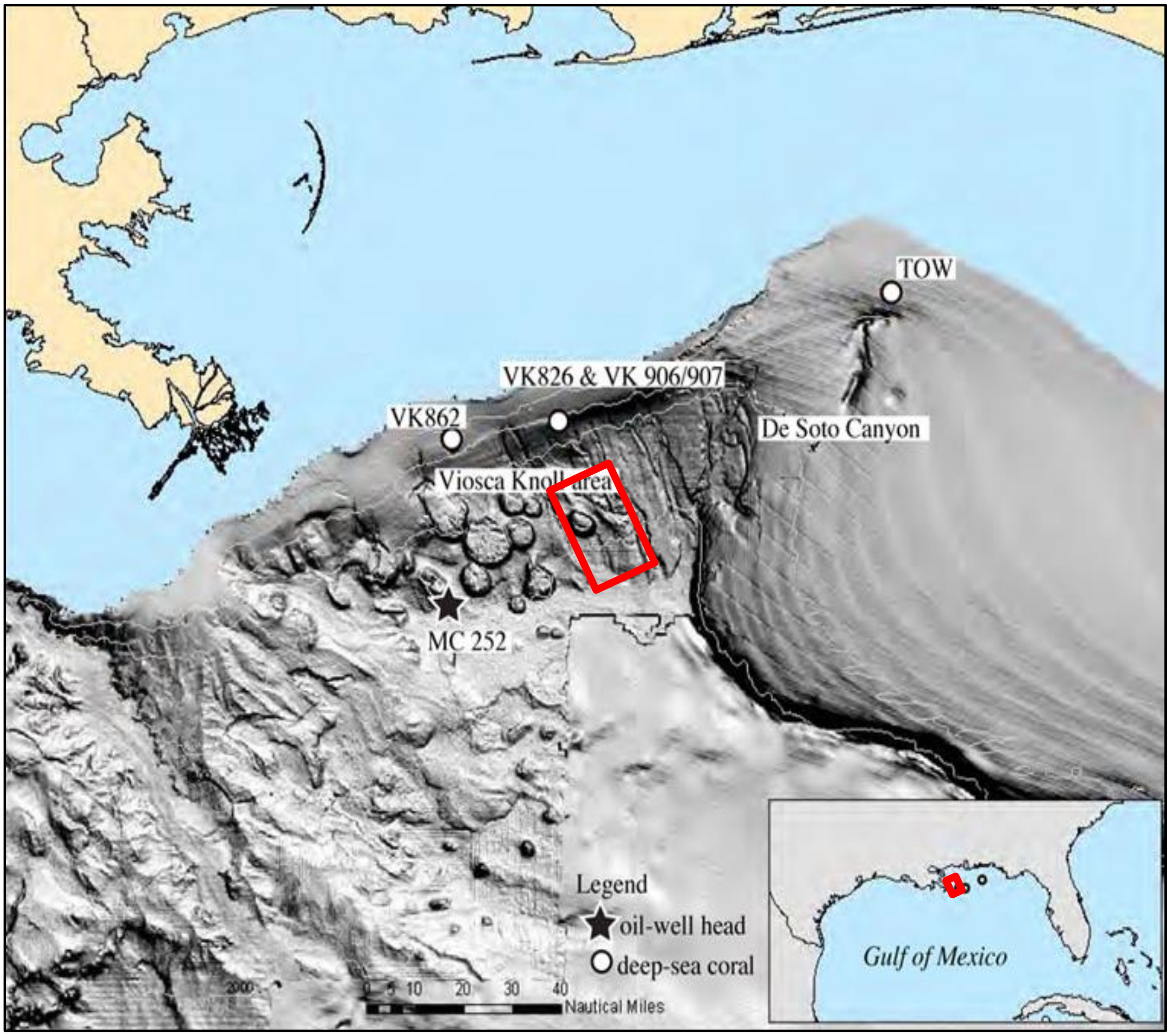


Figure 1: Map view of Viosca Knolls and Mississippi canyon area of the Northern part of offshore Gulf of Mexico basin. *Modified from http://soundwaves.usgs.gov.*

1.3 Research Dataset

Seismic data and well logs are the geophysical data used for this study. WesternGeco provided the seismic data, including stacking velocity and processing report, while well data were obtained from Stone Energy. The seismic data are 3-D, Random Noise Attenuation (RNA), and Kirchhoff pre-stack time migrated, and covers 100 square miles of the Mississippi Canyon 13-Q.

Due to the high computational demand of the seismic inversion algorithms, a sub-volume as shown in the small box (red) in Figure 2 was created from the study area. Inversion techniques used in this study were applied to the sub-volume and one 2D line extending throughout the study area. Figure 2 shows location of acquired well data in reference to the seismic dataset. Four out of seven wells were used. Data from four wells included a suite of logs: gamma ray, resistivity, P-wave, density, caliper, and sonic and neutron porosity logs. Post-stack seismic data have a record length of 10 seconds, sample rate of 2ms, and are 64 fold. Table 1 and Table 2 show detailed processing sequence, and acquisition parameters of the seismic data respectively.

Matlab® and the IHS Kingdom software (including the newly developed seismic inversion plugin) were used for data analysis, processing and interpretation.

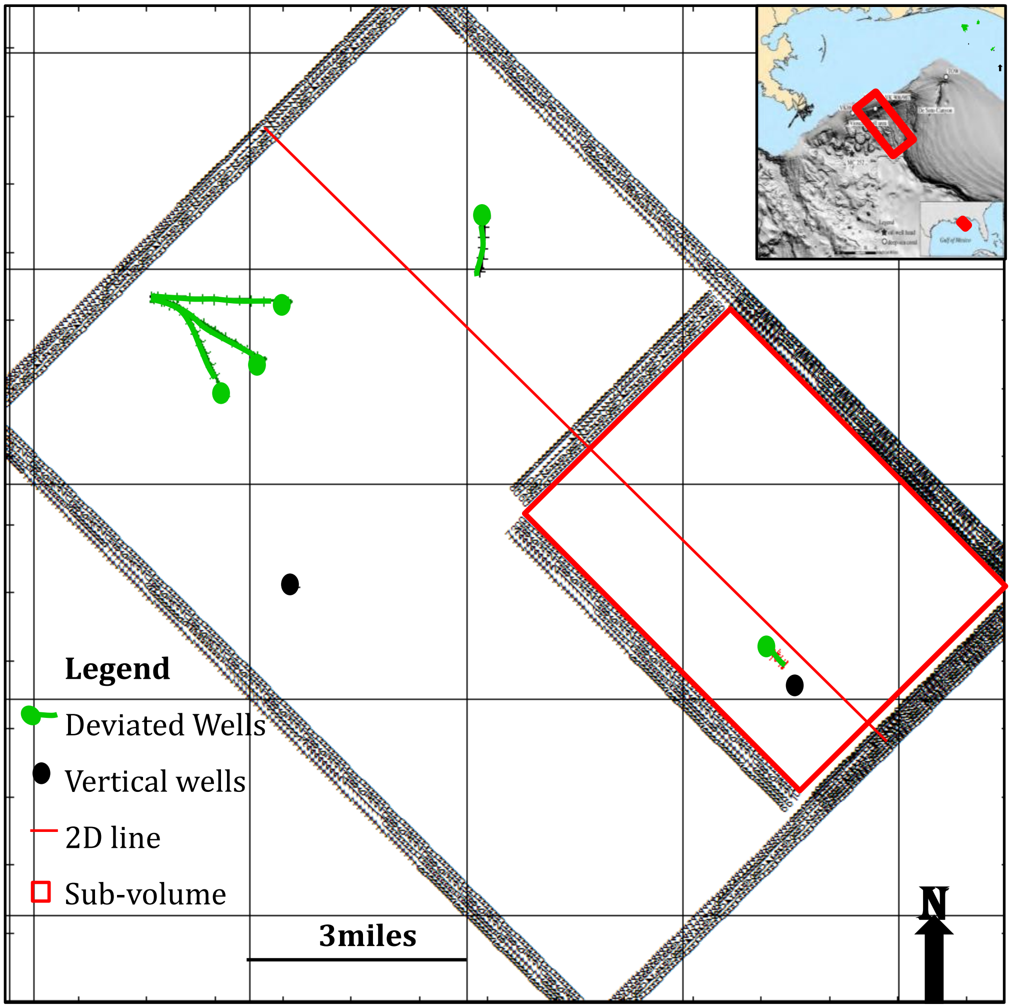


Figure 2: The extent of the acquired 3D seismic data (~100 square miles) with location of wells.

Group forming
Navigation merge
CMS designature
Receiver motion correction
Noise attenuation
IMP/SRMA demultiple
Data reduction 6.25 m to 12.5 m
Radon demultiple
Water velocity correction
Q compensation (phase only)
Interpolation In crossline to 12.5 m spacing
Kirchhoff prestack time migration
CMS whitening
Stack
Poststack processing
Final cell size: 12.5 x 12.5 m
Final fold: 64
Final record length: 10 s
Final sample rate: 2 ms
Processing completed September, 2004

Table 1: Processing Flow of Seismic Survey utilized in this study.

Recording System	Q-Marine* (Mark of Schlumberger)	Streamer Separation	100 m
Energy Source	Dual airgun arrays; 5085 In.^{3}	DGF Group Interval	12.5 m
Line Orientation	NW/SE	Sample Rate	2 ms
Source Separation	50 m	Record Length	12 s
Source Depth	5 m	Number of DGF Channels	640 per streamer
Shot Interval	31.25 m; 62.5 m per source; flip-flop	Nominal Fold	64
Streamer Depth	6 m	Final Bin Size	6.25 x 25 m
Streamer Length	8000 m	Survey Acquired	August, 2002

Table 2: Acquisition Parameters of Seismic MC 13-Q, the 3D seismic utilized in this study.

2.0 METHODS

1.2 Background for Model-Driven Seismic Inversion

Seismic data inversion is a form of inverse modelling used to develop models that describe the true subsurface geology with primarily seismic data as input. Model-driven seismic inversions start with a forward model of the subsurface earth, with parameters describing the model derived from well logs. The forward model can be written as a convolution of the seismic wavelet and reflectivity series with added noise (Figure 3a).

$$s(t) = w(t) * r(t) + n(t), \qquad (2.1)$$

where *s(t)* is the seismic trace, *w(t)* is the wavelet, *r(t)* is the reflectivity series and *n(t)* is the random noise. The rock property can be obtain from equation (2.1) by inverse modelling (Figure 3b).

The primary goal of post-stack seismic inversion is to invert *s(t)* to obtain *r(t)* in equation (2.1). Using *r(t)*, we can deduce layer properties, which give a more detailed and accurate representation of the earth than the geologic section from seismic (Figure 4a). The earth reflectivity series *r(t)* provides interface information between geologic layers. Mathematically, *r(t)* is related to acoustic impedance as follows:

$$r_i(t) = \frac{Z_{i+1} - Z_i}{Z_{i+1} + Z_i}\ , \qquad (2.2)$$

where layer i overlies layer i+1 and Z_i is the acoustic impedance in the ith layer.

The impedance of each layer can be estimated as a product of the density and the compressional velocity of sound waves traveling through the layer:

$$Z_t = v_t \, x \, \rho_t, \quad (2.3)$$

where Z_t is the acoustic impedance, v_t is the sound velocity and ρ_t is the density at a given two-way time t.

Equation 2.2 is the basis for recursive inversion (Russell, 1988). This is valid for small contrast in the impedance trend between adjacent layers such that reflectivity must be less than or equal to 0.1. By linearization, equation 2.2 can be reduced to the equation:

$$r_i(t) \simeq \frac{\Delta Z_i}{2Z_i} = 0.5\Delta \, lnZ_i(t), \qquad (2.4)$$

The above expression can now be written as:

$$Z_m(t) = 2Z_0(t) \int_0^m r(t) \, dt, \qquad (2.5)$$

Where Z_m (t) is the acoustic impedance for the earth model, $Z_0(t)$ is the acoustic impedance for the initial model (low frequency impedance model) and $r(t)$ is the reflectivity.

Figures 3 (a) and (b) show the relationship between forward modelling and inversion in seismic processing. When the effect of the wavelet is properly removed from the convolution model in Figure 4b a true subsurface geology (inversion) is realized, as seen in Figure 4a.

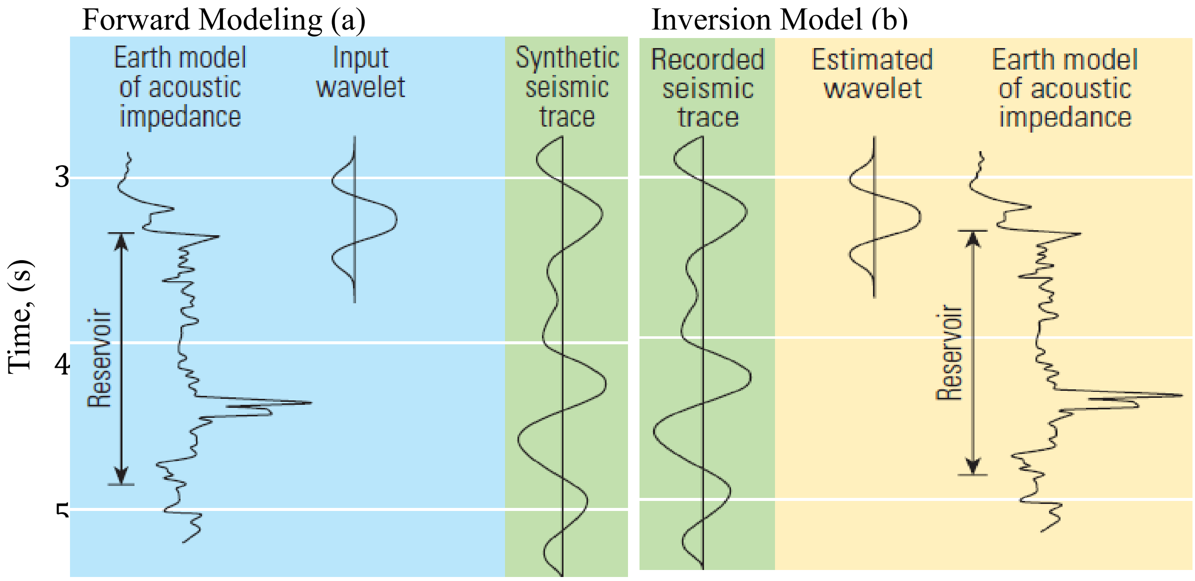


Figure 3: Relationship between forward and inverse modeling using the convolutional model. (a) Takes earth model from well logs, and convolves with wavelet to produce synthetic seismic trace. (b) Takes seismic trace and deconvolves wavelet to generate earth model. *Modified from Oil Review, 2008.*

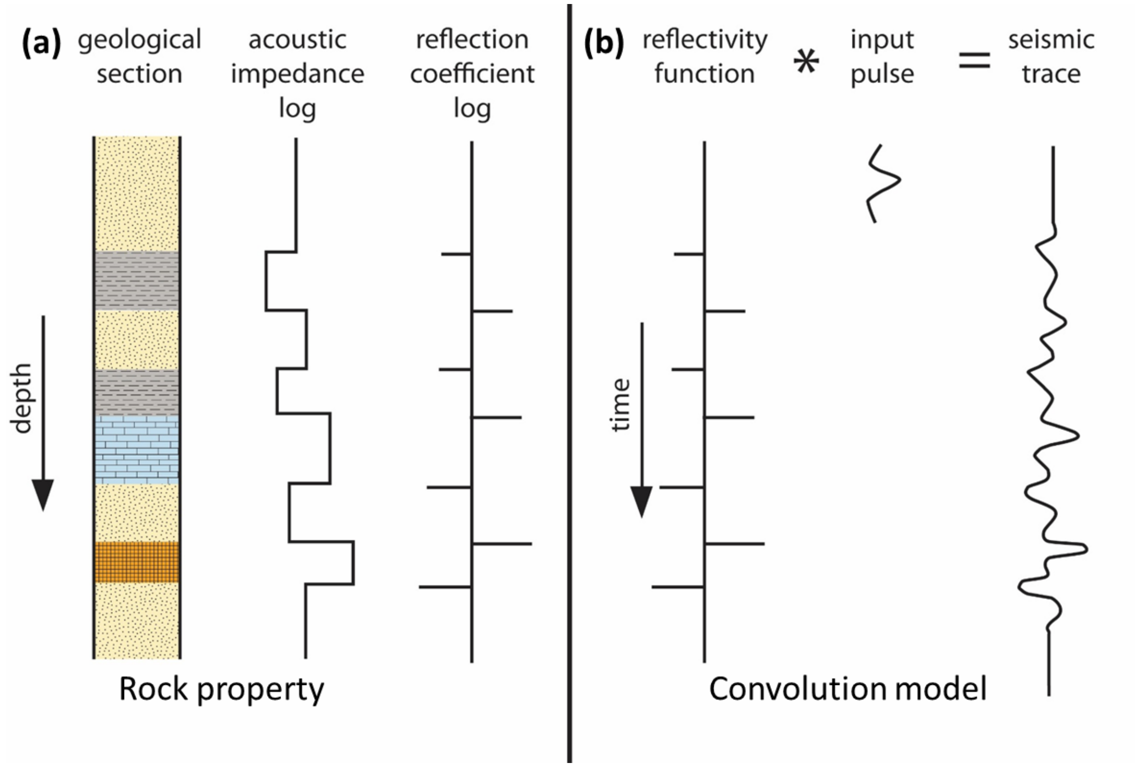


Figure 4: Relationship between rock property and convolution model. *Modified from Keary et al., 2002.*

1.3 Basis Pursuit Inversion (BPI)

Reflectivity inversion using basis pursuit was adapted from the work of Chen *et al.* (2001) on atomic signal decomposition for compressive sensing. It is a convex global optimization method of decomposing a signal into an "optimal" superposition of dictionary elements (from over-complete dictionaries), where "optimal" means having the smallest L_l norm of coefficients among all such decompositions.

Prior to this technique, there were several other methods for optimizing signals, one of which is matching pursuit. Matching pursuit (MP) algorithm decomposes a seismic trace into a series of wavelets that match their respective time-frequency signature. The process is highly iterative and depends heavily on the orthogonality of the dictionary. When the dictionary is not orthogonal, the process becomes significantly prone to errors. It was shown from the work of Chen *et al.* (2001) that basis pursuit decomposition has the potential to resolve interference between dictionary elements better and is therefore computationally efficient. Unlike matching pursuit decomposition (MPD), introduction of a sparsity norm and a regularization parameter into the cost function ensures lateral stability, even when dictionary elements are not orthogonal.

Zhang and Castagna (2011) used the same algorithm developed for compressive sensing to invert for the reflection coefficient; it is called Basis Pursuit Inversion (BPI). This was achieved by decomposing seismic traces with a non-orthogonal dictionary of seismic reflection responses, the odd and even thin layer responses. The reconstructed reflectivity series is a summation of this dipole pair. For over-complete dictionaries, a wedge model is used. This is called sparse layered inversion because the dictionary of basis functions is taken to be a wedge model of reflection coefficients pairs convolved with the extracted seismic wavelet. This algorithm was adapted to geophysical inversion in order to resolve the problem of lateral instability by spectral decomposition and inversion using trace by trace matching pursuit decomposition (Wang, 2010; Nguyen and Castagna, 2010).

When compared to conventional sparse spike inversion (SSI), it has been shown, using synthetic test that BPI can better resolve thin bed stratigraphy than SSI and also gives a better correlation to known reflectivity when an optimal regularization parameter is used for both methods (Zhang and Castagna, 2011).

In addition to its potential to de-tune and de-noise seismic data, BPI provides an alternative method for incorporating *a priori* geologic or geophysical information. The algorithm has been further extended to pre-stack seismic data using angle versus offset (Zhang, 2014). However, our goal is to apply the BPI technique to a post-stack seismic dataset and compare it with another global optimization technique known as Simulated Annealing.

The algorithm for basis pursuit decomposition has been discussed in details by *Chen et al.* (2001). A detailed description of the mathematics of the BPI as applied to post stack seismic data can be found in the work of Zhang and Castagna., 2011. It can be summarized as follows.

Equation 2.1 can also take the form:

$$d = Gm + n \quad (2.6),$$

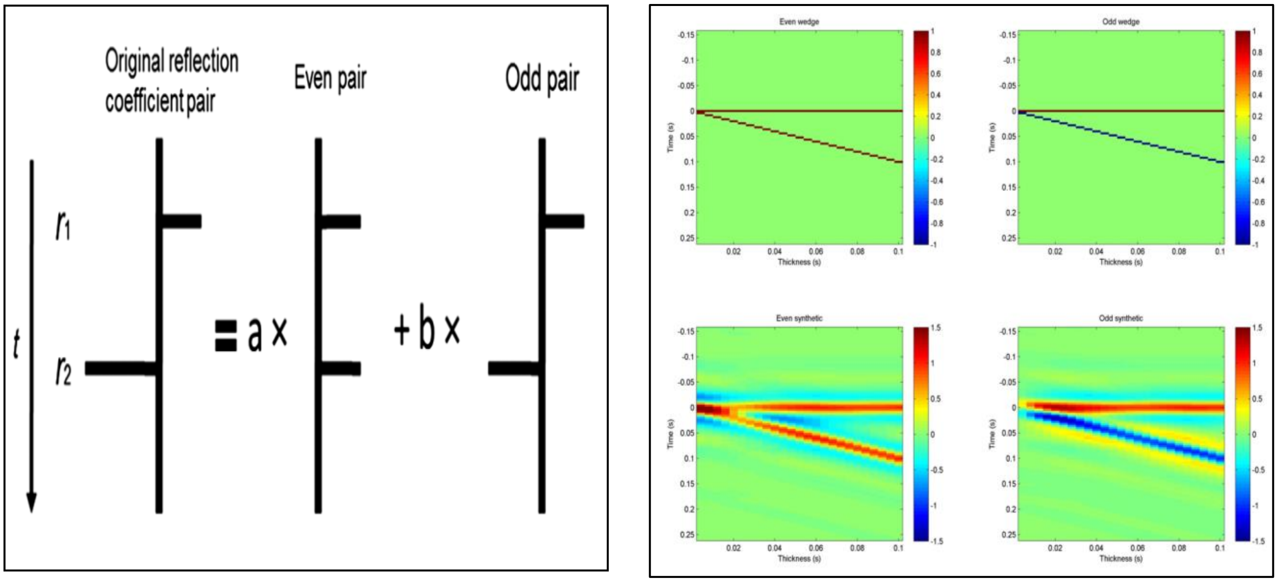
where d is the data vector (seismogram), m are the model parameters, G is the Kernel matrix and n is the noise. In order to solve for the model parameters (m) in equation 2.4, basis pursuit simultaneously minimized both the L_2 norm of the error term and the L_1 norm of the solution as shown below:

$$min\left[\left||d - Gm|\right|_2 \;+\; \lambda\left||m|\right|_1\right], \quad (2.7)$$

where λ is the regularization parameter.

Creating the wedge dictionary is very critical for BPI. A seismic layer can be defined by its top and bottom boundary, known as a reflector pair. Dipole decomposition is applied to separate a reflector pair into its constituent even and odd parts, as shown in Figure 5a. However, it is very difficult to deduce time separation between the top and bottom er in thin beds (which are normally obscured on seismic). Hence the need for a wedge dictionary.

When dipole decomposition is applied, each reflector pair is separated into odd and even parts as shown in Figure 5a. Any arbitrary pair of reflection coefficients r1 and r2 can be represented as the sum of an even and odd function. The ideal dictionary elements in Figure 5b, have a major problem of orthogonality.



(a) **(b)**

Figure 5: (a) Dipole pairs for the wedge model. (b) Synthetic (bottom) & ideal basis functions (top) of the BPI dictionary elements using a ricker wavelet of 30Hz. *Modified from Zhang and Castagna, 2008*

Parameter λ in equation (2.7) acts as a trade-off between noise and reflectivity, and between sparsity and accuracy, as seen from synthetic results carried out by Zhang and Castagna (2011). There is maximum correlation between true reflectivity and inverted reflectivity at λ=2.2 (Figure 6), from λ=0.001 to 0.1, noise is greatly amplified, and true reflectivity is completely distorted by noise. When λ is between 0.1 and 2, noise is significantly reduced, and sparser layers are possibly recovered with minimal distortion in reflectivity. There is relative correlation between true and inverted reflectivity when λ is between 2 and 2.6. BPI begins to drastically lose sparse layers as λ increase from 3 to 300.

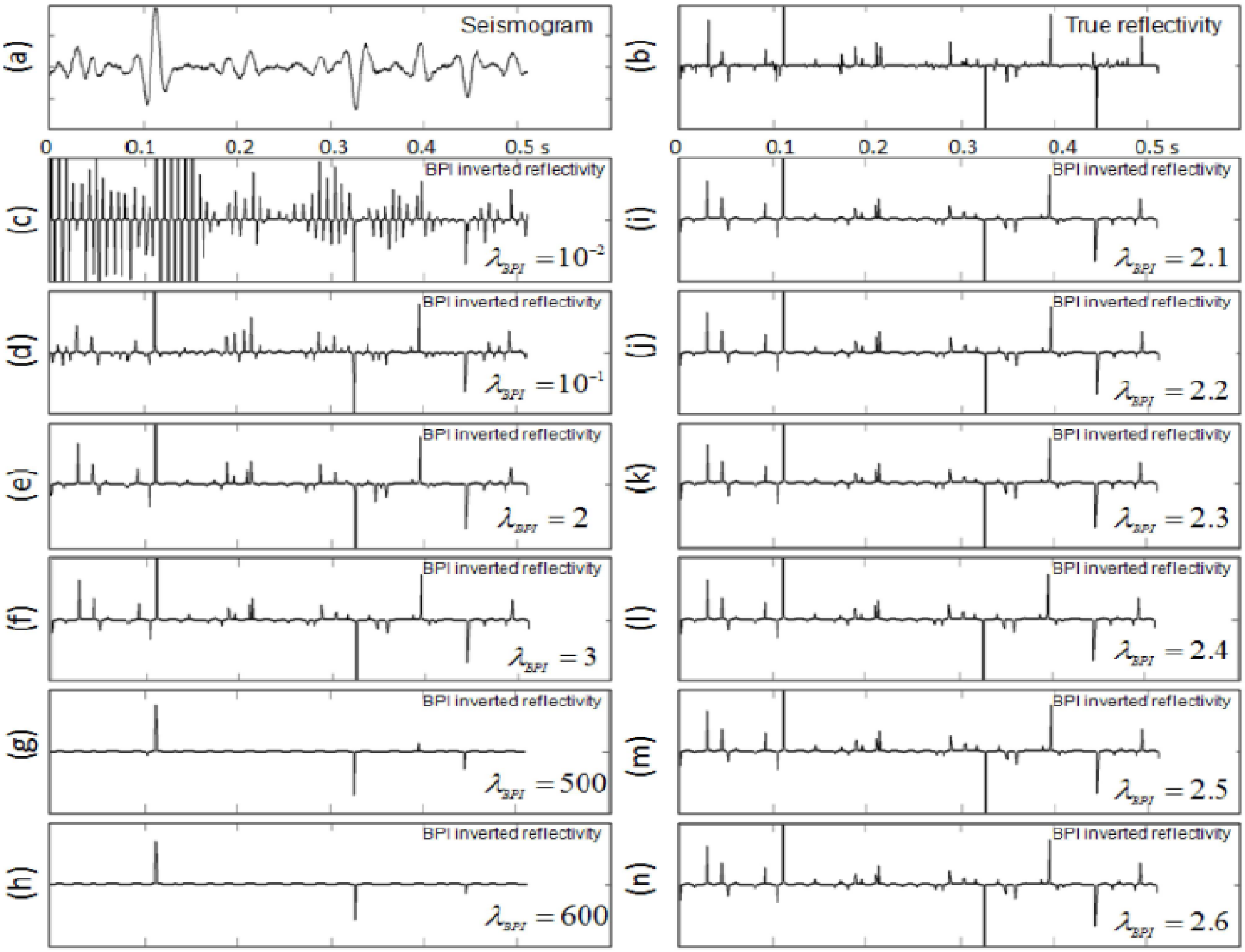


Figure 6: (a) Synthetic seismogram with 40 Hz ricker wavelet and 10% random noise; (b) true reflectivity; Consider (c) to (n) show BPI results with varying λ_{bpi} value; (c) λ_{bpi} =0.01; (d) λ_{bpi} = 0.1; (e) λ_{bpi} = 2; (f) λ_{bpi} = 3; (g) λ_{bpi} = 500; (h) λ_{bpi} = 600; (i) λ_{bpi} = 2.1; (j) λ_{bpi} = 2.2; (k) λ_{bpi} = 2.3; (l) λ_{bpi} = 2.4; (m) λ_{bpi} = 2.5; (n) λ_{bpi} = 2.6. *Zhang and Castagna, 2008*

2.3 Simulated Annealing (SA) Inversion

Simulated annealing is a global, stochastic optimization algorithm that is analogous to the process of thermal cooling, known as annealing in solid state physics. In solid state annealing, we heat the subject solid material to a very high temperature to perturb it, and then allow it to go through a cooling schedule for the atoms to slowly realign until a perfect crystal emerges. We say the newly emerged crystal is in the ground state, corresponding to the global optimum solution in our optimization problem. A metastable state known as a local optimum results if the cooling process is not slow enough or too fast for perfect realignment of the atoms.

There are many variants of the algorithm since Kirkpatrick *et al*. (1986) first applied the Metropolis algorithm (Metropolis *et al*., 1953) to solve the combinatorial optimization problem by taking a random walk through the model space (Corana *et al*., 1987; Sen Stoffa, 1991; and Goffe *et al*., 1994). This includes fast simulated annealing (FSA), very fast simulated annealing (VFSA), etc. Details can be found in Zhu and Hartley (1987), Ingber (1989) and Sen and Stoffa (1995), respectively.

Rothman (1985, 1986) was the first to use the standard SA algorithm in geophysical exploration problems by estimating residual static corrections. Mosegaard and Vestergaard (1991) used the SA algorithm to solve seismic trace inversion problems by determining the two-way travel time and reflection coefficients. Ma (2001) adopted the optimized version of standard SA by Corana *et al*. (1987) and Goffe *et al*. (1994) to invert a 1-D model of the earth's acoustic impedance.

The standard SA algorithm avoids being trapped in a local minima and finds an optimal solution for the model parameters by minimizing the error (misfit) between the model and observation. The new model is accepted unconditionally if the energy associated with the new objective function f ' is lower $(\Delta f = f' - f < 0)$. Otherwise, the new point has $\Delta f > 0$, and the new model is accepted with a probability such that $p = exp\ (-\Delta f/T)$, where T is a control parameter known as the acceptance temperature. The acceptance criterion is based on the Metropolis rule, which is based on Monte Carlo integration over configuration space (Metropolis *et al*. 1953). It is an iterative process that stops when there is very small change in the errors after many trials (Figure 7). Corana *et al*. (1987) optimized the standard SA by including a model perturbation controlled by step length (Figure 8). This improves the accuracy of the model parameters. The objective function for this SA inversion is given as:

$$\Delta f = W_1 \sum_{i=1}^{n} |S_{obs}^{i} - S_{mod}^{i}| + W_2 \sum_{i=1}^{m} |P_{pri}^{i} - P_{mod}^{i}|, \quad (2.8)$$

where S_{obs}^{i} is the observed seismic, S_{mod}^{i} is the synthetic seismic, P_{pri}^{i} is the *a priori* low-frequency impedance trend, P_{mod}^{i} is the modelled impedance, n is the number of samples in the seismic trace, m is the number of microlayers in the initial model, and W_1 and W_2 are the applied weights.

In equation (2.8), the misfit for the first term is computed using the L_1-norm mainly because it does not have the problem of overweighting a large residual when compared to the L_2-norm. The second term is *a priori* information derived from well logs and seismic velocity. It acts as a constraint that reduces the non-uniqueness problem, ensures lateral coherence of the earth models, and forces the model parameters solution to be in the neighborhood of the low frequency trend. In addition to this, the number of microlayers is set up to be more than the true earth medium. This is known as the over-parameterized scheme. It is a more efficient and practical approach to solving both interface location and impedances simultaneously during the SA inversion procedure.

This procedure has been fully integrated as a package in the IHS Kingdom software. Figure 8 shows a detailed workflow for the SA inversion. For optimal solution, a well dataset (containing sonic and density), seismic data and a low frequency model are required. The inversion is performed for a single seismic trace, after which the same inversion parameters and macro-model are applied to a seismic volume, 2D, and 3D (Figure 8).

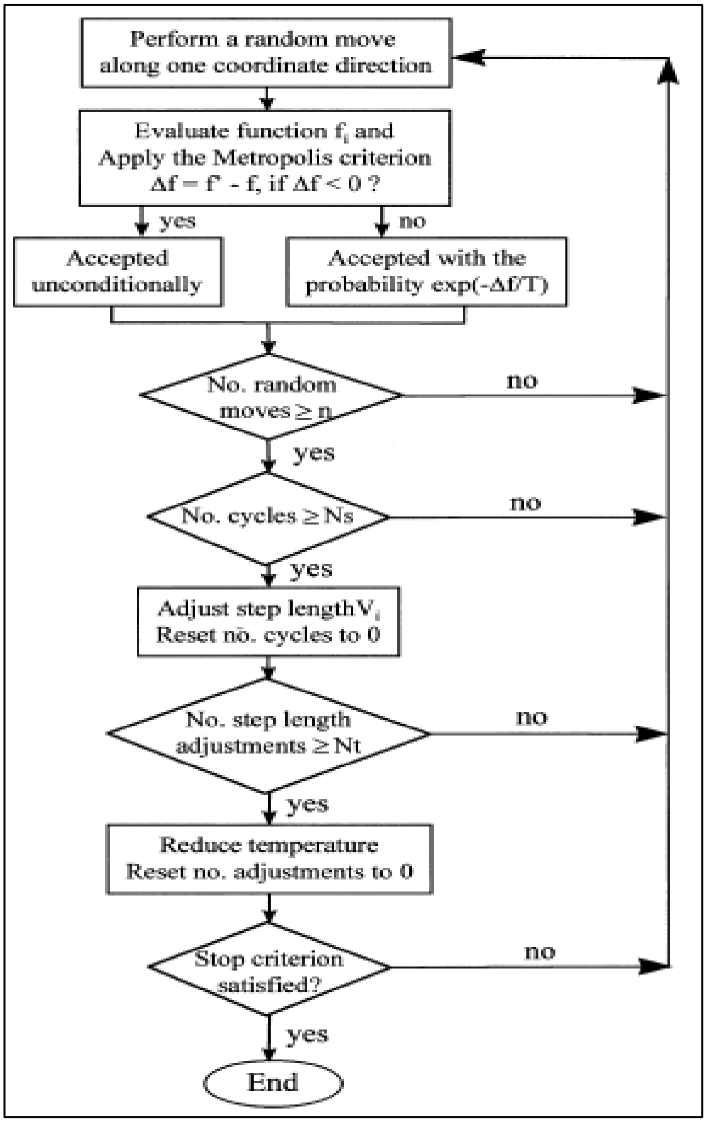

Perform a random move along one coordinate direction
Evaluate function f_i and Apply the Metropolis criterion $\Delta f = f' - f$, if $\Delta f < 0$?
yes
no
Accepted unconditionally
Accepted with the probability $\exp(-\Delta f/T)$
No. random moves ≥ n
no
yes
No. cycles ≥ Ns
no
yes
Adjust step length V_i Reset no. cycles to 0
No. step length adjustments ≥ Nt
no
yes
Reduce temperature Reset no. adjustments to 0
Stop criterion satisfied?
no
yes
End

Figure 7: A flow diagram of an improved simulated annealing optimization routine. *(Xa, 2001*)

The workflow for SA inversion in Figure 8 combines good well-to-seismic tie, estimated wavelet, macro-model, and seismic data to generate an optimal global solution of the earth model or objective function.

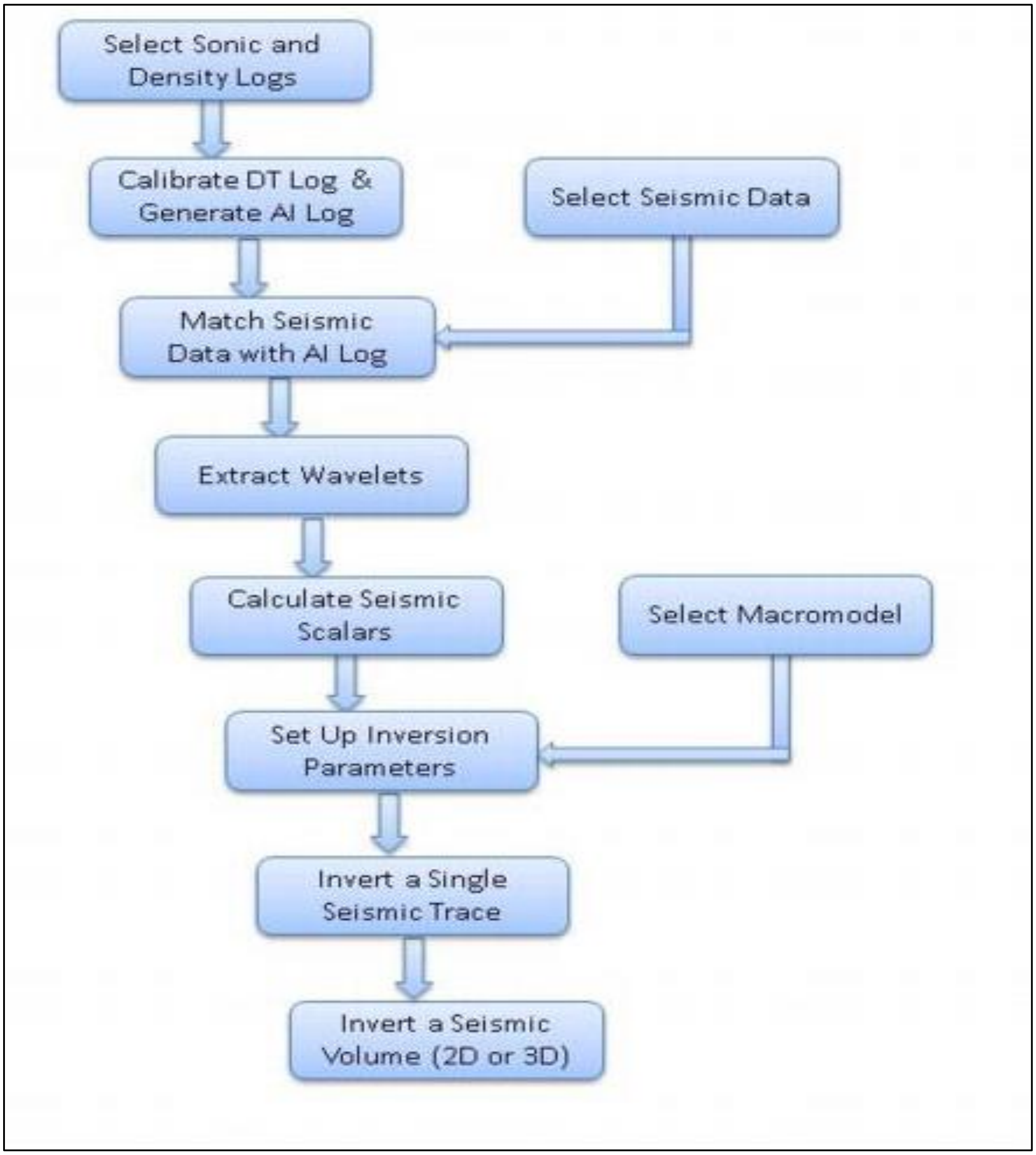

Select Sonic and Density Logs
Calibrate DT Log & Generate AI Log
Select Seismic Data
Match Seismic Data with AI Log
Extract Wavelets
Calculate Seismic Scalars
Select Macromodel
Set Up Inversion Parameters
Invert a Single Seismic Trace
Invert a Seismic Volume (2D or 3D)

Figure 8: Flow chart for seismic inversion using an optimized simulated annealing algorithm for IHS Kingdom software.

An overview of the respective deterministic and stochastic methods used for this research shows that both techniques share the first three steps of the inversion procedure as seen in Figure 9. However, in the case of basis pursuit, another inversion technique known as recursive inversion, is required in order to generate blocky acoustic impedance. This recursive inversion is described by equation (2.5).

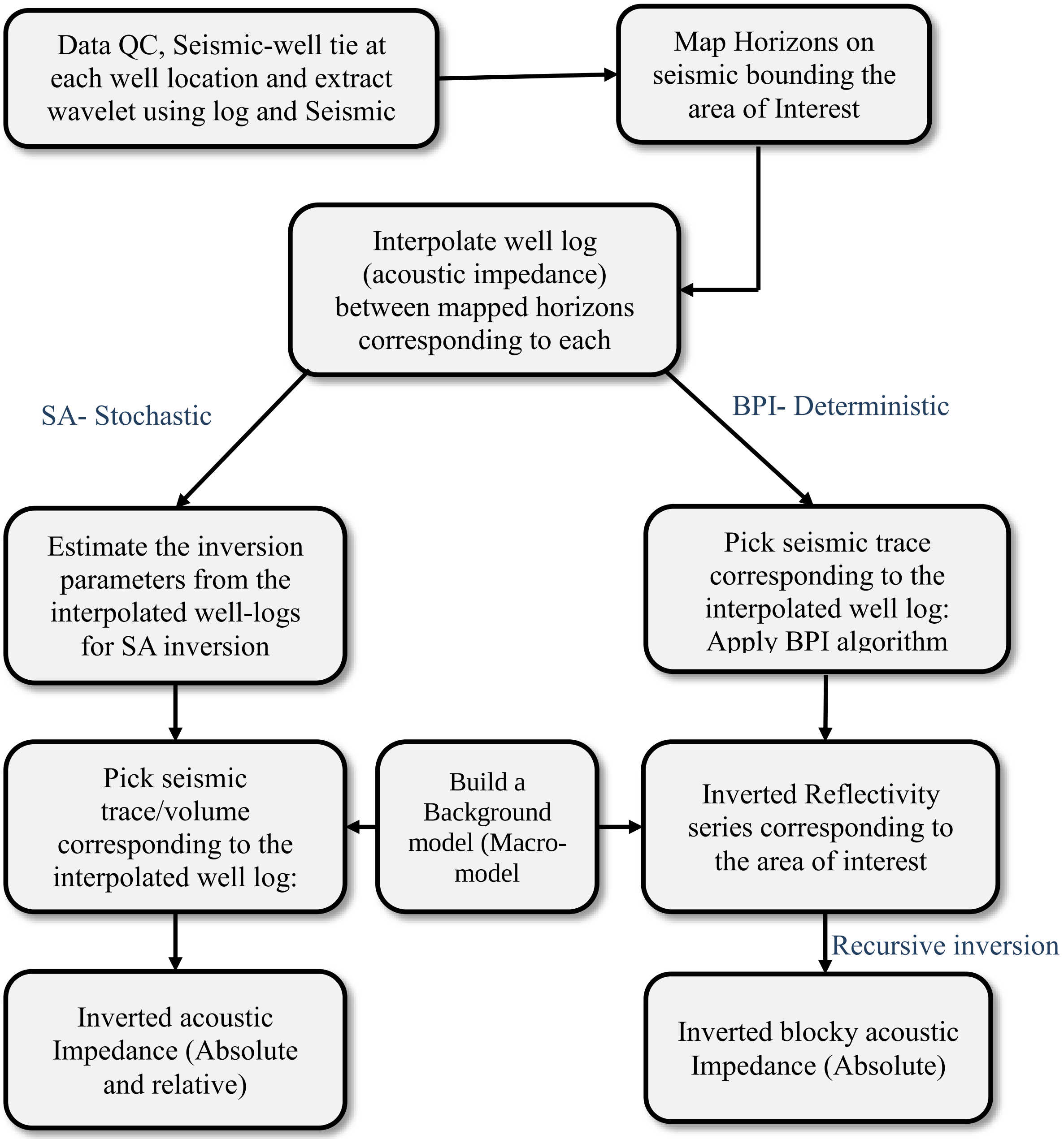

Data QC, Seismic-well tie at each well location and extract wavelet using log and Seismic
Map Horizons on seismic bounding the area of Interest
Interpolate well log (acoustic impedance) between mapped horizons corresponding to each
SA- Stochastic
BPI- Deterministic
Estimate the inversion parameters from the interpolated well-logs for SA inversion
Pick seismic trace corresponding to the interpolated well log: Apply BPI algorithm
Pick seismic trace/volume corresponding to the interpolated well log:
Build a Background model (Macro-model
Inverted Reflectivity series corresponding to the area of interest
Recursive inversion
Inverted acoustic Impedance (Absolute and relative)
Inverted blocky acoustic Impedance (Absolute)

Figure 9: Flow chat for stochastic versus deterministic inversion. *Modified from Zhang and Sen., 2002*

From Figure 9, processing steps for model-driven seismic inversion can be summarized as follows:

- Proper QC, loading and pre-conditioning of input data set, *i.e*. Seismic and well.
- Wavelet extraction based on zones of interest.
- Synthetic seismogram using extracted wavelet from Seismic with well data.
- Initiate inversion algorithm.
- Visualize and interpret the results.

2.4 Building a Macro-model for Global Methods of Seismic Inversion Techniques

The macro-model is also known as the initial model in various seismic inversion techniques. In this study, the earth model is assumed to be over-parameterized with *a priori* information and is used as the background/macro-model for two main reasons. First, it reduces the non-uniqueness problem by providing the bounds through which the algorithm searches through the model space, *i.e.* a constraint for generating the impedance volume. Second, it contains the low frequency component of the model (absolute impedance) that is missing as a consequence of seismic band-limitation.

There are several ways to build a macro-model from *a priori* information (velocity or impedance model). Here, we used stacking velocity data and combine it with horizons within the interval of interest to generate the background model (Figure 10). The Hidef tools plugin is then used to generate a smoother and more reliable background model. The low frequency component is incorporated into the background model by interpolating the sonic velocity (from well logs) with the velocity from seismic data.

BPI is a reflectivity inversion technique. In order to generate an impedance volume from this algorithm, the recursive relationship between impedance and reflectivity in equation (2.5) is used. In this case, a low frequency model is used as the macro-model, generated from interpolating well-log data and then low-pass filtered. A similar process can be applied to invert for reflectivity from the absolute impedance generated by the SA inversion algorithm.

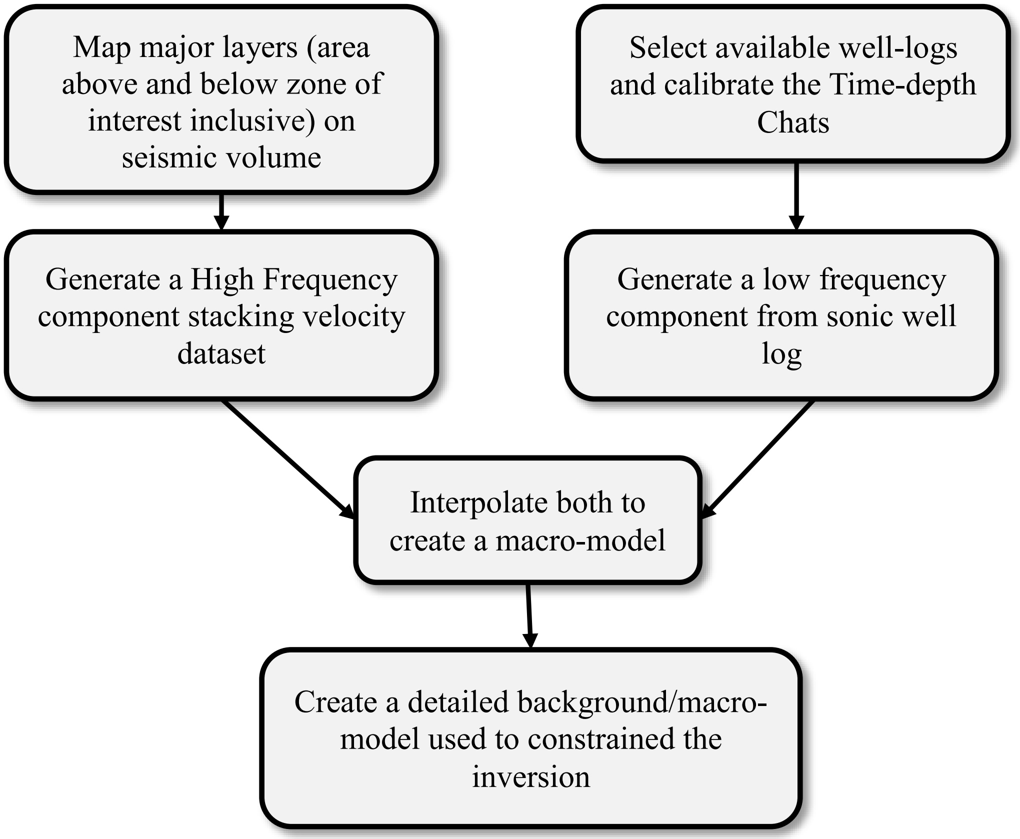

Map major layers (area above and below zone of interest inclusive) on seismic volume
Select available well-logs and calibrate the Time-depth Chats
Generate a High Frequency component stacking velocity dataset
Generate a low frequency component from sonic well log
Interpolate both to create a macro-model
Create a detailed background/macro-model used to constrained the inversion

Figure 10: Flow chart for building a macro-model for global methods of seismic inversion techniques

3.0 REAL DATA RESULTS

BPI and SA inversions are tested on a 3D post-stack dataset from the Gulf of Mexico. The first three steps of the procedure are the same (Figure 10). After all required data have been carefully loaded and quality checked, proper seismic-to-well tie is required. Figure 11 shows synthetic tie between input seismic data and well-log data. Seismic traces used for this analysis are located around a well location where an acoustic impedance log is available, and penetrate through the area of interest.

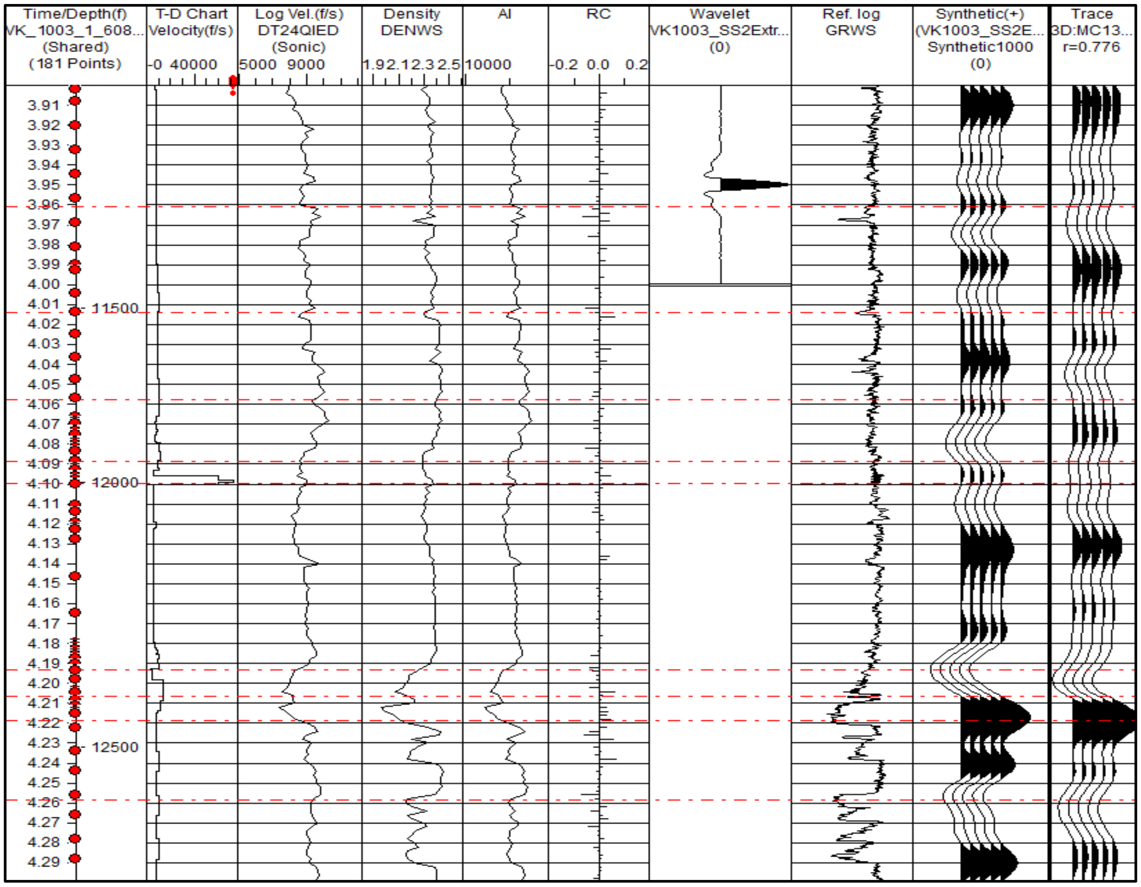
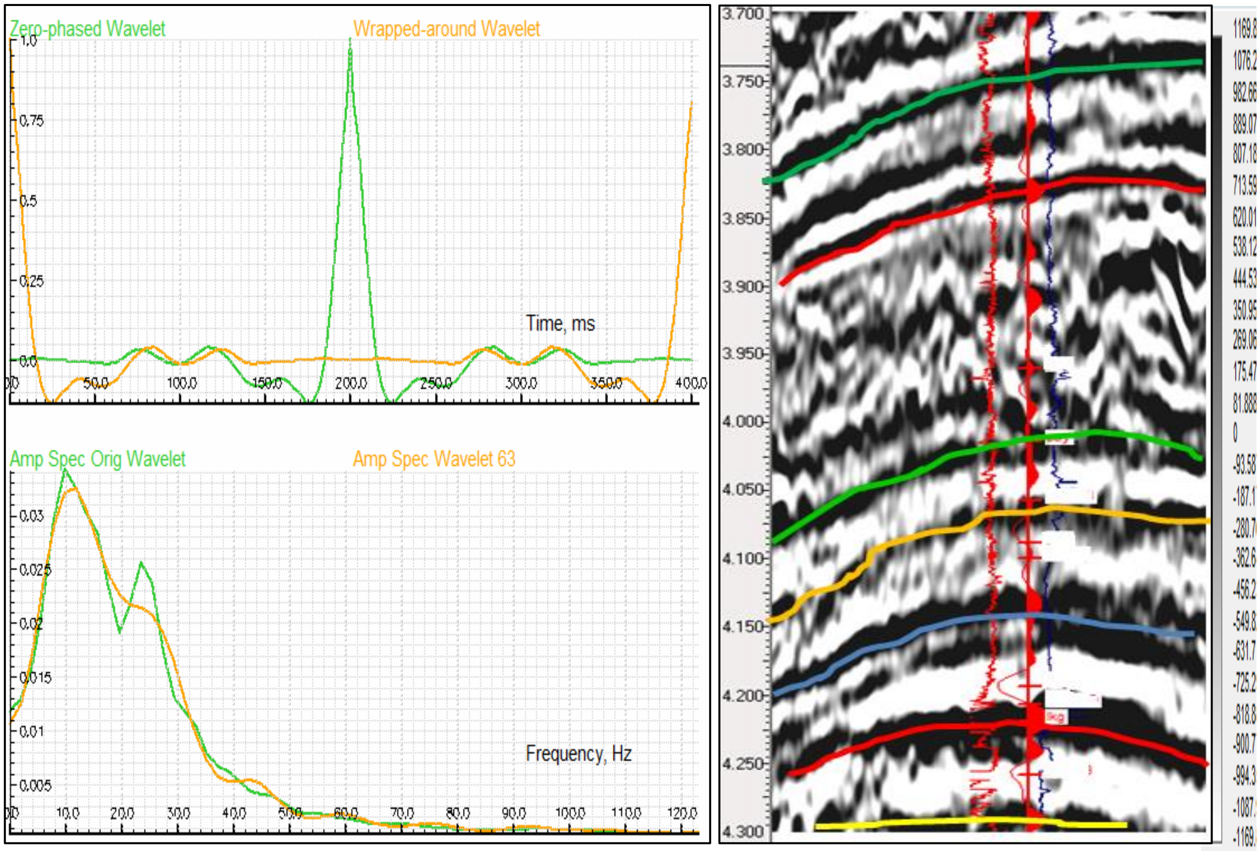


Figure 11. Seismic well tie, including gamma ray, computed impedance and reflectivity, along with the synthetic seismogram, composite seismic trace at the well location and original seismic traces near the well. Correlation coefficient R is 78%.

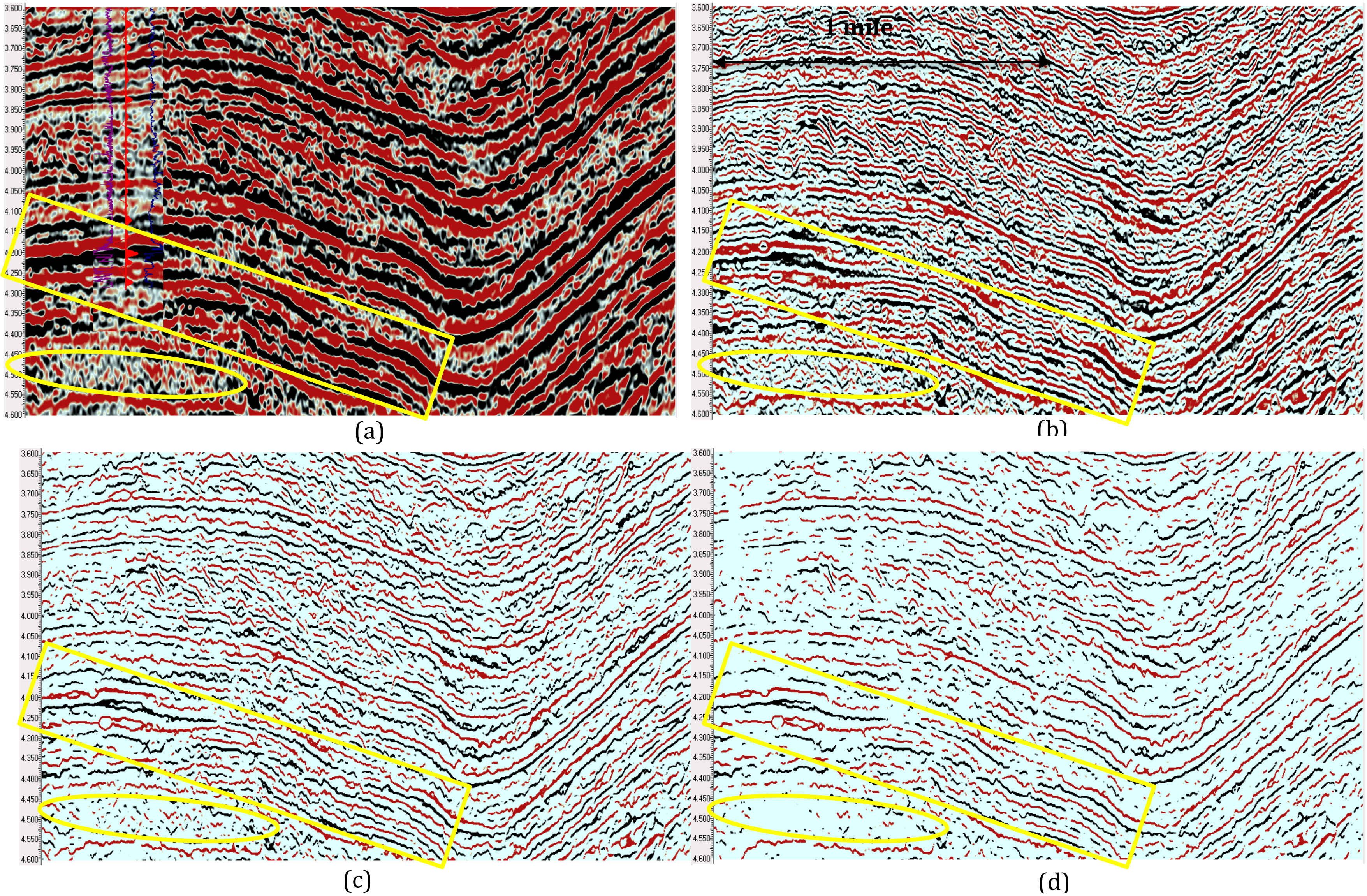


(a) (b)

Figure 12: (a) Extracted seismic wavelet at the well location using the amplitude spectrum of the seismic data averaged over 600ms. (b) Horizons in the area of interest plotted on seismic traces tied to the well. Gamma ray log on the left and resistivity on the right.

Correlation coefficient value from the well-to-seismic tie in Figure 11 is 78%, which is very good for any high-resolution inversion procedure. Extracted wavelet is zero phase, with bandwidth 10-35Hz and dominant frequency 15Hz. Seven horizons over 600ms interval on the seismic traces were mapped as the region of interest (Figure12).

BPI method relies heavily on the correct estimation of the wavelet and also depends on proper selection of the regularization parameter value that shows the best correlation to both seismic data and well logs.



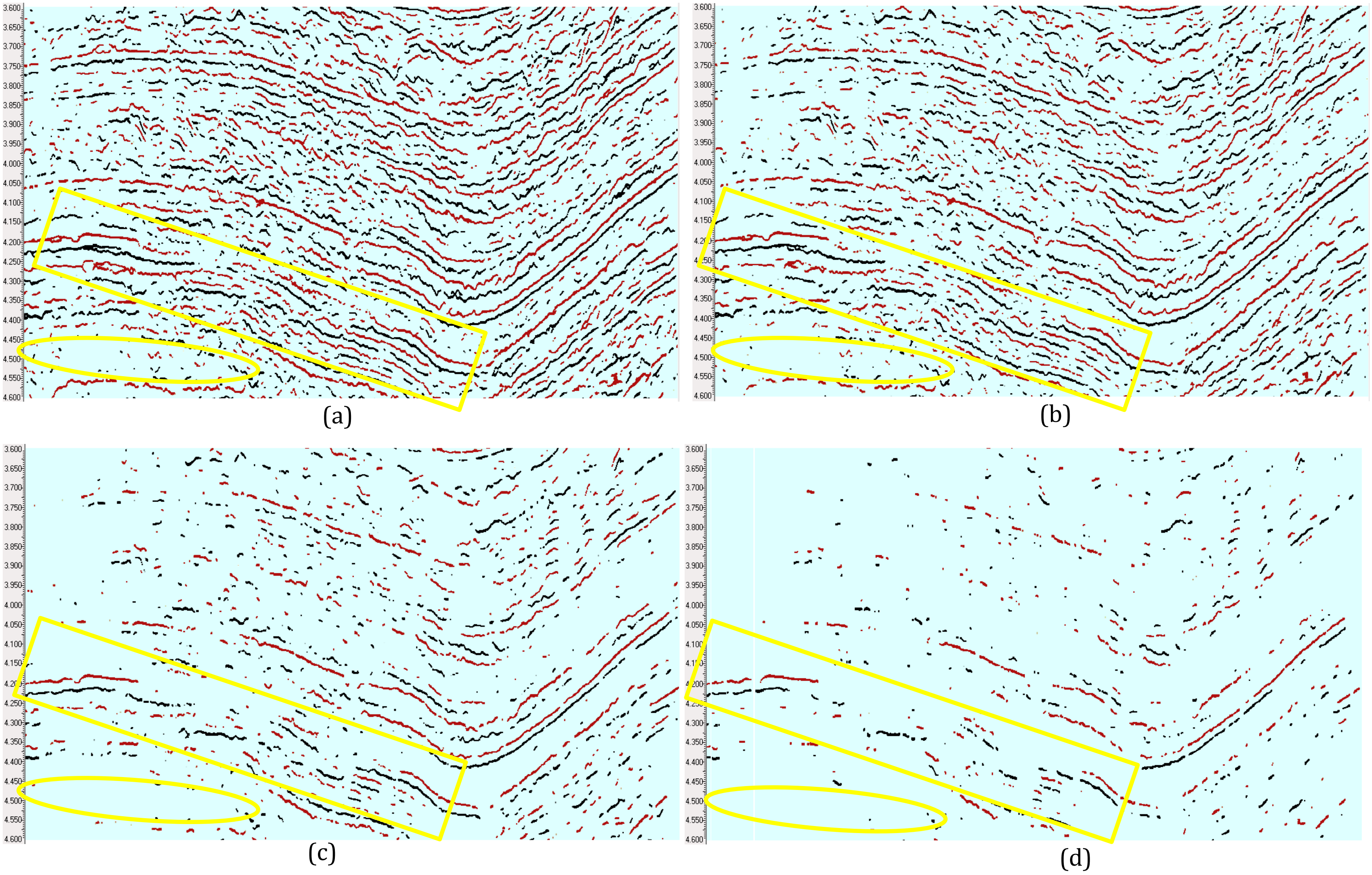
(a)

(b)

(c)

(d)

Figure 13: (a) Original Seismic. Consider (b) to (d), BPI inverted reflectivity results with varying λ_{bpi} value for a single 2D line. (b) λbpi =0.1 (c) λ_{bpi} =1; (d) λ_{bpi} = 2. Areas highlighted in yellow show the recovery of sparse layers with appropriate λ value



(a)

(b)

(c)

(d)

Figure 14: Consider (a) to (d). BPI inverted reflectivity results with varying λ_{bpi} value for a single 2D line. (a) $\lambda_{bpi} = 2.4$; (b) $\lambda_{bpi} = 3$; (c) $\lambda_{bpi} = 5$; (d) $\lambda_{bpi} = 7$. Areas highlighted in yellow shows the loss of sparse layers as λ value increases beyond 2.4

At λ= 0.1 in Figure 13b, inverted reflectivity is, but obscured by noise. There is a reduction in noise for λ =1 to 2. There is maximum correlation with the original seismic at λ=2.4 (Figure14a). Inverted reflectivity begin to dwindle in sparsity for λ= 3 to 7 (Figure14b-d). The computing time decreases also with an increase in the λ value. Figure 13d shows the potential of BPI technique to reveal layers that are below tuning.

For SA inversion, a stochastic process unlike the BPI method, estimation of the inversion parameter is very iterative and rigorous. The inversion was run for a 2D line covering the actual 3D survey, and the volume inversion was for a small portion of the survey.

The macro-model is created before the estimation of the SA inversion parameters. Figure 15 is an inline cross-section for velocity model (macro-model) generated from stacking velocity data. This model is not smooth and low frequency component from sonic log is missing. The Hidef tool is used to generate a high definition (detailed) macro-model by interpolating low-frequency component with stacking velocity from seismic data. The final output (Figure 16) is smoothed and contains more detailed information. Figure 17 shows the difference between the Hidef macro-model and the stacking velocity from seismic data. The detailed macro-model is required for generating absolute impedance from the SA inversion. Figure 18 shows the plot of velocity both seismic and sonic log respectively. For the volume inversion in this study, there is no apparent difference between the macro-model and the impedance model from well-logs; however, the former was used for more reliable results.

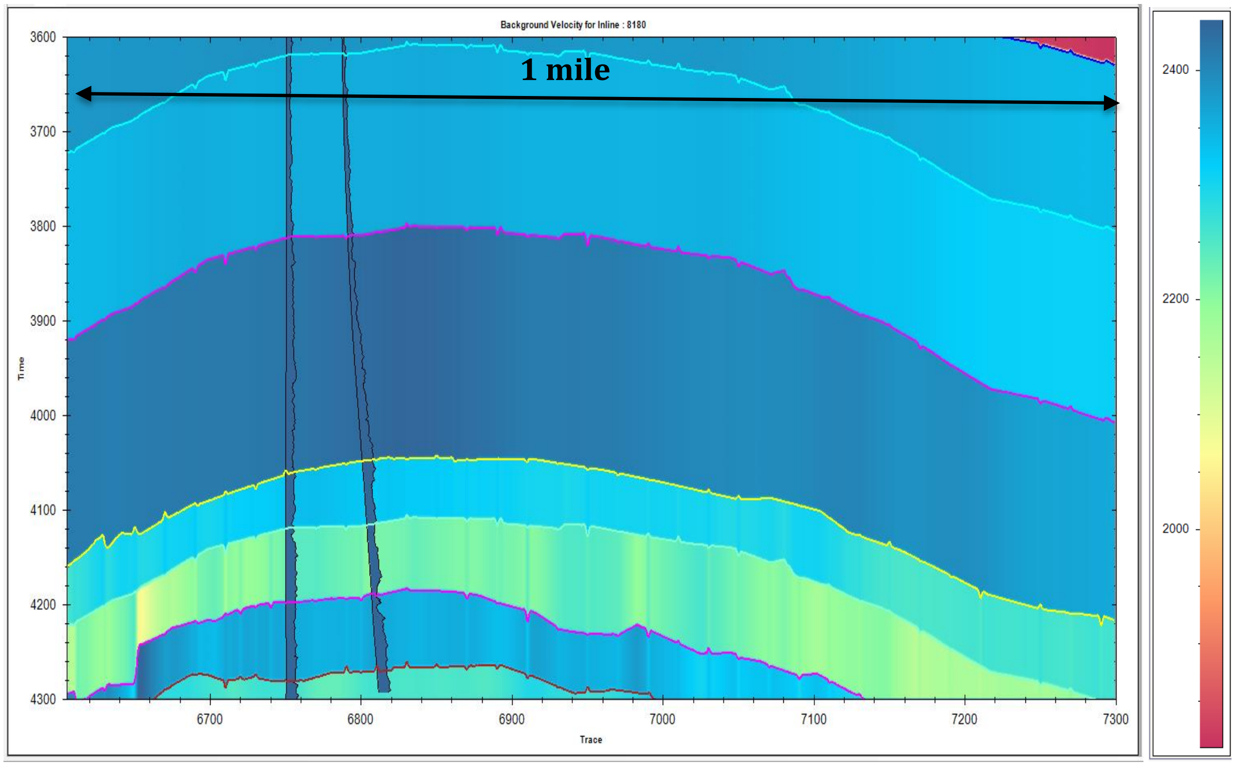


Figure 15: A velocity model (Macro-model) from the seismic stacking velocity dataset.

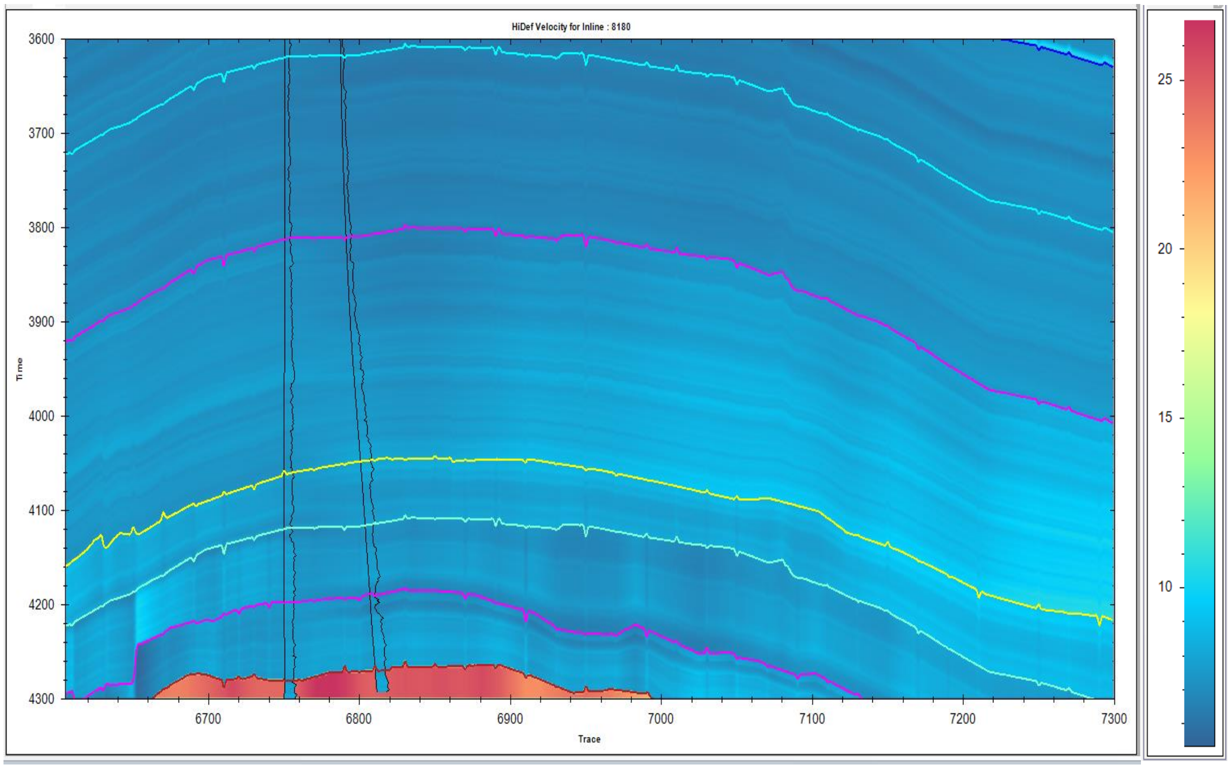


Figure 16: Detailed macro-model containing the missing low-frequency component

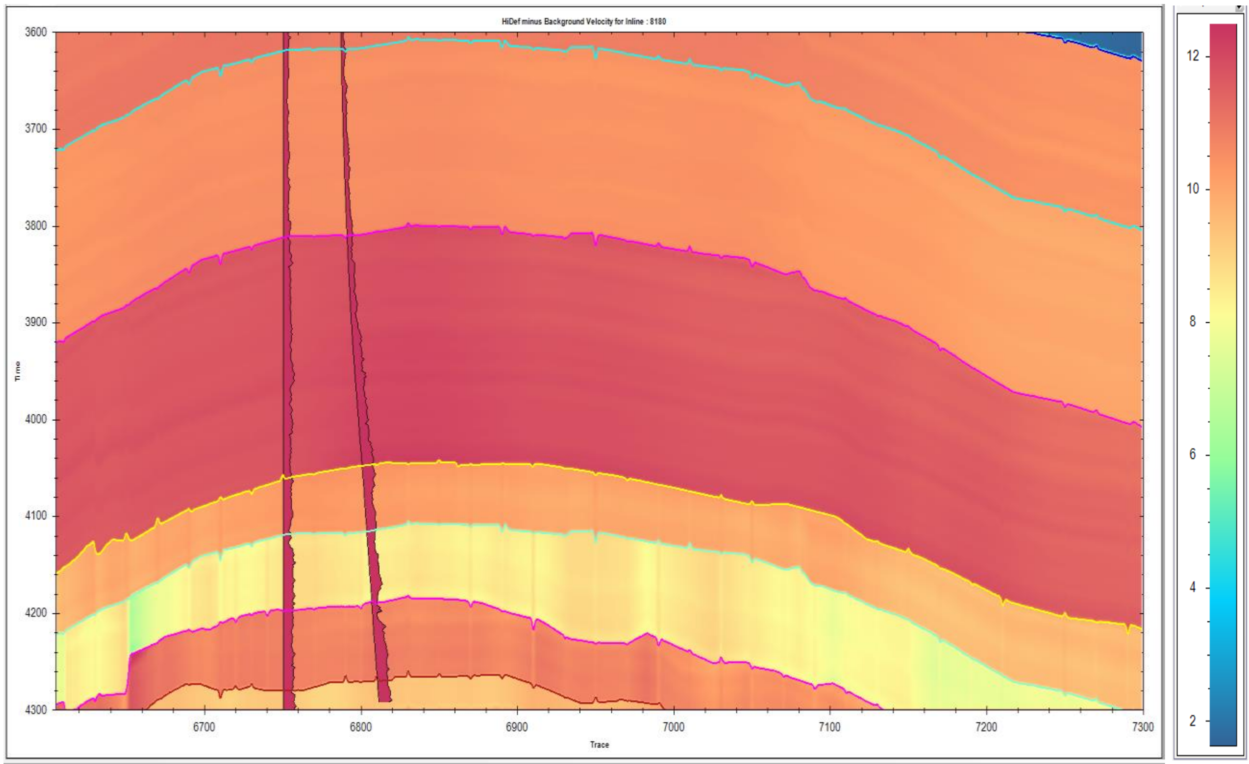


Figure 17: Residual difference between stacking velocity and well-logs velocity data.

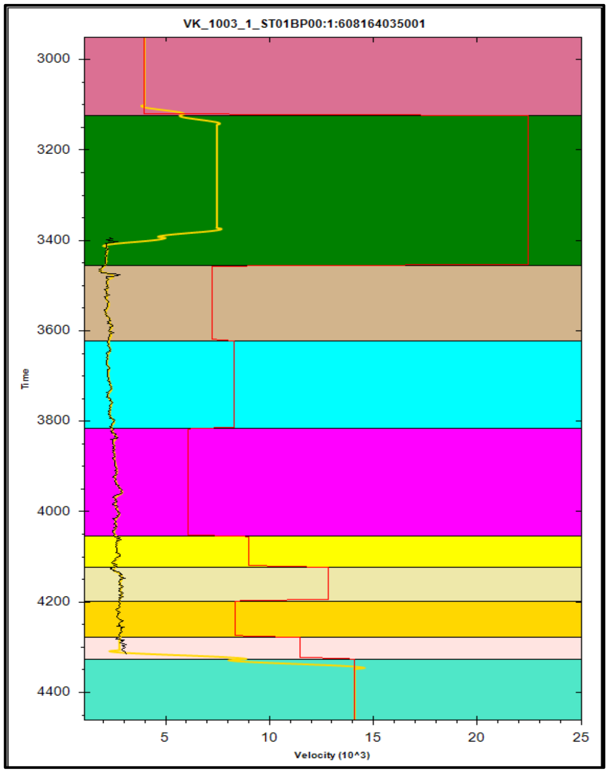


Figure 18: Sonic log (black) and seismic velocity (red) trend for low-frequency macro-model.

Details of parameter estimation can be seen in Appendix B. Next is the parameter testing phase. This was accomplished by applying the estimated parameters to the selected traces around the well location as shown in Figure 19 and Figure 20 for both the 2D line and 3D volume, respectively. There are two other critical parameters that are calculated at the testing phase, namely convergence tolerance (EPS) and initial temperature (T0), which are estimated to be 0.001 and 0.1 respectively.

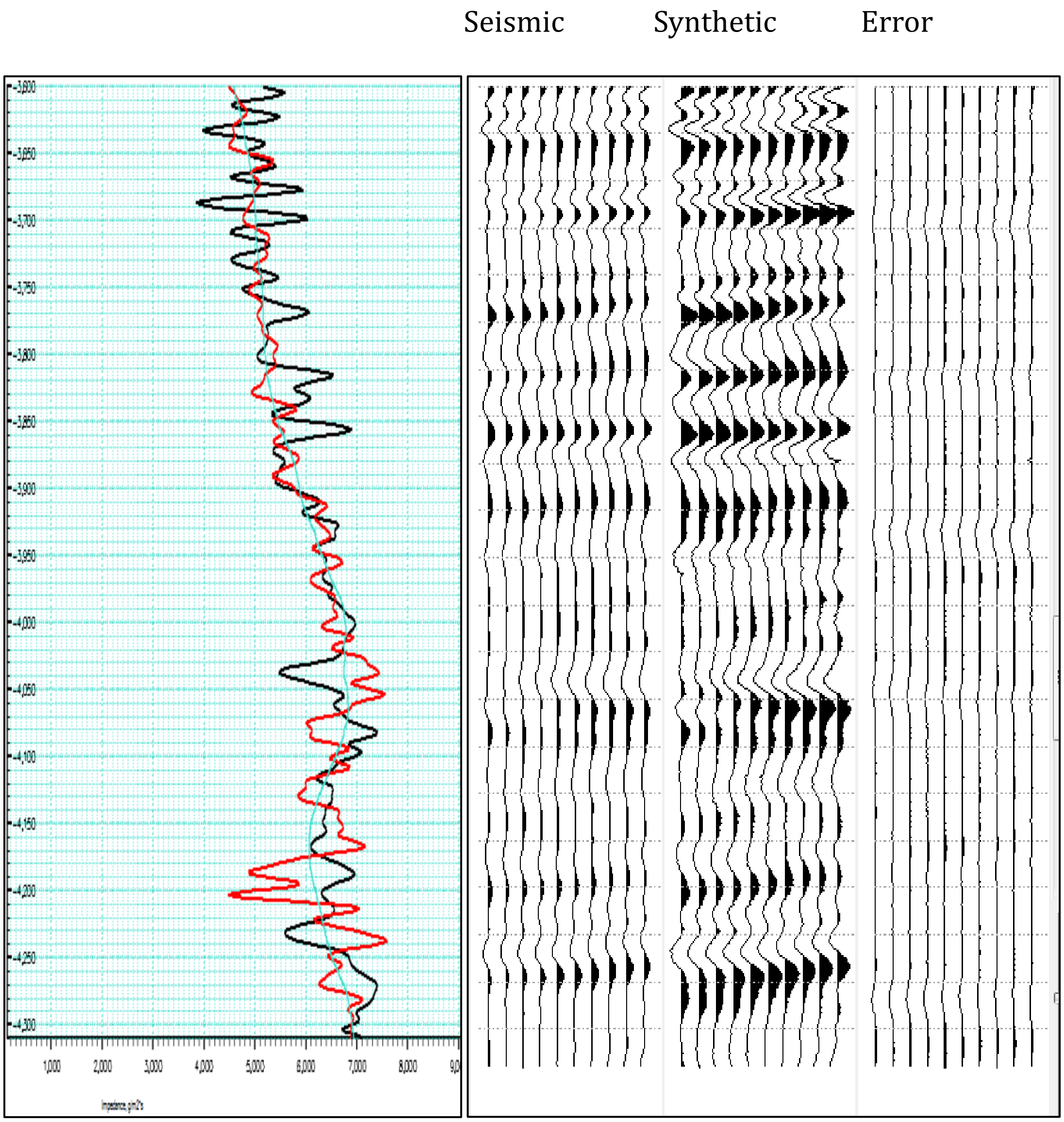


Figure 19: Estimation of inversion parameters using selected traces for 2D line.
Red: Impedance from well logs. Black: Inverted impedance. Green: background/macro-model.

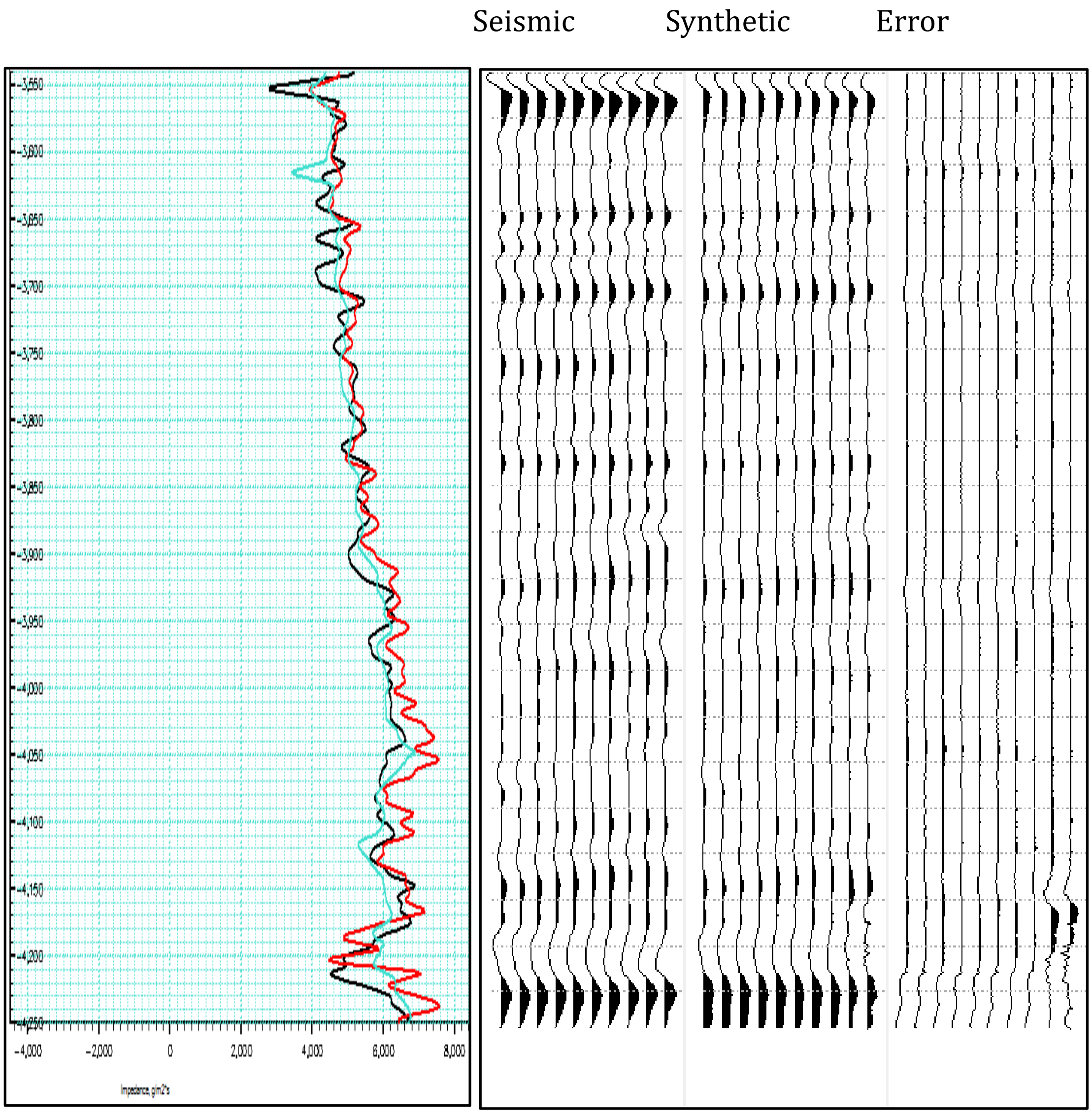


Figure 20: Estimation of inversion parameters using selected traces for seismic volume.
Red: Impedance from well logs. Black: Inverted impedance. Green: background/macro-model.

In both Figures 19 and 20, it can be observed that the error is minimal, with a maximum correlation between the inverted impedance and impedance from the log, using smooth background velocity as a constraint. The background velocity was built using the Hidef tool in Kingdom suite. See Figure 8 for the workflow.

The inversion is then run using the estimated parameters from Figures 19 and 20 on both the 2D line and the 3D volume respectively. Figures 21 and 22 show a cross section view (2D) and map view (3D) of the both results. The inverted relative impedance (Figure 21b) shows an improvement in vertical resolution and lateral continuity when compared to the original seismic (Figure 21a). This result is crucial to enhancing qualitative interpretations of seismic data. The absolute impedance result (Figure 21c) gives an actual impedance trend which is essential to understand exact bed thickness and improves lateral continuity.

A similar trend can be seen on times-slices (Figure 22) generated from the seismic volume at 4.2secs. The absolute impedance time slice (Figure 22c) shows clearer structural and stratigraphic delineation of sand bodies, which is very useful when studying reservoir connectivity, and volumetric reserve estimations.

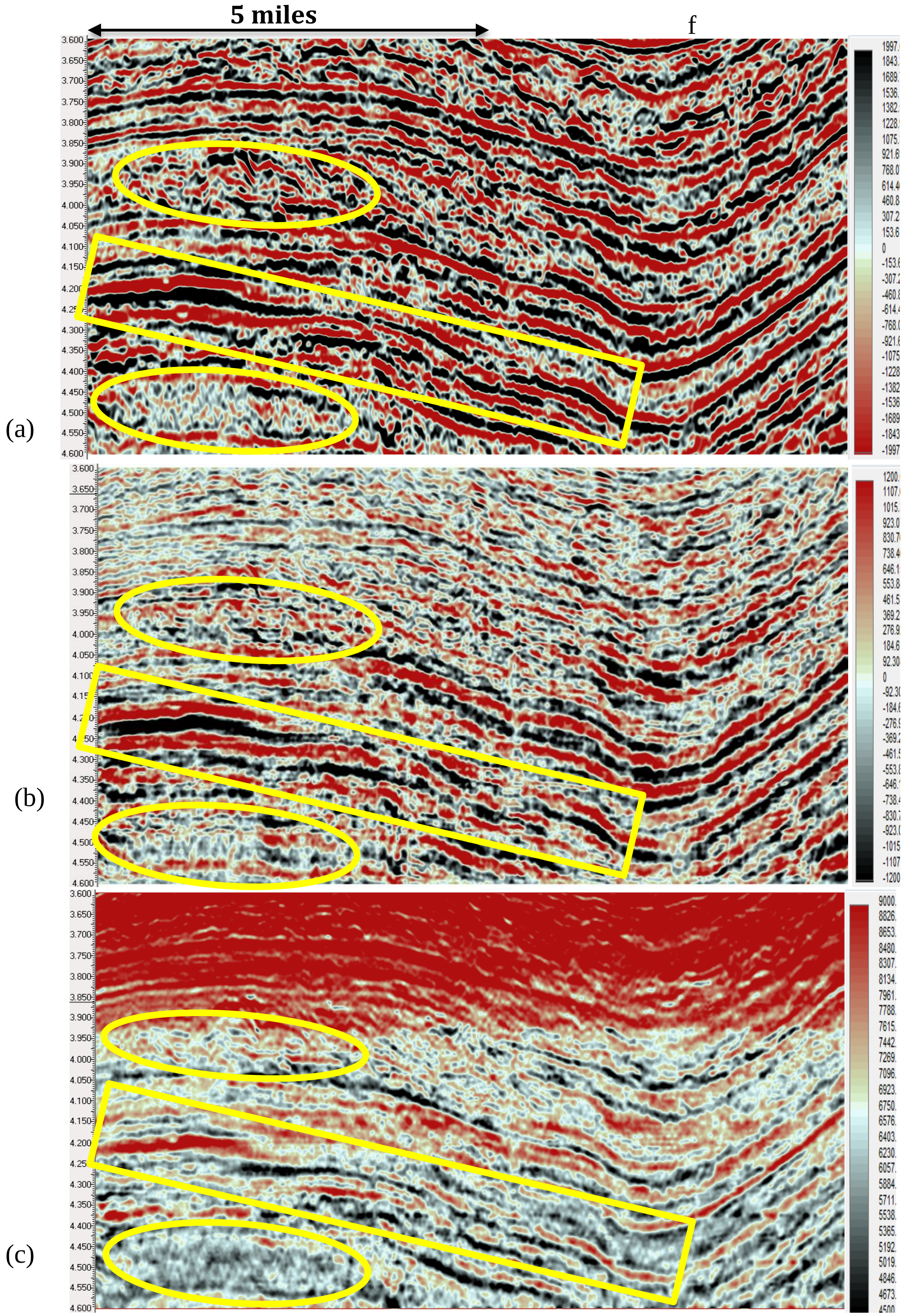


Figure 21: (a) Seismic data. (b) Inverted relative impedance. (c) Inverted absolute impedance. Notice the improved vertical resolution and lateral continuity of the of the layers in the highlighted region

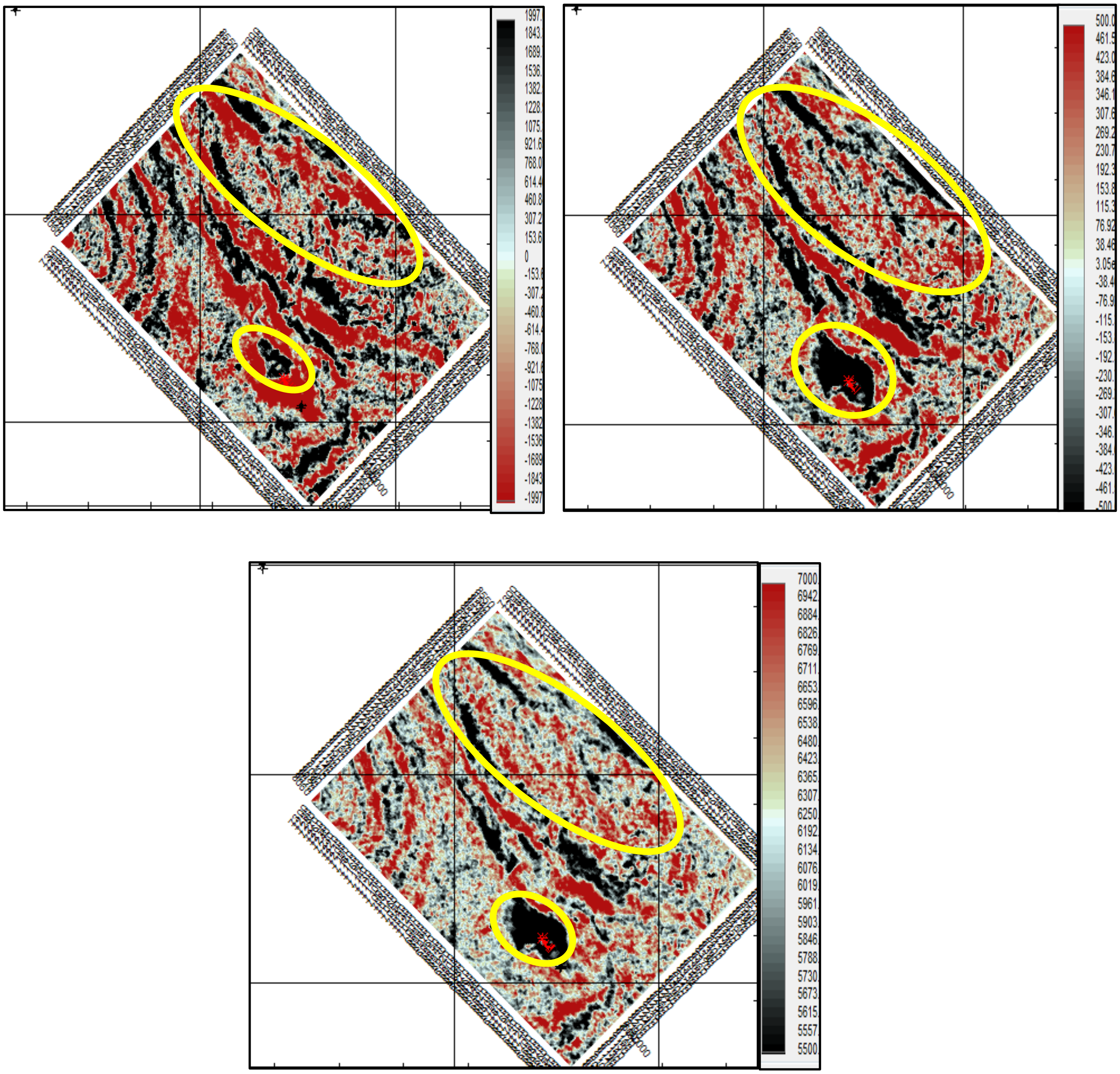


Figure 22: Time-slice at 4.2secs for (a) Original seismic data. (b) Relative impedance. (c) Absolute impedance. Highlighted in yellow demonstrate better stratigraphic delineation, and improved reservoir connectivity

4.0 DISCUSSION

Inversion results depend greatly on the quality of the seismic-to-well tie and log calibration. Wavelet estimation is also a crucial step in any inversion procedure. The latter is required in the forward modelling needed to generate a synthetic model that is used to drive the inversion process.

The regularization parameter λ acts as the only constraint in the BPI method. Quality time should be spent on estimating the correct λ value. If done properly, seismic bandwidth can be greatly increased, otherwise, the final output will not be a proper representation of the true subsurface geology. The method was developed as an alternative method to *a priori* information, hence, does not require nor depend on any form of an initial model.

SA inversion on the other hand is a stochastic process that is highly independent of the *a priori* information. The *a priori* information in this case is the background model. This implies that one can generate reliable inversion results even when the initial model is not very good. This is not the case for the BPI inverted acoustic impedance as it is highly dependent on the initial (impedance) model. Since seismic data are bandlimited, therefore, low and high frequencies are missing in the objective function, hence, the need to interpolate the missing low frequencies into the macro-model.

Building a macro-model that contains the missing low frequencies is an important aspect of the model-based inversion. The model is needed to generate absolute elastic properties from seismic data. This was achieved by interpolating low-frequency components from the sonic well-log with seismic velocities within our 600ms interval of interest using the Hidef tool in Kingdom.

The background model also can be generated from impedance using well logs but this is more effective in areas with non-complex geological formations. Hence, it is more effective to build a macro-model when inversion is applied to a 3D seismic volume. This will account for the structural dip and possible lateral changes in the reservoir or area of interest.

Parameter estimation is a very crucial process in SA inversion. The quality of the results depends on how much time is devoted to this stage. These parameters are derived from *a priori* information, but the inversion algorithm itself does not use the *a priori* information during the inversion process.

Two parameters that control SA inversion are the convergence tolerance (EPS) and initial temperature (T0). The EPS can be seen as a trigger-point where the algorithm stops on the assumption that a solution has been found. The time taken for the algorithm to find a solution depends on this number. The higher the number, the sooner it stops searching for a solution and vice-versa. On the other hand, T0 controls how much 'energy' is put into the inversion algorithm's search for a solution, *i.e.*. energy of perturbation. The higher this value, the longer the algorithm will rigorously explore the search space for solutions.

Once the wavelet and other parameters are estimated, the next step is to apply the SA algorithm to a few selected traces. Parameters to be chosen include starting models, impedance perturbation windows, seismic scalar, seismic and macro-model weights, SA initial temperature, convergence tolerance, etc. Optimal parameters are determined in such a way that the inverted impedance near the well shows a best match with the impedance log and that the misfit between the synthetic trace and seismic trace is at a minimum.

Although the workflow for SA inversion has been designed to be more computationally efficient than the standard SA algorithm, it is an iterative and time-consuming process. The processing time is determined by the convergence tolerance (EPS) and the initial temperature (T0). They both have values ranging from 0.001<= EPS => 0.1 and 0.001 < T0 <= 0.01

A major advantage of the SA algorithm is that it generates both absolute and relative impedance from post-stack seismic data, which provides qualitative and quantitative characterization of reservoirs. Unlike the BPI technique, SA inversion depends on the determination of many parameters before it can run, so there is a trade-off between the quality of earth models and the time spent computing them.

Improved vertical resolution comes from the macro-model for the stochastic method and from the regularization parameter with a wedge model for the deterministic method. The macro-model and regularization parameter are based on the L_1 norm error function. They also act as a constraint for the respective methods of inversion. The error term, *i.e.* the least error, between the synthetic model data and seismic data is based on L_2 norm and the L_1 norm for both the SA and BPI methods. Both methods can be extended to the pre-stack domain for the simultaneous inversion of acoustic and shear impedances from angle stack traces.

5.0 CONCLUSIONS

We have applied deterministic and stochastic global methods of inversion to invert for reflectivity and acoustic impedance using a 3D post-stack seismic dataset from the Gulf of Mexico. The BPI method is a more efficient method for removing the effects of the source wavelet and generating a high resolution reflectivity volume. The efficiency of the BPI method is largely due to the wedge dictionary used in the inversion process, producing a reflectivity volume with higher vertical resolution. The quality of inverted reflectivity can be greatly distorted by noise. This problem can be overcome with careful selection of the regularization parameter. The BPI method has the potential to enhance frequencies higher than the seismic bandwidth without amplifying the noise. This in turn implies that the algorithm can improve detectability, resolution, and lateral continuity of thin layers.

The SA inversion uses an over-parameterized scheme, *i.e.*, a series of micro-layers with each layer having the same acoustic impedance. The background model provides the bounds for the algorithm to search the model space. It also helps reduce the problem of seismic band-limitation as the low frequency component of impedance is interpolated into the *a priori* model.

Both algorithms are robust and can find a global solution of the earth model without being trapped in a local minimum. Parameter estimation is iterative and time-consuming when applied to a large seismic volume. Results show an improved vertical resolution, lateral continuity, and better reservoir connectivity, which are essential for high resolution reservoir characterization, volumetric estimation, and fluid mapping.

The SA method finds greater use in reservoir characterization because it finds missing impedance trends using Monte-Carlo simulation. Results from the BPI method reveal that it has greater potential to resolve beds below tuning. BPI shows better lateral continuity because it takes advantage of the gradient-oriented greedy search and applied lateral-smoothness constraint. When the goal is to invert acoustic impedance using both algorithms, a reliable background model is crucial to successful processing. The SA inversion algorithm is very iterative and hence, cumbersome, while, BPI is very straightforward once the proper regularization parameter is determined.

5.1 Future work

Blocky acoustic impedance can be generated from BPI results by implementing the recursive relationship between the reflectivity series and acoustic impedance. A very good low frequency macro-model can be generated from interpolating well-log data and low-pass filtering.

It is expected that the blocky impedance results via BPI will show a better stratigraphic and structural actualization due to it’s to delineate layers thinner than tuning thickness. The same recursive inversion technique can be used to invert the reflectivity volume from the absolute acoustic impedance generated from the SA algorithm.

References

Ache, B., Bylsma, D., Crossett K,. David C, Edwards A., Wallace, N., 2008, National Ocean Service, NOAA. Gulf of Mexico at a Glance. Washington, D.C.: U.S. Department of Commerce, National Oceanic and Atmospheric Administration.

Barclay F., Bruun A., Alfaro J., Cooke A., Cooke D., Godfrey R., Volterrani S., Roberts R., 2008, Seismic Inversion: Reading between the Lines, Oilfield Review, p. 42-63.

Bormann, C., Cowgill, M., Ginal, E., Mandley, R., Thompson, J., Roberts, G. and Archer, S. 2006, Huge potential still waiting in the Gulf of Mexico: GEO Expro Issue No. 4/5, P. 110-115.

Bouma, A.H., J.M. Coleman, and DPSP Leg 96 Shipboard Scientists. 1985. Mississippi Fan:Leg 96 program and principal results, Chapter 36. In: Bouma, A.H., W.R. Normark, and N.E. Barnes, eds. Submarine Fans and Related Turbidite Sequences. New York: Springer-Verlag.

Cheman K.C., 2006, Optimization techniques for solving basis pursuit problems.

Chen, S. S., D. L. Donoho, Michael, and A. Saunders, 2001, Atomic decomposition by basis pursuit: SIAM Rev, 129-159.

Chen, S. S., 1995, Basis pursuit: Ph.D. Thesis, Stanford University.

Cooke,D. A., and Schneider,W. A., 1983, Generalized linear inversion of reflection seismic data: Geophysics, 48, 665–676.

Corana, A., Marchesi, M., Martini, C., and Ridella, S., 1987, Minimizing multimodal functions of continuous variables with the simulated annealing algorithm: ACM Trans. of Math. Soft. 13, 262–280.

Debeye H W J and van Riel P 1990 lP-norm deconvolution Geophys. Prospect. 38 381–404.

Elad, M., 2011, Sparse and Redundant: From Theory to Applications in Signal and Image Processing, Springer New York Dordrecht Heidelberg, London.

Emilson Pereira Leite (2010). Seismic Model-Based Inversion Using Matlab, Matlab - Modelling, Programming and Simulations, Emilson Pereira Leite (Ed.), ISBN: 9789533071251.

Goffe, W. L., Ferrier, G. D., and Rogers, J., 1994, Global optimization of statistical functions with simulated annealing: J. Economet., 60, 65–100.

Filippova K, Kozhenkov A, and Alabushin A., 2011, Seismic inversion techniques: choice and benefits first break, 29.

Francis, A.M., 2006a, Understanding stochastic seismic inversion: First break Vol 24, No 11, pp. 69-77.

Francis A.M, 2006b, Understanding stochastic inversion: part 2 First Break 24 79–84

Ingber L., 1989 Very fast simulated re-annealing Math. Comput. Model. 12 967–73.

J. Dreo A. Petrowski, P. Siarry E. Taillard, Metaheuristics for Hard Optimization, Springer-Verlag, 2006.

Keary, P., M. Brooks and I. Hill, 2002, An Introduction to Geophysical Exploration, 3rd edition Blackwell Science, 2002, ISBN0632049294.

Kirkpatrick, S., Gelatt, C. D., Jr., and Vecchi, M. P., 1983, Optimization by simulated annealing. Science 220, 4598 (May 1983), 671-680.

Li, Q., 2001, LP sparse spike inversion, Strata Technique Document, Hampson-Russell Software Services Ltd.

Ma, X.Q. [2001] A constrained global inversion method using an over-parameterised scheme: application to poststack seismic data. Geophysics 66, 613-626.

Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., and Teller, E., 1953, Equation of state calculations by fast computing machines: J. Chem. Phys., 21, 1087–1092.

Moyen, R., and Doyen, P. M., 2009, Reservoir connectivity uncertainty from stochastic seismic inversion: 79th Annual International Meeting, Society of Exploration Geophysicist, Expanded Abstracts, 2378-2382.

Oldenburg D W, Scheuer T and Levy S 1983 Recovery of the acoustic impedance from reflection seismograms Geophysics 48 1318–37.

Posamentier, H.W., and V. Kolla, 2003, Seismic Geomorphology and Stratigraphy of Depositional Elements in Deep-Water Settings: The Journal of Sedimentary Research, 73, 367–388.

Puryear, C. I. and Castagna, J. P., 2008, Layer-thickness determination and stratigraphic interpretation using spectral inversion: Theory and application: Geophysics, 73, R37-R48.

Puryear, C.I., 2012, Constrained Least-Square Spectral Analysis: Ph.D. thesis, University of Houston.

Russell B H., 1988, Introduction Seismic Inversion Methods, SEG Course Notes Series, No 27. Issue 2 of course note series. ISBN: 0931830656

Russell, B. and Hampson, D., 1991, A comparison of post-stack seismic inversion methods: Ann. Mtg. Abstracts, Society of Exploration Geophysicists, 876-878.

Sacchi, M. D. and Ulrych, T. J., 1995, High-resolution velocity gathers and offset space reconstruction: Geophysics, 60, 1169-1177.

Sen, M. K., and Stoffa, P. L., 1991, Nonlinear one-dimensional seismic waveform inversion using simulated annealing: Geophysics, 56, 1624–1638.

Sen M K and Stoffa P L 1995 Global Optimization Methods in Geophysical Inversion (Amsterdam: Elsevier) p279.

Sen, M.K., 2006, Seismic Inversion (Society of Petroleum Engineers: USA).

Srivastava, R. P., and Sen, M. K., 2009, Fractal-based stochastic inversion of poststack seismic data using very fast simulated annealing J. Geophys. Eng. 6 412–25.

Szu, H., Hartley, R., 1987, Fast simulated annealing. Phys. Lett. A 122, 157–162.

Tarantola, A., 1984, Linearized inversion of seismic reflection data: Geophys. Prosp. 32, 998-1015.

Vestergaard, P. D., and Mosegaard, K., 1991, Inversion of post-stack seismic data using simulated annealing: Geophys. Prosp., 39, 613–624.

Widess., 1985, How thin is a thin bed: Geophysics, 50, 2061-2065.

Xi, X., Ling, Y., Zou, Z., Sun, D., Lin, J., Wang, J. and Wang, H., 2013, The application of a low-frequency model constrained by seismic velocity to acoustic impedance inversion: 83th Annual International Meeting, Society of Exploration Geophysicist, Expanded Abstracts, 3278-3282.

Zhang, R., 2010, Basis pursuit seismic inversion: Ph.D. thesis, University of Houston.

Zhang, R., and J. Castagna, 2011, Seismic sparse-layer reflectivity inversion using basis pursuit decomposition: Geophysics, 76, no. 6, R147–R158**.**

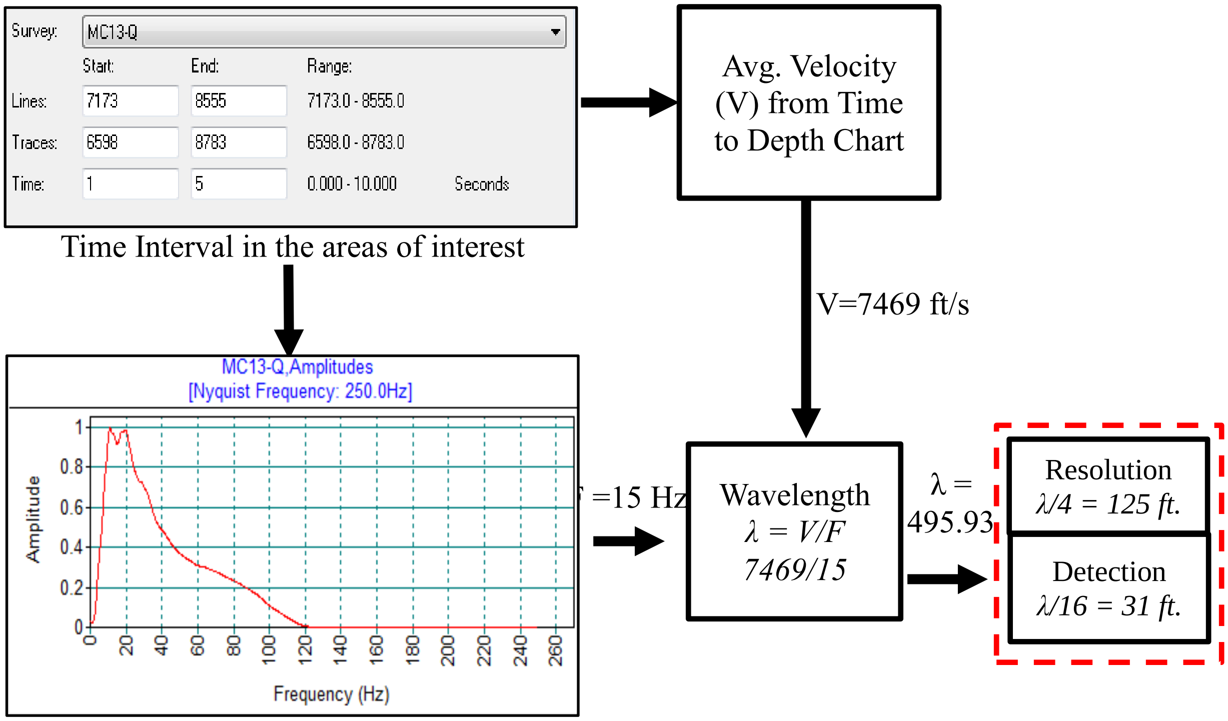
Zhang R., Sen M and Srinivasan S, 2013, A prestack basis pursuit seismic inversion Geophysics, 78, R1–11**.**

Zhang, R., Daley T. M., and Vasco, D., 2015, Improving thin-bed resolution: Application of a sparse-layer inversion on 3D seismic observations from the In Salah carbon dioxide storage project. Interpretation, 3(3), SS65-SS71

Zhang, R., Sen M.K., Phan, S., and Srinivasan, S., 2012, Stochastic and deterministic seismic inversion methods for thin-bed resolution. J. Geophys. Eng. 9 (2012) 611–61

Appendices

Appendix A: Pre-inversion analysis of geophysical dataset



Survey:	MC13-Q		
	Start:	End:	Range:
Lines:	7173	8555	7173.0 - 8555.0
Traces:	6598	8783	6598.0 - 8783.0
Time:	1	5	0.000 - 10.000 Seconds

Time Interval in the areas of interest

Avg. Velocity (V) from Time to Depth Chart

V=7469 ft/s

=15 Hz

Wavelength
λ = V/F
7469/15

λ = 495.93

Resolution
λ/4 = 125 ft.

Detection
λ/16 = 31 ft.

Figure A1: Limit of seismic resolution and detection. Bandwidth 10-40Hz; dominant frequency (F) ~15 Hz

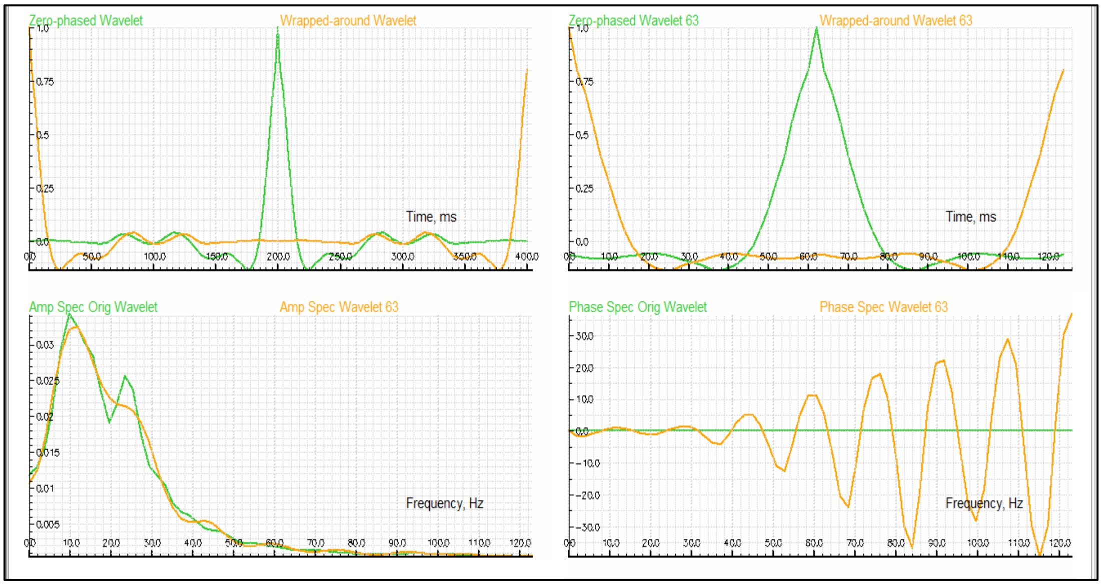


Figure A2: Wavelet estimation with both amplitude and phase spectrum.

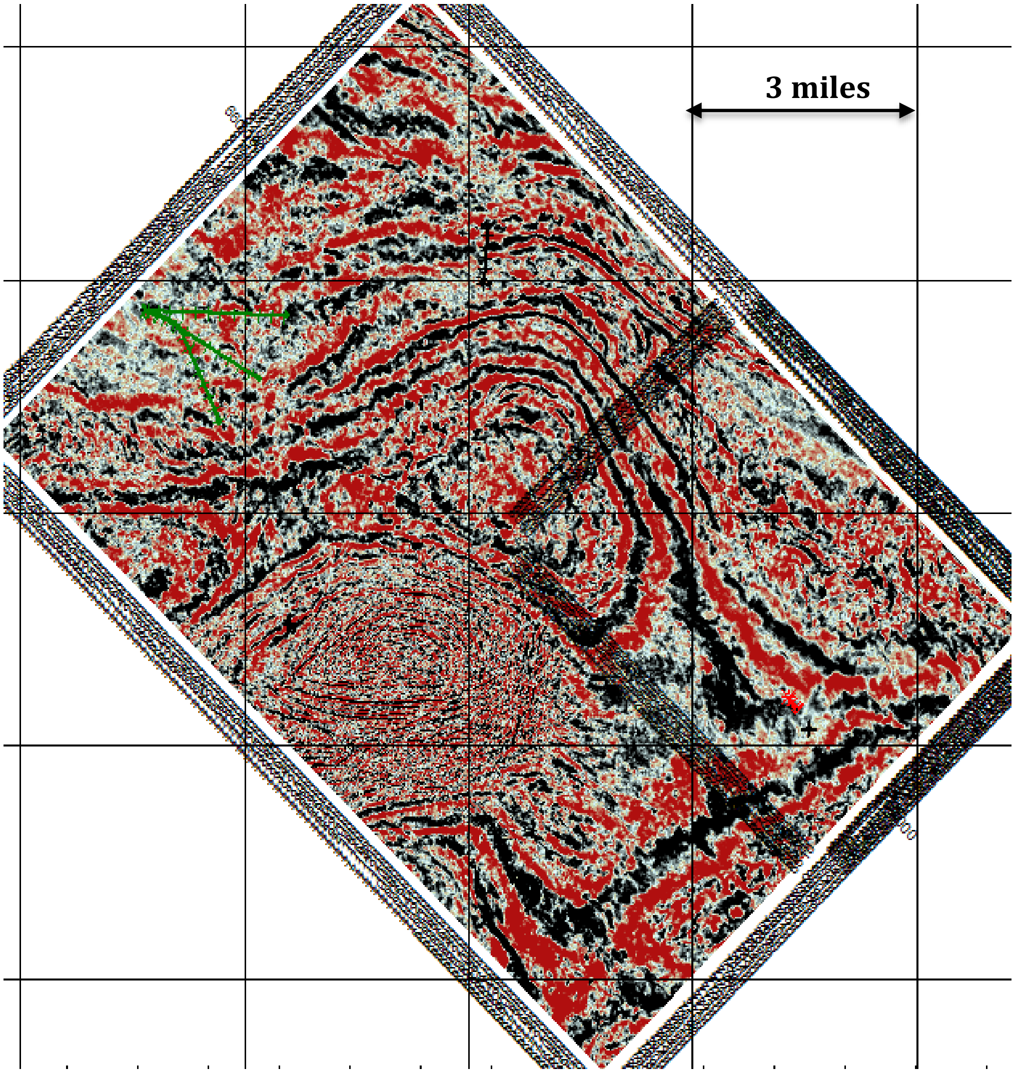
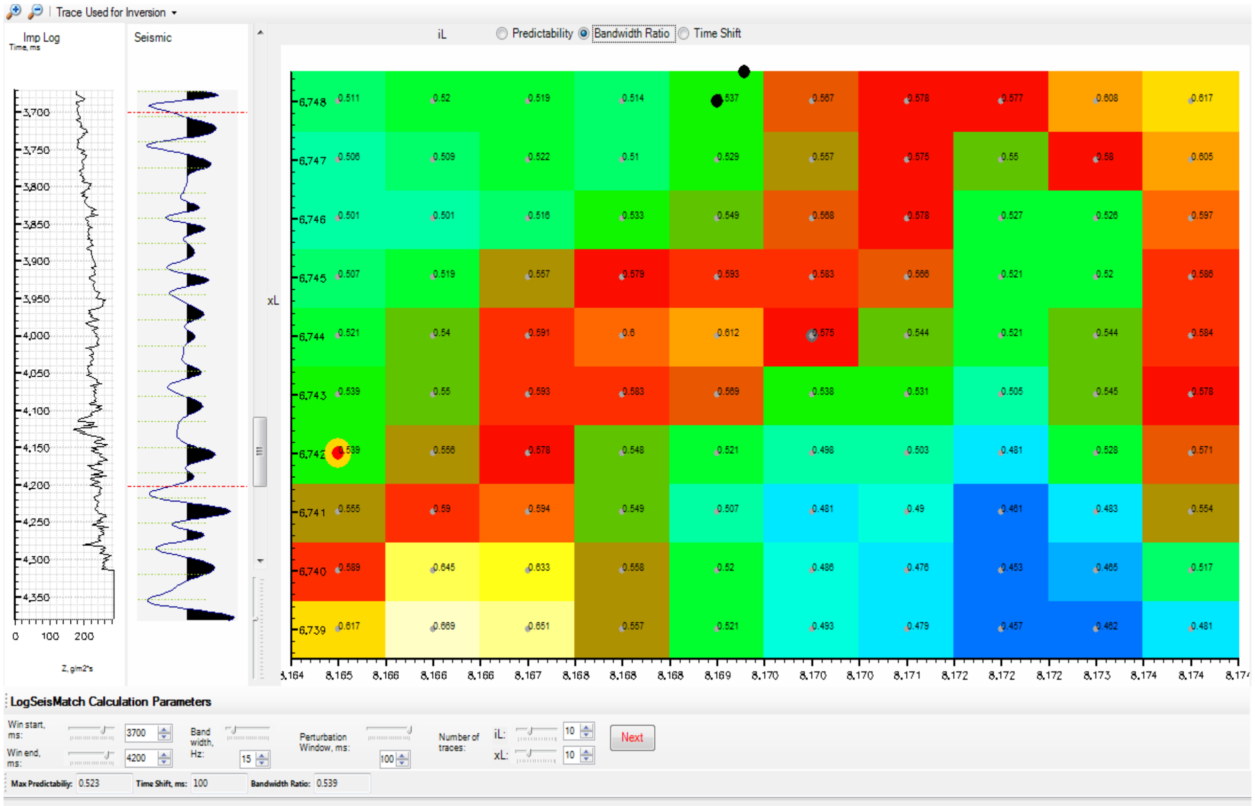


Figure A3: 4secs time-slice showing the extent of the acquired 3D seismic data (~100 square miles) with the location of wells



○ Predictability ● Bandwidth Ratio ○ Time Shift

xL										
6,748	0.511	0.52	0.519	0.514	0.537	0.567	0.578	0.577	0.608	0.617
6,747	0.506	0.509	0.522	0.51	0.529	0.557	0.575	0.55	0.58	0.605
6,746	0.501	0.501	0.516	0.533	0.549	0.568	0.578	0.527	0.526	0.597
6,745	0.507	0.519	0.557	0.579	0.593	0.583	0.566	0.521	0.52	0.586
6,744	0.521	0.54	0.591	0.6	0.612	0.575	0.544	0.521	0.544	0.584
6,743	0.539	0.55	0.593	0.583	0.569	0.538	0.531	0.505	0.545	0.578
6,742	0.539	0.556	0.578	0.548	0.521	0.498	0.503	0.481	0.528	0.571
6,741	0.555	0.59	0.594	0.549	0.507	0.481	0.49	0.461	0.483	0.554
6,740	0.589	0.645	0.633	0.558	0.52	0.486	0.476	0.453	0.465	0.517
6,739	0.617	0.669	0.651	0.557	0.521	0.493	0.479	0.457	0.462	0.481

LogSeisMatch Calculation Parameters

Win start, ms: 3700

Win end, ms: 4200

Band width, Hz: 15

Perturbation Window, ms: 100

Number of traces: iL: 10 xL: 10

Next

Max Predictabiliy: 0.523 Time Shift, ms: 100 Bandwidth Ratio: 0.539

Figure A4: LogSeisMatch finds the best trace that correlates to the well for the purpose of wavelet estimation. This trace is also used in seismic scalar estimations and determination of the inversion parameters. The trace is selected automatically (red dot) according to the highest predictability value as seen on the map; but there is also an option to select the trace manually.

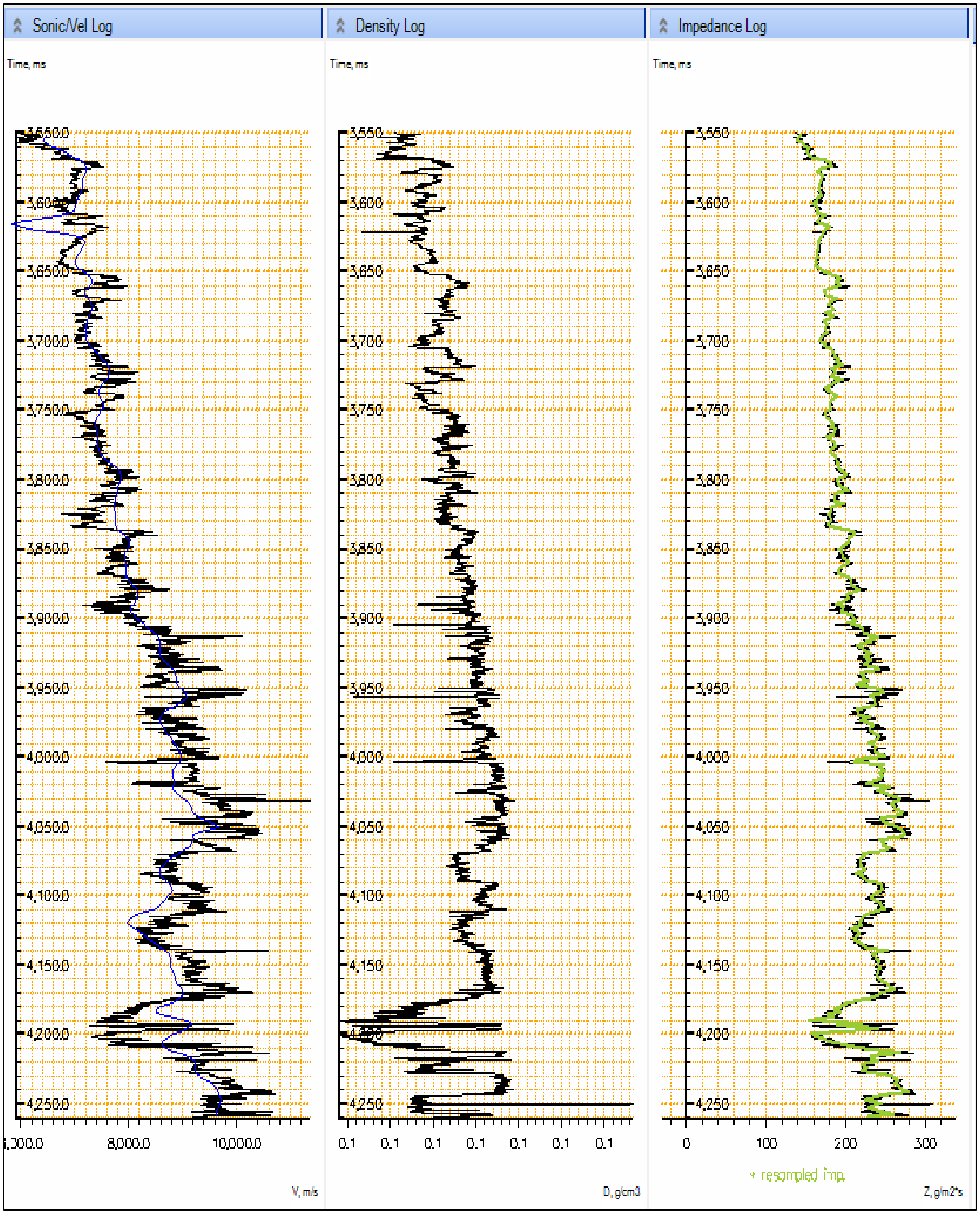
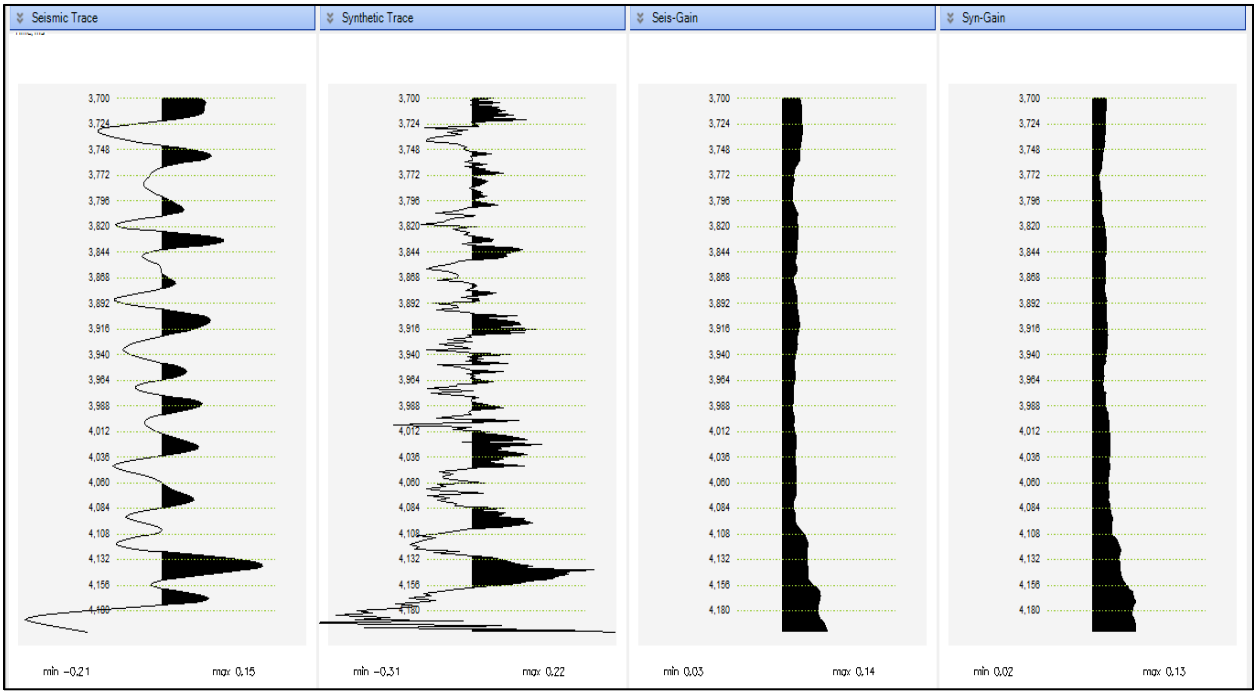


Figure A5: Detailed impedance generation (resampled) from both sonic and density log i.e Well log calibration.



Appendix A6: Seismic scalar estimation for gain control. This works out a scalar of seismic data by comparing the amplitudes of a synthetic trace with that of the seismic trace. The synthetic trace is generated using the log reflectivity and the estimated wavelet.

Appendix B: SA inversion Job Parametrization

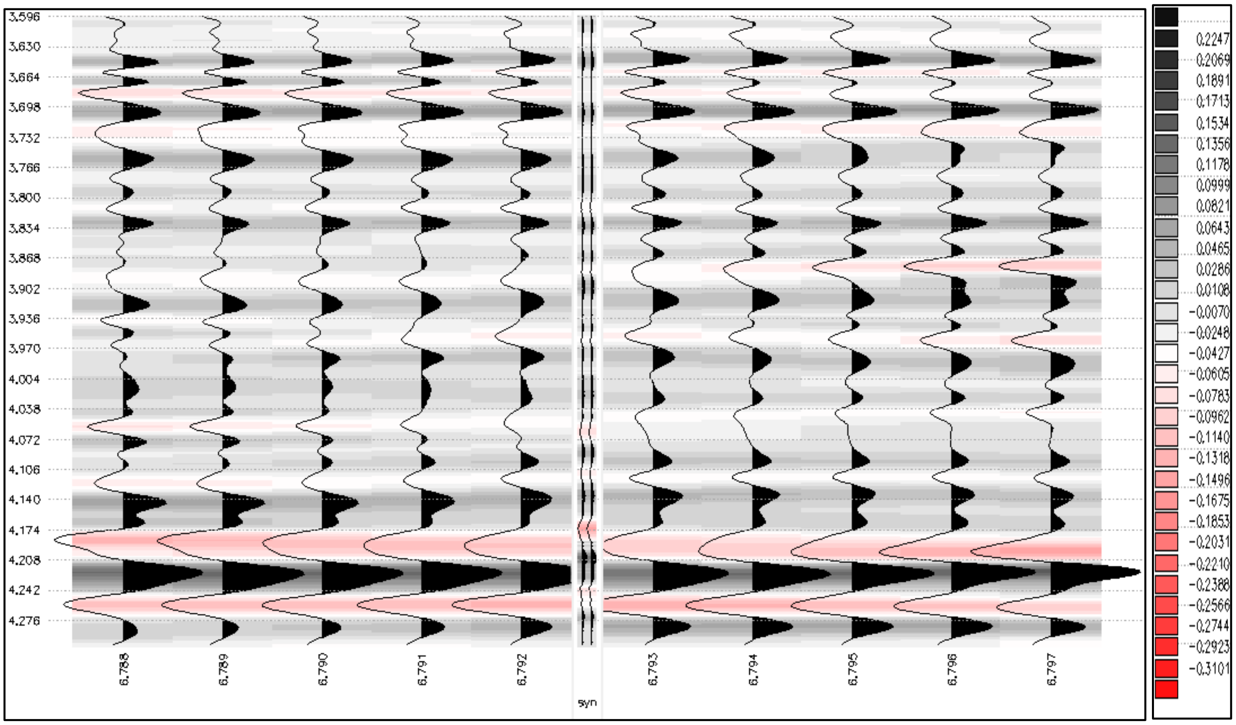


Figure B1: Inversion Test phase showing strong correlation between seismic traces and synthetic

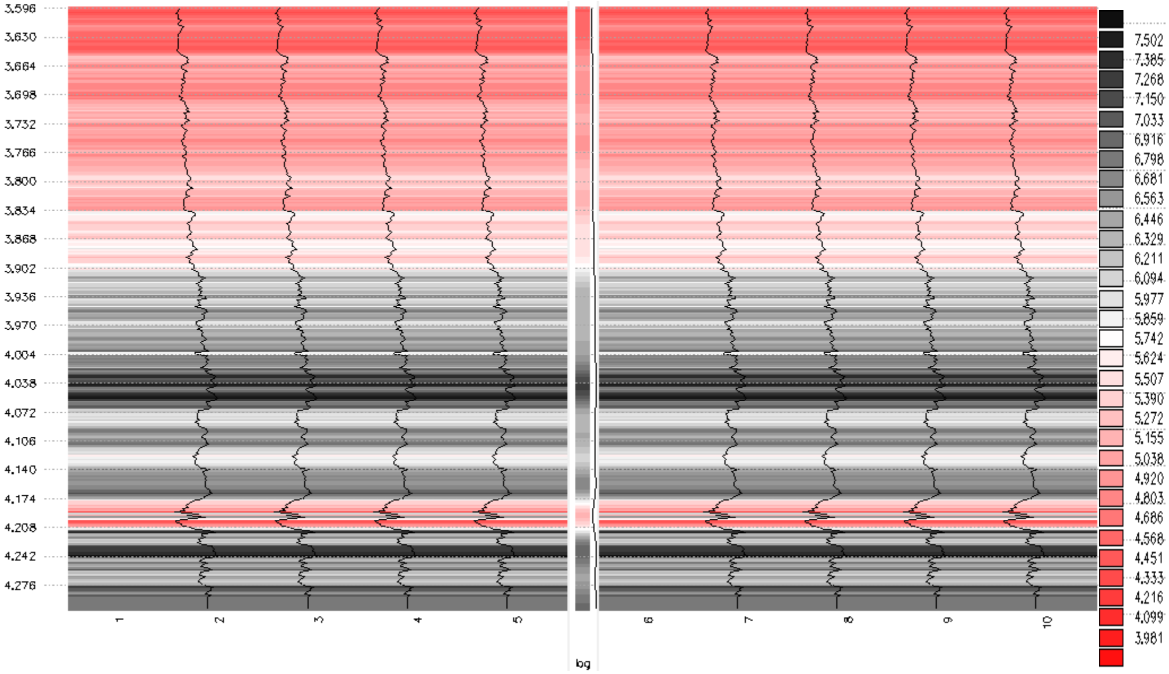


Figure B2: Macro-model from impedance log with few selected traces

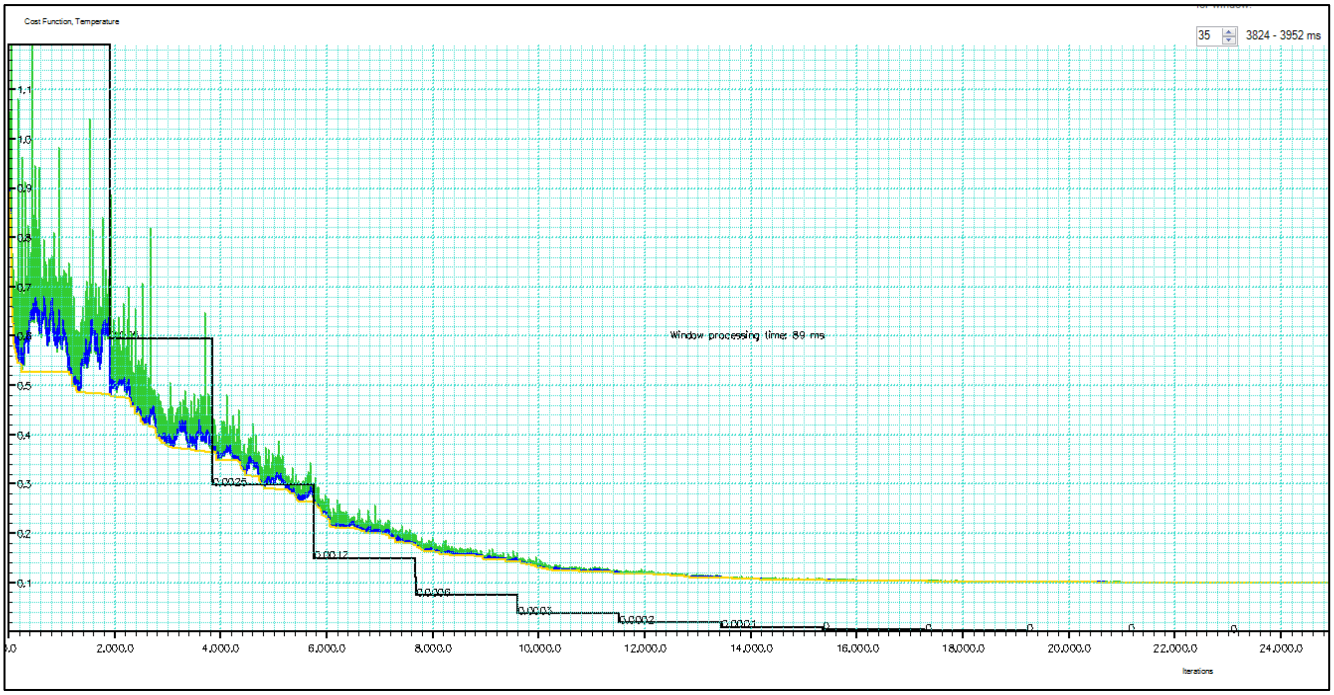
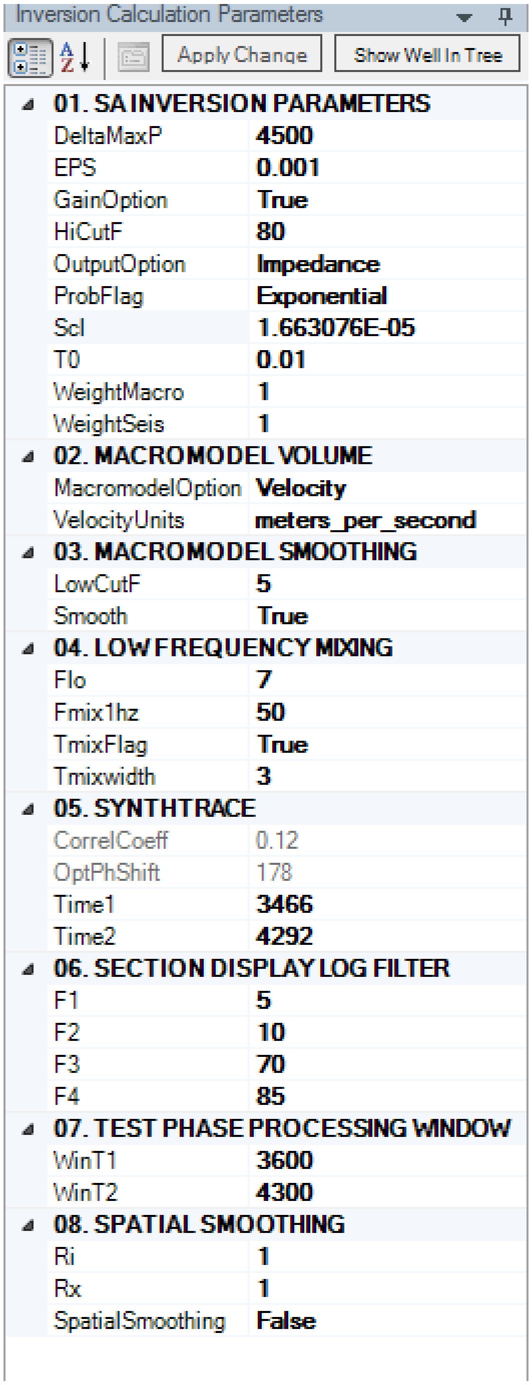
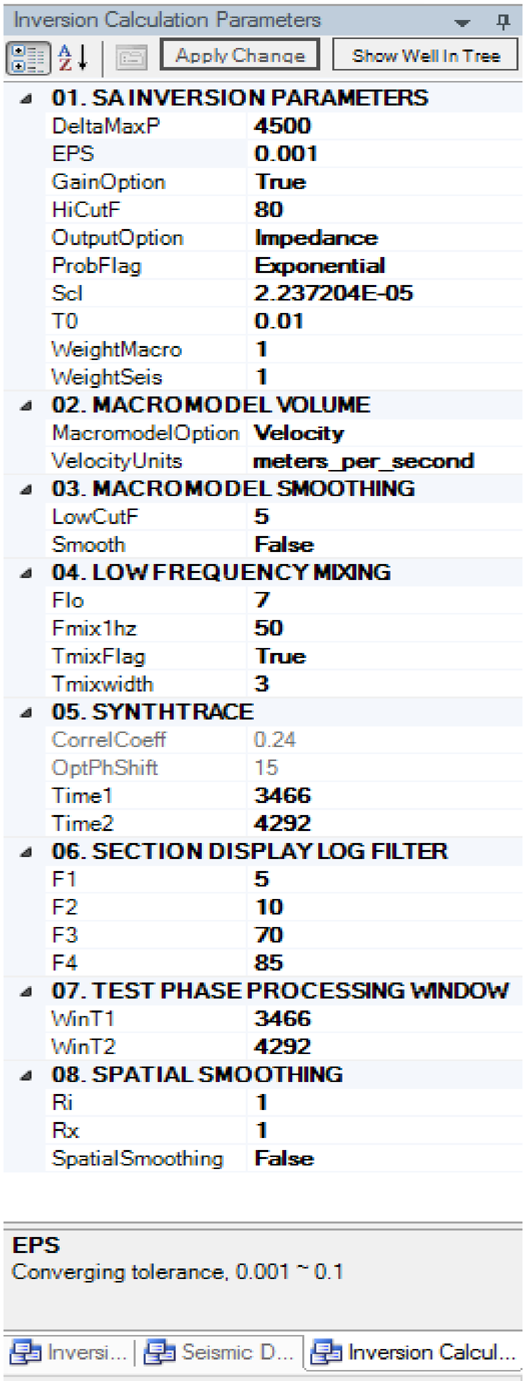


Figure B3: Cost Function estimation (best) or SA Algorithm chart. This is required in order to select optimal EPS and T0 parameters for both 2D and 3D Volume. This graph shows objective function variations as number of iterations increases. The green curve is for all objective functions evaluated regardless of their acceptances or rejections. The blue curve is for those accepted which include both downhill and uphill moves. The yellow curve shows the best search so far within a constant temperature. The black curve is the temperature steps.



01. SA INVERSION PARAMETERS	
DeltaMaxP	4500
EPS	0.001
GainOption	True
HiCutF	80
OutputOption	Impedance
ProbFlag	Exponential
Scl	1.663076E-05
T0	0.01
WeightMacro	1
WeightSeis	1
02. MACROMODEL VOLUME	
MacromodelOption	Velocity
VelocityUnits	meters_per_second
03. MACROMODEL SMOOTHING	
LowCutF	5
Smooth	True
04. LOW FREQUENCY MIXING	
Flo	7
Fmix1hz	50
TmixFlag	True
Tmixwidth	3
05. SYNTHTRACE	
CorrelCoeff	0.12
OptPhShift	178
Time1	3466
Time2	4292
06. SECTION DISPLAY LOG FILTER	
F1	5
F2	10
F3	70
F4	85
07. TEST PHASE PROCESSING WINDOW	
WinT1	3600
WinT2	4300
08. SPATIAL SMOOTHING	
Ri	1
Rx	1
SpatialSmoothing	False



01. SA INVERSION PARAMETERS	
DeltaMaxP	4500
EPS	0.001
GainOption	True
HiCutF	80
OutputOption	Impedance
ProbFlag	Exponential
Scl	2.237204E-05
T0	0.01
WeightMacro	1
WeightSeis	1
02. MACROMODEL VOLUME	
MacromodelOption	Velocity
VelocityUnits	meters_per_second
03. MACROMODEL SMOOTHING	
LowCutF	5
Smooth	False
04. LOW FREQUENCY MIXING	
Flo	7
Fmix1hz	50
TmixFlag	True
Tmixwidth	3
05. SYNTHTRACE	
CorrelCoeff	0.24
OptPhShift	15
Time1	3466
Time2	4292
06. SECTION DISPLAY LOG FILTER	
F1	5
F2	10
F3	70
F4	85
07. TEST PHASE PROCESSING WINDOW	
WinT1	3466
WinT2	4292
08. SPATIAL SMOOTHING	
Ri	1
Rx	1
SpatialSmoothing	False

EPS
Converging tolerance, 0.001 ~ 0.1

Figure B4: Overview of other selected inversion parameters for the 2D line and 3D Volume respectively. The difference between both are scaling factor and perturbation window. Imported wavelet was used for the inversion, hence, the correlation coefficient being displayed is unconnected to the process.

Appendix C: Other results using SA algorithm

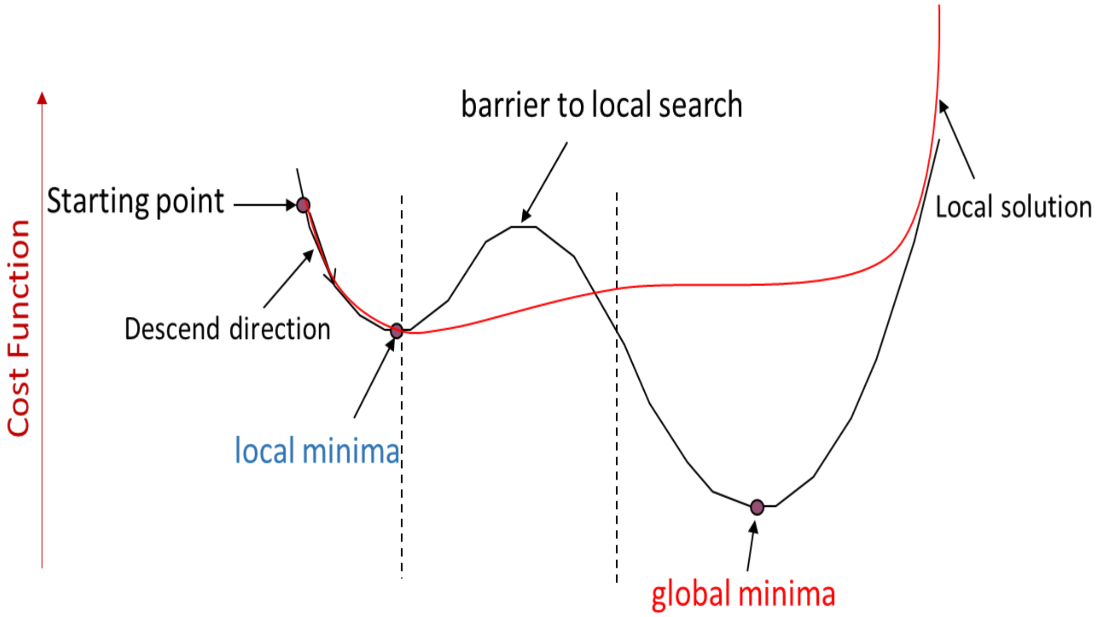


Figure C1: Diagram: Ball on terrain example: Illustrates how global optimization method (SA) overcomes local minima.

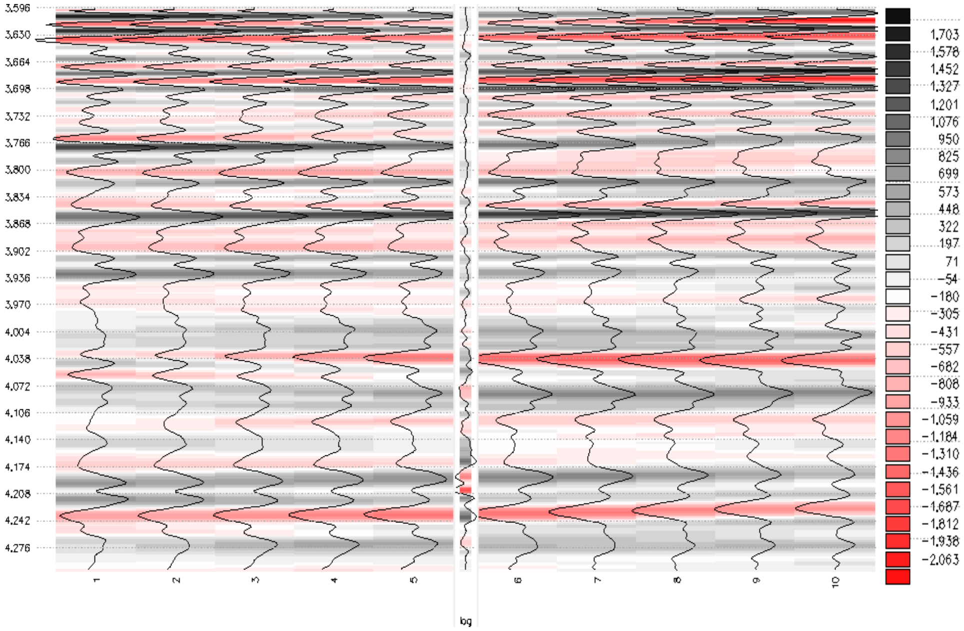


Figure C2: Relative impedance with selected traces.

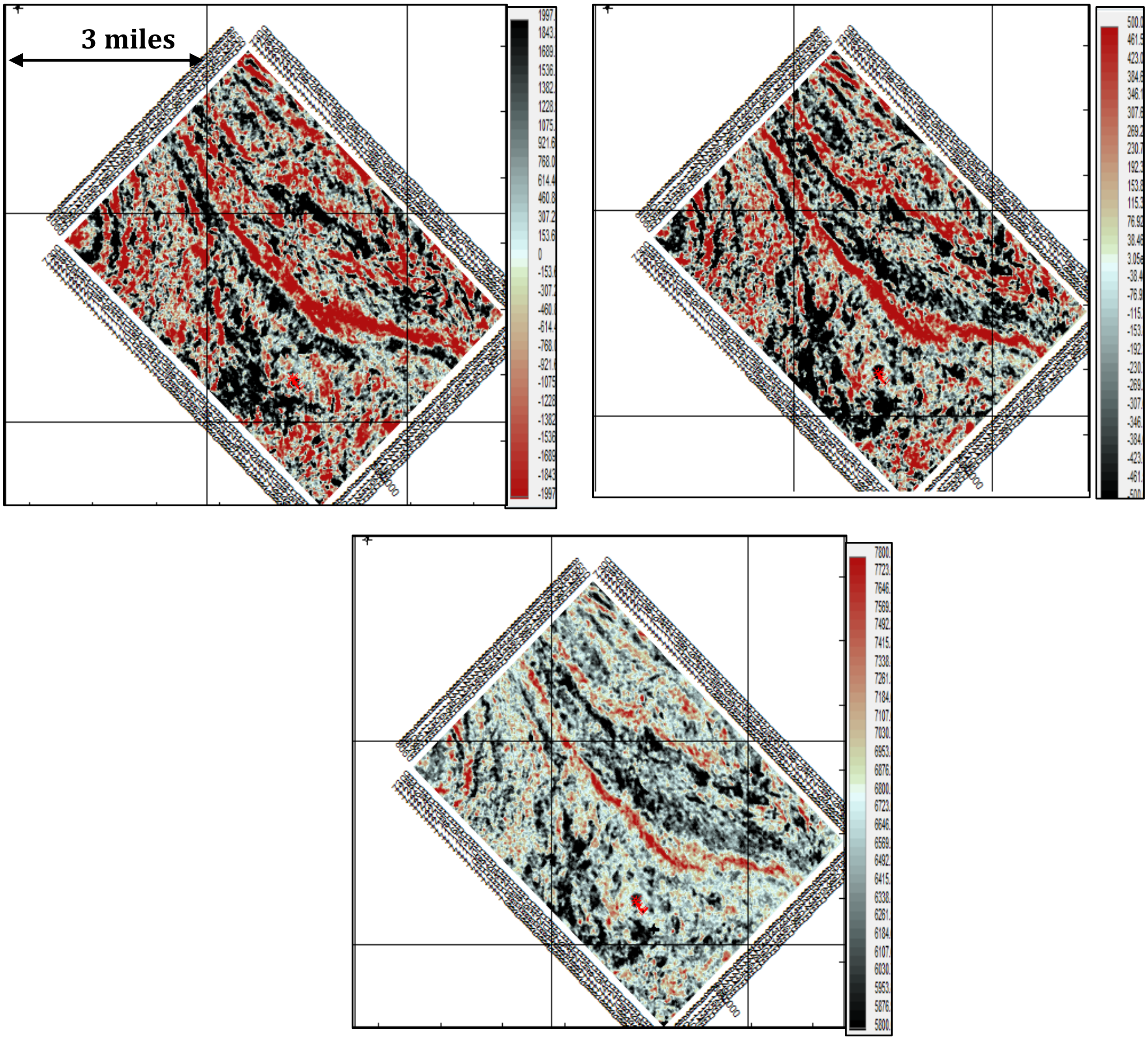


Figure C3: Time-slice at 4.0secs for (a) Original seismic data. (b) Relative impedance. (c) Absolute impedance. Better stratigraphic delineation and improved reservoir connectivity can be seen in (b) and (c)

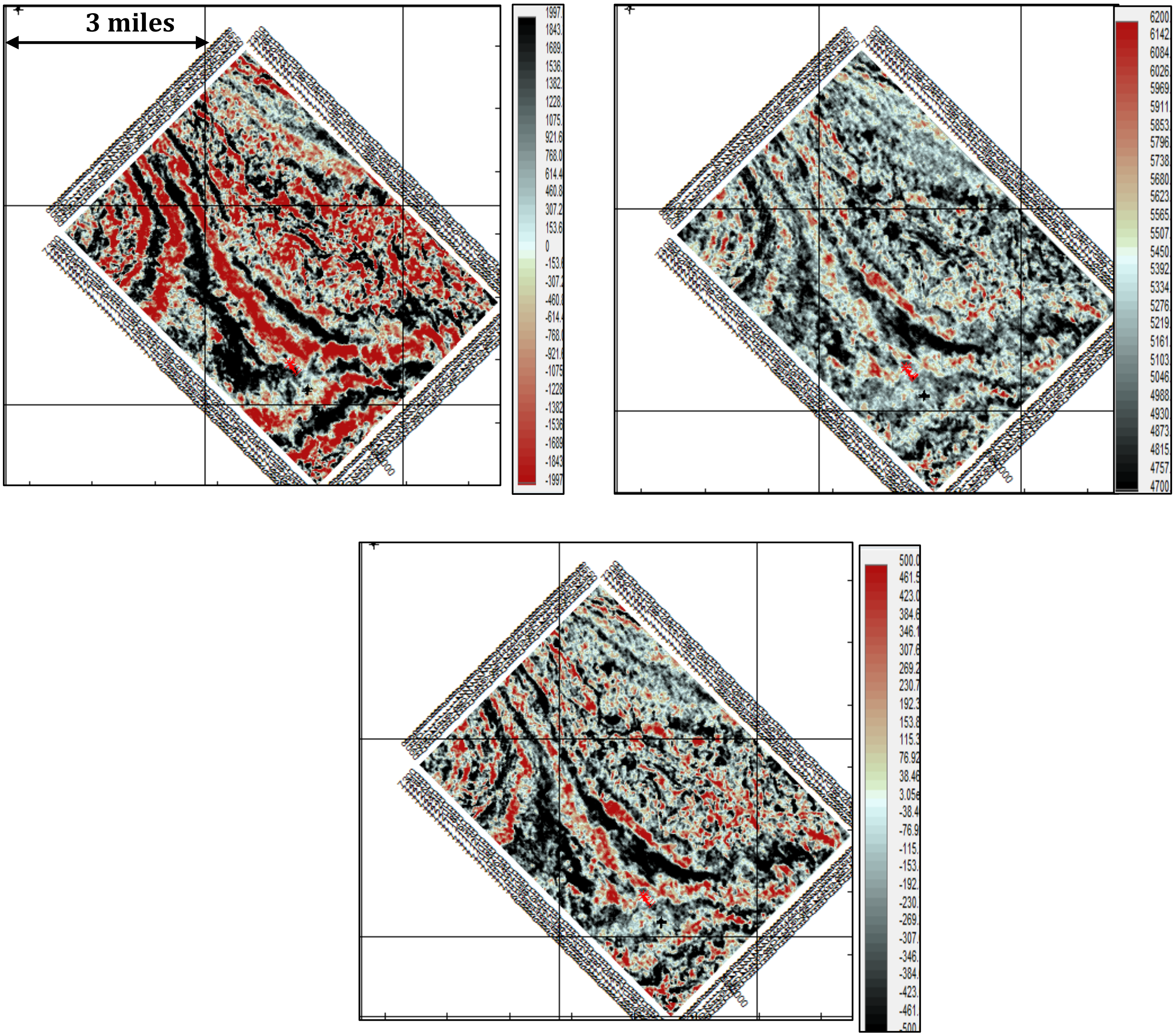
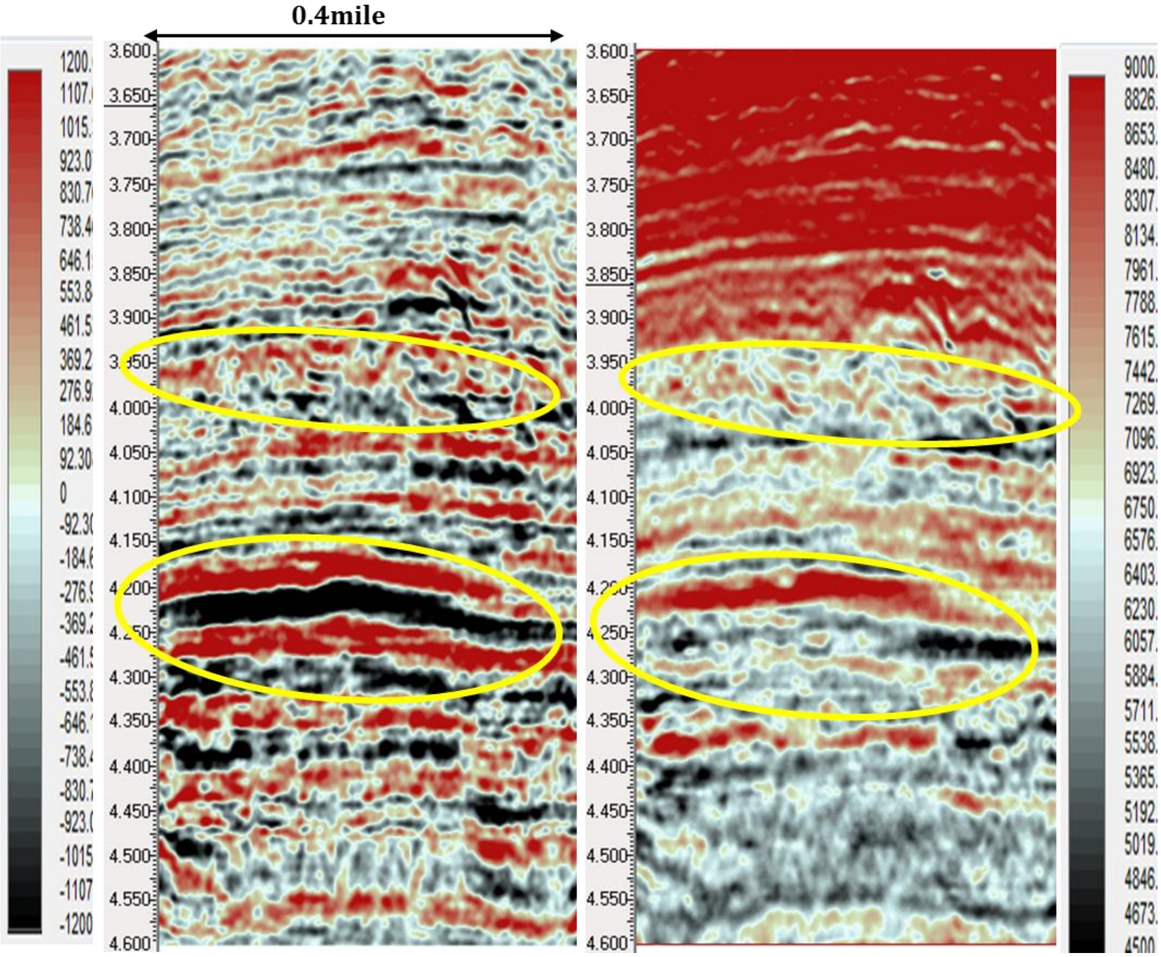


Figure C4: Time-slice at 3.8secs for (a) Original seismic data. (b) Relative impedance. (c) Absolute impedance. Better stratigraphic delineation and improved reservoir connectivity can be seen in (b) and (c)



C5: Detailed comparison of relative and absolute impedance.

Appendix D: Other results from BPI method

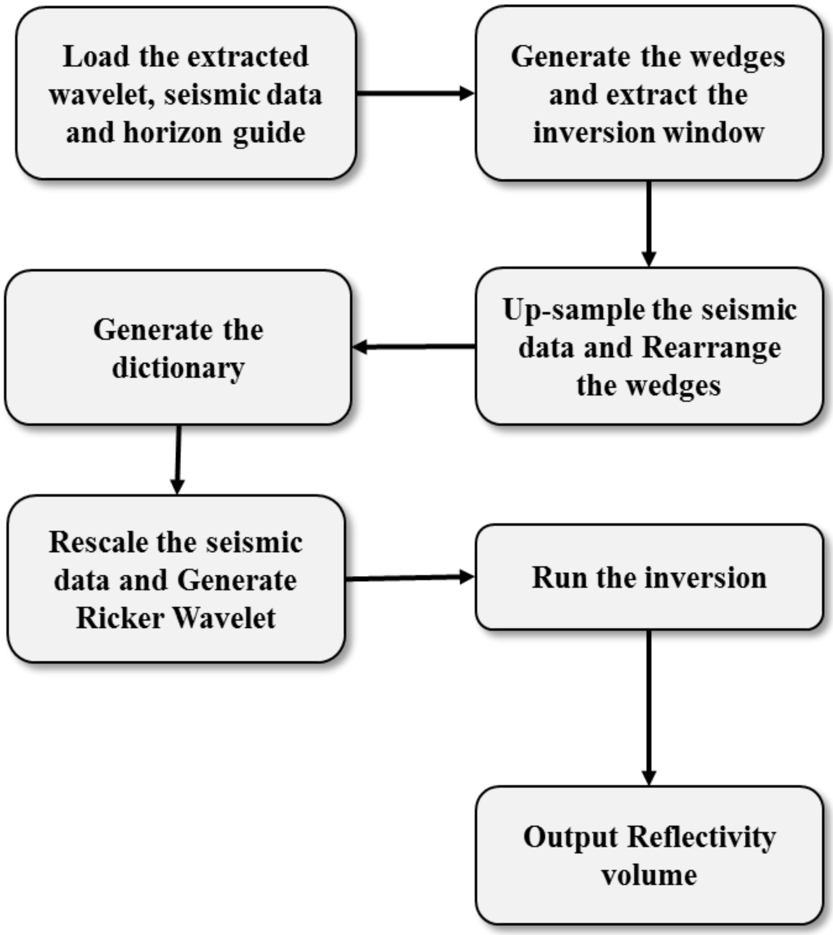

Load the extracted wavelet, seismic data and horizon guide
Generate the wedges and extract the inversion window
Up-sample the seismic data and Rearrange the wedges
Generate the dictionary
Rescale the seismic data and Generate Ricker Wavelet
Run the inversion
Output Reflectivity volume

Figure D1: Workflow for Basis Pursuit Inversion

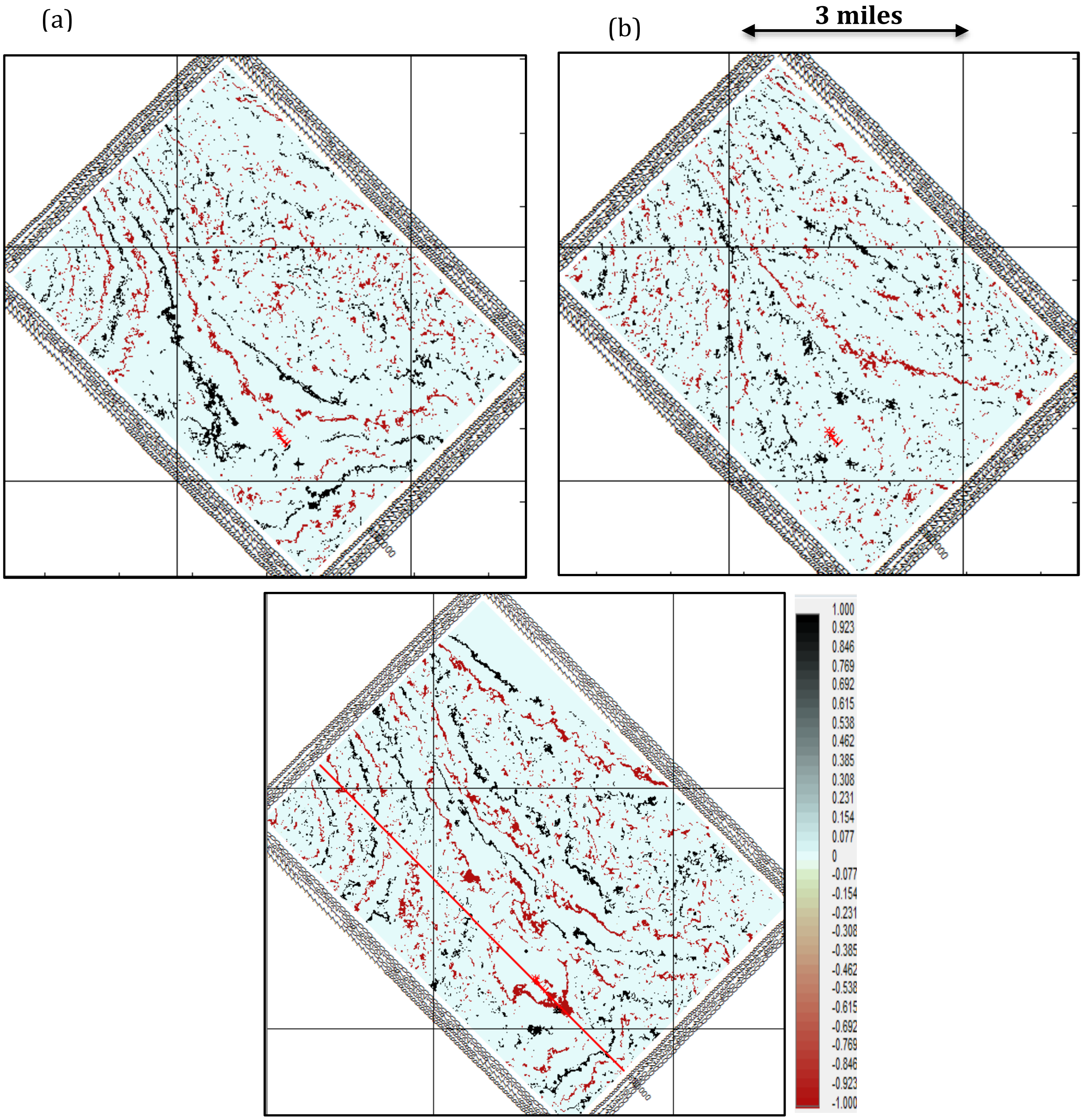


Figure D2: Reflectivity volume time-slices. (a) Time-slice at 3.8secs. (b) Time-slice at 4.0secs. (c) Time-slice at 4.2secs.

Appendix E: Relevant Maps

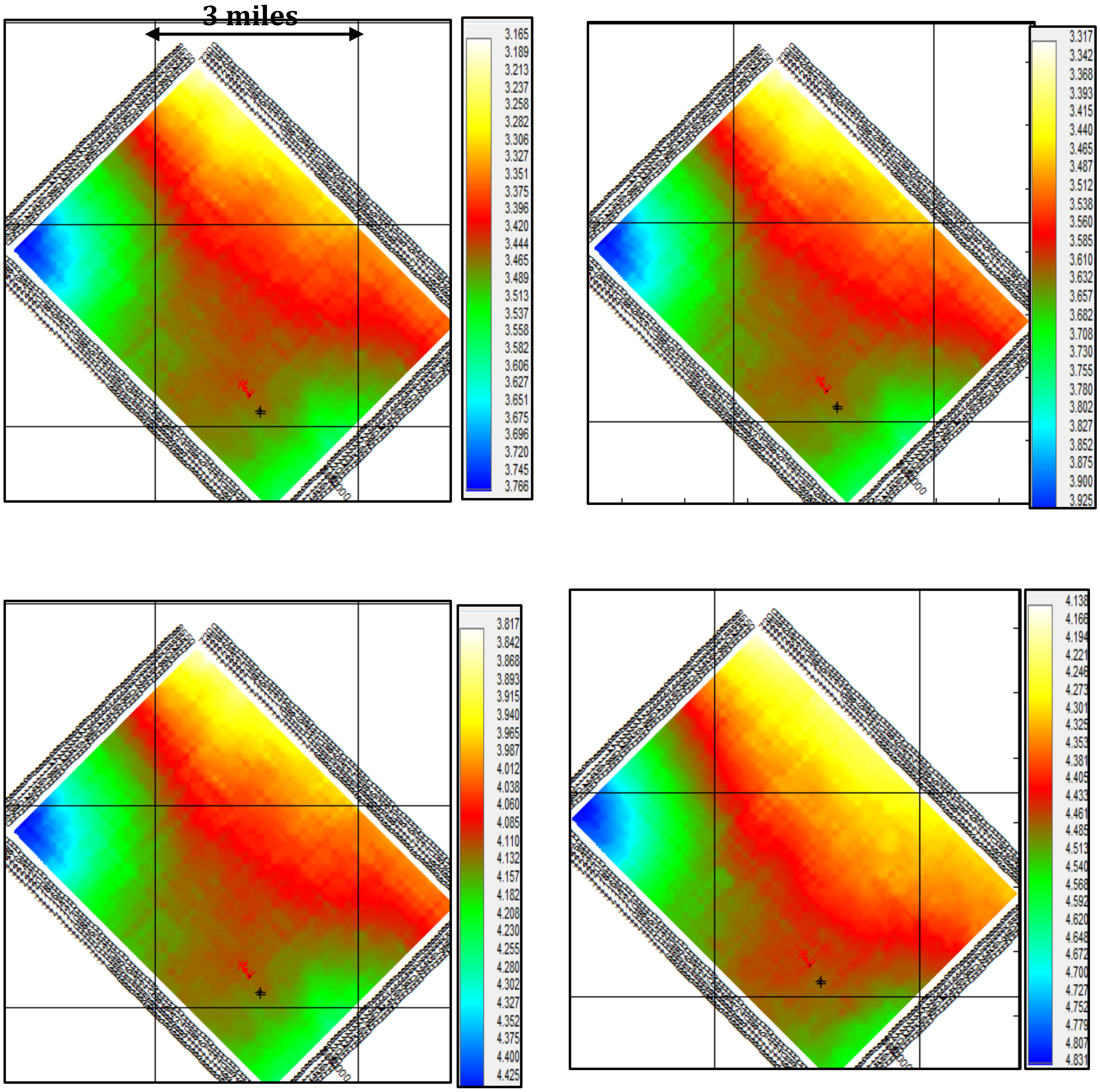


Figure E1: Time Grid horizons with the intervals of interest

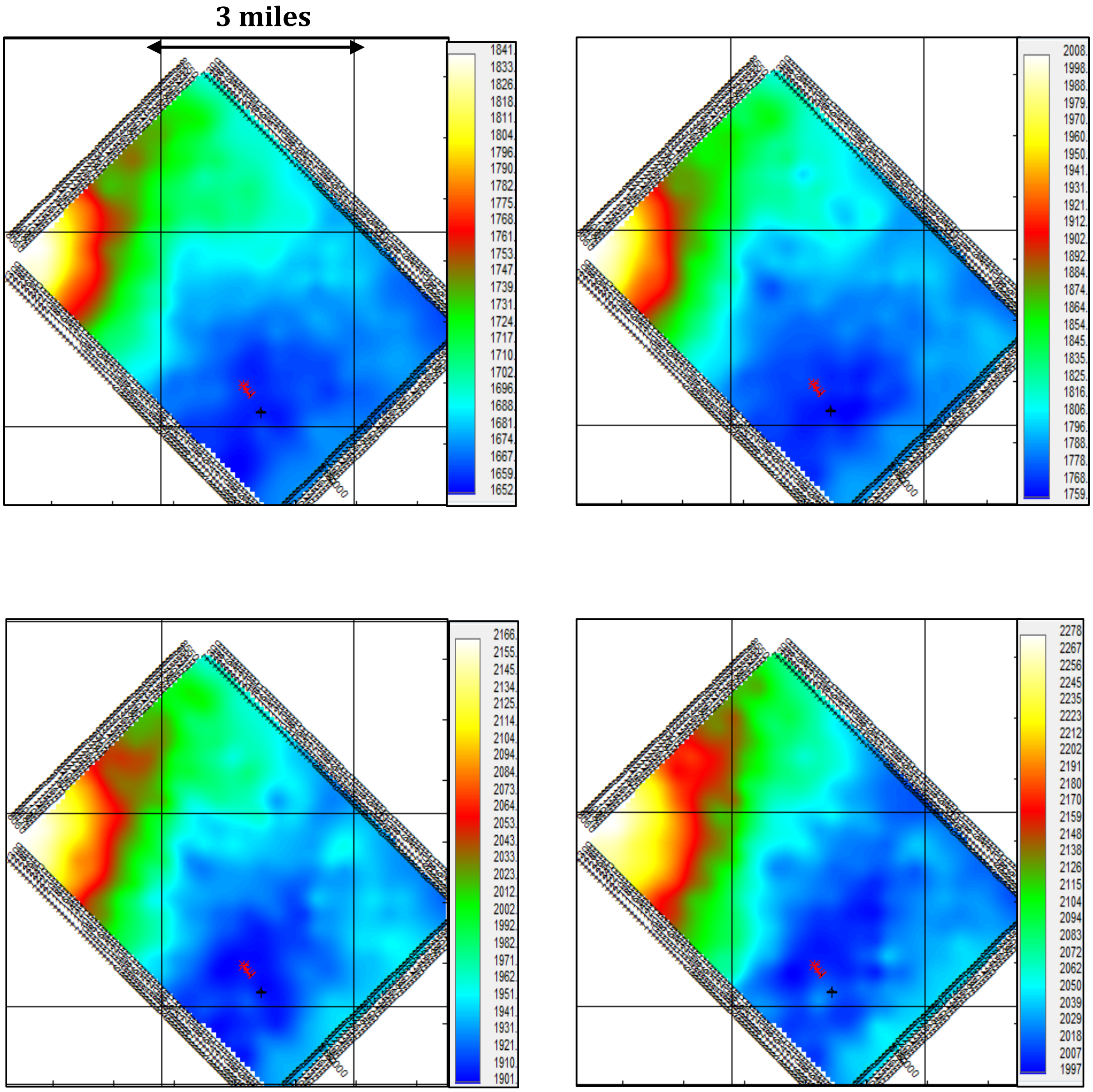


Figure E2: Velocity maps generated from stacking velocity data.

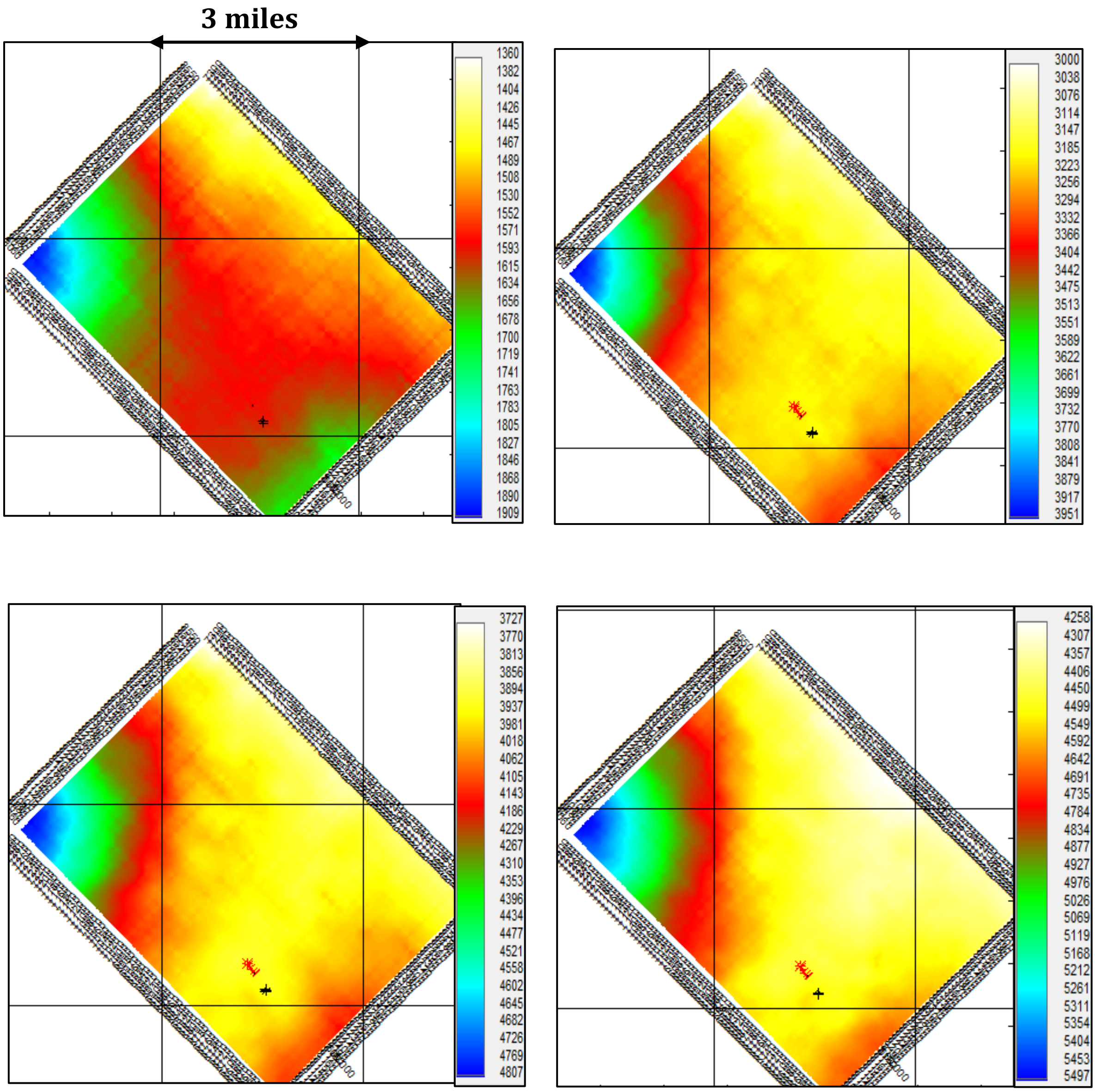


Figure E3: Major depth maps in the areas of interest.

Appendix F: Matlab Codes for Basis Pursuit Inversion

```
% Appendix F
% BP Inversion for Seismic line. ProcessingLineUpsampling.m

clc;
clear all;
close all;

Directory.Input = 'C:\Users\eaadedej\Desktop\ThesisData\';
Directory.Output = 'C:\Users\eaadedej\Desktop\ThesisData\';

Regularization = 2.4;
Inversion.Delta.T = 0.002;
Inversion.Window = 1000;

Wedge.Thickness.Seconds.Min = 0.002;
Wedge.Thickness.Seconds.Max = 0.08;

Limit.Y = [3.6 4.6];

%% Importing seismic wavelet data
disp('Loading wavelet');
tic;
Input.Wavelet.Filename = 'Well_VK_956_A8_ST02BP00_Wavelet_3.6-4.3Secs.txt';
A = importdata([Directory.Input Input.Wavelet.Filename],' ',4);
Input.Wavelet.Time = A.data(:,1);
Input.Wavelet.Amplitude = A.data(:,2);
Input.Wavelet.Delta.T = roundn(mode(diff(A.data(:,1))),-3);
clear A;

if(Inversion.Delta.T == Input.Wavelet.Delta.T)
    A = (min(Input.Wavelet.Time):...
        Inversion.Delta.T:max(Input.Wavelet.Time))';
    Input.Wavelet.Amplitude = interp1(Input.Wavelet.Time,...
        Input.Wavelet.Amplitude,A,'pchip');
    Input.Wavelet.Delta.T = Inversion.Delta.T;
    Input.Wavelet.Time = A;
    clear A;
end;
toc

%% Importing seismic data
disp('Loading seismic');
tic;
Input.Seismic.Filename = 'Inline8180_0to10.sgy';

A = ReadSegyHeader([Directory.Input Input.Seismic.Filename]);
```

```
Input.Seismic.Delta.T = A.dt.*10^(-6);
Input.Seismic.Time.Vector = A.time';

if (rem(Inversion.Window, Input.Seismic.Delta.T*1000) ~= 0)
    Inversion.Window = round(Inversion.Window./...
        (Input.Seismic.Delta.T*1000)).*Input.Seismic.Delta.T*1000;
end;
Input.Seismic.Inline = ReadSegyTraceHeaderValue([Directory.Input...
    Input.Seismic.Filename],'key','EnergySourcePoint');
Input.Seismic.Crossline = ReadSegyTraceHeaderValue([Directory.Input...
    Input.Seismic.Filename],'key','cdpTrace');

Input.Seismic.Traces = zeros(numel(Input.Seismic.Time.Vector),...
    numel(Input.Seismic.Inline));
Input.Seismic.Traces = ReadSegy([Directory.Input...
    Input.Seismic.Filename],'traces',1:numel(Input.Seismic.Inline));
clear A B C D E F i;
toc

%% Importing guide horizon
disp('Loading guide horizon');
tic;
Input.Guide.Filename = 'GuidedHorizonUVIG_3.dat';
A = importdata([Directory.Input Input.Guide.Filename]);
Input.Guide.Inline = A(:,1);
Input.Guide.Crossline = A(:,2);
Input.Guide.Time = A(:,3);
Input.Seismic.Time.Guide =  zeros(numel(Input.Seismic.Inline),1);
j = zeros(numel(Input.Seismic.Inline),1);
for i = 1:numel(Input.Seismic.Inline)
    A = find(Input.Guide.Inline == Input.Seismic.Inline(i));
    if(isempty(A) ~= 1)
        B = find(Input.Guide.Crossline(A) == Input.Seismic.Crossline(i));
        if(isempty(B) ~= 1)
            Input.Seismic.Time.Guide(i) = Input.Guide.Time(A(B));
        else
            j(i) = 1;
        end;
    else
        j(i) = 1;
    end;
end;
Input.Seismic.Time.Guide =
round(Input.Seismic.Time.Guide./Input.Seismic.Delta.T).*Input.Seismic.Delta.T;
if(sum(j) ~= 0)
    Input.Seismic.Inline = Input.Seismic.Inline(j == 0);
    Input.Seismic.Crossline = Input.Seismic.Crossline(j == 0);
    Input.Seismic.Time.Guide = Input.Seismic.Time.Guide(j == 0);
    Input.Seismic.Traces = Input.Seismic.Traces(:,j == 0);
```

```
end;
clear A B i j;
toc

%%
disp('Generating wedges');
tic;
Wedge.Offset.Seconds = (Input.Wavelet.Time(end) - Input.Wavelet.Time(1))/2 - 0.04;
Wedge.Offset.Samples = ceil(Wedge.Offset.Seconds/Inversion.Delta.T);

Wedge.Thickness.Samples.Min =...
    ceil(Wedge.Thickness.Seconds.Min./Inversion.Delta.T);
Wedge.Thickness.Seconds.Min = Wedge.Thickness.Samples.Min.*...
    Inversion.Delta.T;

Wedge.Thickness.Samples.Max =...
    ceil(Wedge.Thickness.Seconds.Max./Inversion.Delta.T);
Wedge.Thickness.Seconds.Max = Wedge.Thickness.Samples.Max*...
    Inversion.Delta.T;

Wedge.Time.Samples.NumberOf = 2*Wedge.Offset.Samples +...
    Wedge.Thickness.Samples.Max;
Wedge.Time.Vector = (1:Wedge.Time.Samples.NumberOf)'.*...
    Inversion.Delta.T - Wedge.Offset.Seconds;

Wedge.Thickness.Samples.NumberOf = Wedge.Thickness.Samples.Max -...
    Wedge.Thickness.Samples.Min + 1;
Wedge.Thickness.Vector = Wedge.Thickness.Seconds.Min:...
    Inversion.Delta.T:Wedge.Thickness.Seconds.Max;

Wedge.Even = zeros(Wedge.Time.Samples.NumberOf,...
    Wedge.Thickness.Samples.NumberOf);
Wedge.Odd = zeros(Wedge.Time.Samples.NumberOf,...
    Wedge.Thickness.Samples.NumberOf);
Synthetic.Even = zeros(Wedge.Time.Samples.NumberOf,...
    Wedge.Thickness.Samples.NumberOf);
Synthetic.Odd = zeros(Wedge.Time.Samples.NumberOf,...
    Wedge.Thickness.Samples.NumberOf);
Wedge.Even(Wedge.Offset.Samples,:) = 1;
Wedge.Odd(Wedge.Offset.Samples,:) = 1;
for i = 1:Wedge.Thickness.Samples.NumberOf
    Wedge.Even(Wedge.Offset.Samples + Wedge.Thickness.Samples.Min +...
        (i - 1),i) = 1;
    Wedge.Odd(Wedge.Offset.Samples + Wedge.Thickness.Samples.Min +...
        (i - 1),i) = -1;
    Synthetic.Even(:,i) = conv(Wedge.Even(:,i),...
        Input.Wavelet.Amplitude,'same');
    Synthetic.Odd(:,i) = conv(Wedge.Odd(:,i),...
        Input.Wavelet.Amplitude,'same');
```

```
end;
clear i;
toc

%%
disp('Extracting seimic window');
tic;
A = Input.Seismic.Traces;
B = Input.Seismic.Time.Vector;
C = Inversion.Window./1000;
Input.Seismic.Extracted.Time.Vector = (-C:Input.Seismic.Delta.T:C)';
D = repmat(Input.Seismic.Extracted.Time.Vector,1,numel(Input.Seismic.Inline));
E = repmat(Input.Seismic.Time.Guide',numel(Input.Seismic.Extracted.Time.Vector),1);
F = E + D;
G = round(F./Input.Seismic.Delta.T);
H = zeros(size(G));
for i = 1:numel(Input.Seismic.Inline);
    H(:,i) = A(G(:,i),i);
end;
Input.Seismic.Extracted.Traces = H;
Input.Seismic.Extracted.Time.Indexes = G;
Input.Seismic.Extracted.Time.Samples.NumberOf = numel(Input.Seismic.Extracted.Time.Vector);
clear A B C D E F G H i;
toc

%%
disp('Upsampling seismic data');
tic;
A = min(Input.Seismic.Extracted.Time.Vector);
B = max(Input.Seismic.Extracted.Time.Vector);
Input.Seismic.Upsample.Time.Vector = (A:Inversion.Delta.T:B)';
Input.Seismic.Upsample.Traces = zeros(numel(...
    Input.Seismic.Upsample.Time.Vector), numel(Input.Seismic.Inline));
for i = 1:numel(Input.Seismic.Inline)
    Input.Seismic.Upsample.Traces(:,i) =...
        interp1(Input.Seismic.Extracted.Time.Vector,...
        Input.Seismic.Extracted.Traces(:,i),...
        Input.Seismic.Upsample.Time.Vector, 'pchip');
end;
Input.Seismic.Upsample.Time.Samples.NumberOf =...
    numel(Input.Seismic.Upsample.Time.Vector);
clear A B i;
toc

%%
disp('Rearranging Wedges');
tic;
A = size(Wedge.Even);
B = zeros(A(1),2*A(2));
```

```
C = zeros(A(1),2*A(2));
for i = 1:A(2)
    B(:,1 + (i - 1)*2) = Wedge.Even(:,i);
    B(:,2*i) = Wedge.Odd(:,i);
    C(:,1 + (i - 1)*2) = Synthetic.Even(:,i);
    C(:,2*i) = Synthetic.Odd(:,i);
end
Wedge.Combined = B;
Synthetic.Combined = C;
clear A B C D i;
toc

%%
disp('Generating dictionary');
tic;
A = Input.Seismic.Upsample.Time.Samples.NumberOf;
B = size(Wedge.Combined);
C = A + B(1) - 1;
D = A + 2*Wedge.Offset.Samples;
E = zeros(C,B(2)*A);
F = zeros(C,B(2)*A);

for i = 1:A
    E(i:(i + B(1) - 1),((i - 1)*B(2)+ 1):(i*B(2))) = Wedge.Combined;
    F(i:(i + B(1) - 1),((i - 1)*B(2)+ 1):(i*B(2))) = Synthetic.Combined;
end;
G = sum(E((A + Wedge.Offset.Samples + 1):end,:));
Wedge.Dictionary = E(1:D,G == 0);
Synthetic.Dictionary = F(1:D,G == 0);
clear A B C D E F G i;
toc

%%
disp('Rescaling seismic');
tic;
A = size(Input.Seismic.Upsample.Traces);
B = size(Wedge.Dictionary);
C = zeros(B(1),A(2));
C((Wedge.Offset.Samples + 1):(Wedge.Offset.Samples + A(1)),:) =...
    Input.Seismic.Upsample.Traces;
Input.Seismic.Upsample.Extended = C;
D = max(abs(C));
Input.Seismic.Upsample.Scaling = D;
C = C./(ones(B(1),1)*D);
Input.Seismic.Upsample.Scaled = C;
Input.Seismic.Upsample.Time.Extended = ((1:B(1)) - 1)'*Inversion.Delta.T +...
    Input.Seismic.Upsample.Time.Vector(1) - Wedge.Offset.Seconds;
clear A B C D;
toc
```

```
%%
disp('Generating Ricker wavelet');
tic;
Wavelet.Ricker.Frequency = 50;
Wavelet.Ricker.Time.Length.Seconds = 0.2;
Wavelet.Ricker.Time.Length.Samples = Wavelet.Ricker.Time.Length.Seconds./...
    Input.Seismic.Delta.T;
Wavelet.Ricker.Time.Shift.Samples = round(Wavelet.Ricker.Time.Length.Samples/2);
Wavelet.Ricker.Time.Vector = ((1:Wavelet.Ricker.Time.Length.Samples) -...
    Wavelet.Ricker.Time.Shift.Samples).*...
    Input.Seismic.Delta.T;
A = - pi^2.*Wavelet.Ricker.Frequency.^2.*Wavelet.Ricker.Time.Vector.^2;
Wavelet.Ricker.Amplitude = (1 + 2.*A).*exp(A);
clear A;
toc

%%
disp('Running inversion');
tic;
A = size(Input.Seismic.Upsample.Scaled);
B = size(Synthetic.Dictionary);
Inversion.Coefficients = zeros(B(2),A(2));
Inversion.Trace = zeros(A);
Inversion.Spike = zeros(A);
Inversion.Time = zeros(A(2),1);
Inversion.Convolved = zeros(A);
for i = 1:A(2)
    Solved = 0;
    C = Regularization;
    while(Solved == 0)
        [D, E] = as_bpdn(Synthetic.Dictionary...
            ,Input.Seismic.Upsample.Scaled(:,i),C);
        if(E.stat == 0)||(E.stat == 6);
            Solved = 1;
        else
            C = C*2;
        end;
    end;
    Inversion.Coefficients(:,i) = D;
    Inversion.Time(i) = E.time;
    Inversion.Spike(:,i) = Wedge.Dictionary*D*Input.Seismic.Upsample.Scaling(i);
    Inversion.Trace(:,i) = Synthetic.Dictionary*D*Input.Seismic.Upsample.Scaling(i);
    clear D E;
    F = circshift(conv(Inversion.Spike(:,i),Wavelet.Ricker.Amplitude), -...
        Wavelet.Ricker.Time.Shift.Samples + 1);
    Inversion.Convolved(:,i) = F(1:A(1));
    disp([num2str(A(2) - i) '   ' num2str(Inversion.Time(i)) '   ' num2str(C)]);
    clear C F;
```

```
end;
clear i;
toc

%%
A = max(Input.Seismic.Time.Guide) + max(Input.Seismic.Upsample.Time.Extended) +0.1;
B = roundn((Inversion.Delta.T:Inversion.Delta.T:A),-3)';
C = zeros(numel(B),numel(Input.Seismic.Inline));
F = zeros(numel(B),numel(Input.Seismic.Inline));
G = zeros(numel(B),numel(Input.Seismic.Inline));
for i = 1:numel(Input.Seismic.Inline);
   D = roundn(Input.Seismic.Upsample.Time.Extended(1) + Input.Seismic.Time.Guide(i),-3);
   E = find(B == D);
   C(E:(E + numel(Input.Seismic.Upsample.Time.Extended) - 1),i) = Inversion.Spike(:,i);
   F(E:(E + numel(Input.Seismic.Upsample.Time.Extended) - 1),i) = Inversion.Trace(:,i);
   G(E:(E + numel(Input.Seismic.Upsample.Time.Extended) - 1),i) = Inversion.Convolved(:,i);
end;
Inversion.Export.Time = B;
Inversion.Export.Spike = C;
Inversion.Export.Trace = F;
Inversion.Export.Convolved = G;
clear A B C D E F G i;

%%
k = 1;
%%
figure('Visible','on');
set(gcf,'Position',[100 100 1600 1200]);
imagesc(1:numel(Input.Seismic.Inline),...
   Input.Seismic.Time.Vector,...
   Input.Seismic.Traces);
ylabel('Time(s)');
title('Input Seismic');
colormap(flipud(jet));
colorbar;
ylim(Limit.Y);
set(gca,'CLim',10000*[-1 1]);
set(gcf,'PaperPositionMode','auto');
print('-djpeg','-r300',[Directory.Output 'Figure' num2str(k)]);
k = k + 1;
close all;

%%
figure('Visible','on');
set(gcf,'Position',[100 100 1600 1200]);
imagesc(1:numel(Input.Seismic.Inline),...
   Inversion.Export.Time,...
   Inversion.Export.Spike);
ylabel('Time(s)');
```

```
xlabel('Crossline');
title(['Spike Inversion with ' num2str(Regularization) ' regularization']);
colormap(flipud(jet));
colorbar;
ylim(Limit.Y);
set(gca,'CLim',2500*[-1 1]);
set(gcf,'PaperPositionMode','auto');
print('-djpeg','-r300',[Directory.Output 'Figure' num2str(k)]);
k = k + 1;
close all;

%%
figure('Visible','on');
set(gcf,'Position',[100 100 1600 1200]);
imagesc(1:numel(Input.Seismic.Inline),...
    Inversion.Export.Time,...
    Inversion.Export.Convolved);
ylabel('Time(s)');
title(['Spike Inversion + Ricker ' num2str(Wavelet.Ricker.Frequency) ' Hz with '
num2str(Regularization) ' regularization']);
colormap(flipud(jet));
colorbar;
ylim(Limit.Y);
set(gca,'CLim',10000*[-1 1]);
set(gcf,'PaperPositionMode','auto');
print('-djpeg','-r300',[Directory.Output 'Figure' num2str(k)]);
k = k + 1;
close all;

%%
WriteSegy([Directory.Output 'InversionSpikeVolume.sgy'],...
    Inversion.Export.Spike,'dt',Inversion.Delta.T,...
    'Inline3D',Input.Seismic.Inline,...
    'Crossline3D',Input.Seismic.Crossline);
WriteSegy([Directory.Output 'InversionTraceVolume.sgy'],...
    Inversion.Export.Trace,'dt',Inversion.Delta.T,...
    'Inline3D',Input.Seismic.Inline,...
    'Crossline3D',Input.Seismic.Crossline);
WriteSegy([Directory.Output 'InversionRickerVolume.sgy'],...
    Inversion.Export.Convolved,'dt',Inversion.Delta.T,...
    'Inline3D',Input.Seismic.Inline,...
    'Crossline3D',Input.Seismic.Crossline);

save([Directory.Output 'SeismicTestInverted' num2str(Regularization) '.mat'],'-v7.3');
```

```
% Appendix F
% BP Inversion for Seismic Volume. ProcessingVolumeUpsampling.m

clc;
clear all;
close all;

Directory.Input = 'C:\Users\eaadedej\Desktop\ThesisData\';
Directory.Output = 'C:\Users\eaadedej\Desktop\ThesisData\';

Regularization = 2;
Inversion.Delta.T = 0.002;
Inversion.Window.Up = 560;
Inversion.Window.Down = 260;

Wedge.Thickness.Seconds.Min = 0.002;
Wedge.Thickness.Seconds.Max = 0.08;

%% Importing seismic wavelet data
disp('Loading wavelet');
tic;
Input.Wavelet.Filename = 'Well_VK_956_A8_ST02BP00_Wavelet_3.6-4.3Secs.txt';
A = importdata([Directory.Input Input.Wavelet.Filename],' ',4);
Input.Wavelet.Time = A.data(:,1);
Input.Wavelet.Amplitude = A.data(:,2);
Input.Wavelet.Delta.T = roundn(mode(diff(A.data(:,1))),-3);
clear A;

if(Inversion.Delta.T == Input.Wavelet.Delta.T)
    A = (min(Input.Wavelet.Time):...
        Inversion.Delta.T:max(Input.Wavelet.Time))';
    Input.Wavelet.Amplitude = interp1(Input.Wavelet.Time,...
        Input.Wavelet.Amplitude,A,'pchip');
    Input.Wavelet.Delta.T = Inversion.Delta.T;
    Input.Wavelet.Time = A;
    clear A;
end;
toc

%% Importing seismic data
disp('Loading seismic');
tic;
Input.Seismic.Filename = 'InversionSmallTest2.sgy';

A = ReadSegyHeader([Directory.Input Input.Seismic.Filename]);
Input.Seismic.Delta.T = A.dt.*10^(-6);
Input.Seismic.Time.Vector = A.time';

if (rem(Inversion.Window.Up, Input.Seismic.Delta.T*1000) ~= 0)
```

```
    Inversion.Window.Up = round(Inversion.Window.Up./...
        (Input.Seismic.Delta.T*1000)).*Input.Seismic.Delta.T*1000;
end;
if (rem(Inversion.Window.Down, Input.Seismic.Delta.T*1000) ~= 0)
    Inversion.Window.Down = round(Inversion.Window.Down./...
        (Input.Seismic.Delta.T*1000)).*Input.Seismic.Delta.T*1000;
end;
Input.Seismic.Inline = ReadSegyTraceHeaderValue([Directory.Input...
    Input.Seismic.Filename],'key','EnergySourcePoint');
Input.Seismic.Crossline = ReadSegyTraceHeaderValue([Directory.Input...
    Input.Seismic.Filename],'key','cdpTrace');

Input.Seismic.Traces = zeros(numel(Input.Seismic.Time.Vector),...
    numel(Input.Seismic.Inline));
Input.Seismic.Traces = ReadSegy([Directory.Input...
    Input.Seismic.Filename],'traces',1:numel(Input.Seismic.Inline));
clear A B C D E F i;
toc

%% Importing guide horizon
disp('Loading guide horizon');
tic;
Input.Guide.Filename = 'GuidedHorizonUVIG_3_2.dat';
A = importdata([Directory.Input Input.Guide.Filename]);
Input.Guide.Inline = A(:,1);
Input.Guide.Crossline = A(:,2);
Input.Guide.Time = A(:,3);
k = zeros(numel(Input.Guide.Inline),1);
for i = 1:numel(Input.Guide.Inline)
    A = find(Input.Seismic.Inline == Input.Guide.Inline(i));
    if(isempty(A) ~= 1)
        B = find(Input.Seismic.Crossline(A) == Input.Guide.Crossline(i));
        if(isempty(B) ~= 1)
            k(i) = A(B);
        end;
    end;
end;
Input.Seismic.Inline = Input.Seismic.Inline(k(k ~= 0));
Input.Seismic.Crossline = Input.Seismic.Crossline(k(k ~= 0));
Input.Seismic.Traces = Input.Seismic.Traces(:,k(k ~= 0));
Input.Guide.Inline = Input.Guide.Inline(k ~= 0);
Input.Guide.Crossline = Input.Guide.Crossline(k ~= 0);
Input.Guide.Time = Input.Guide.Time(k ~= 0);
Input.Guide.Time = round(Input.Guide.Time./Input.Seismic.Delta.T).*Input.Seismic.Delta.T;
clear A B i j;
toc

%%
disp('Generating wedges');
```

```
tic;
Wedge.Offset.Seconds = (Input.Wavelet.Time(end) - Input.Wavelet.Time(1))/2 - 0.04;
Wedge.Offset.Samples = ceil(Wedge.Offset.Seconds/Inversion.Delta.T);

Wedge.Thickness.Samples.Min =...
   ceil(Wedge.Thickness.Seconds.Min./Inversion.Delta.T);
Wedge.Thickness.Seconds.Min = Wedge.Thickness.Samples.Min.*...
   Inversion.Delta.T;

Wedge.Thickness.Samples.Max =...
   ceil(Wedge.Thickness.Seconds.Max./Inversion.Delta.T);
Wedge.Thickness.Seconds.Max = Wedge.Thickness.Samples.Max*...
   Inversion.Delta.T;

Wedge.Time.Samples.NumberOf = 2*Wedge.Offset.Samples +...
   Wedge.Thickness.Samples.Max;
Wedge.Time.Vector = (1:Wedge.Time.Samples.NumberOf)'.*...
   Inversion.Delta.T - Wedge.Offset.Seconds;

Wedge.Thickness.Samples.NumberOf = Wedge.Thickness.Samples.Max -...
   Wedge.Thickness.Samples.Min + 1;
Wedge.Thickness.Vector = Wedge.Thickness.Seconds.Min:...
   Inversion.Delta.T:Wedge.Thickness.Seconds.Max;

Wedge.Even = zeros(Wedge.Time.Samples.NumberOf,...
   Wedge.Thickness.Samples.NumberOf);
Wedge.Odd = zeros(Wedge.Time.Samples.NumberOf,...
   Wedge.Thickness.Samples.NumberOf);
Synthetic.Even = zeros(Wedge.Time.Samples.NumberOf,...
   Wedge.Thickness.Samples.NumberOf);
Synthetic.Odd = zeros(Wedge.Time.Samples.NumberOf,...
   Wedge.Thickness.Samples.NumberOf);
Wedge.Even(Wedge.Offset.Samples,:) = 1;
Wedge.Odd(Wedge.Offset.Samples,:) = 1;
for i = 1:Wedge.Thickness.Samples.NumberOf
   Wedge.Even(Wedge.Offset.Samples + Wedge.Thickness.Samples.Min +...
      (i - 1),i) = 1;
   Wedge.Odd(Wedge.Offset.Samples + Wedge.Thickness.Samples.Min +...
      (i - 1),i) = -1;
   Synthetic.Even(:,i) = conv(Wedge.Even(:,i),...
      Input.Wavelet.Amplitude,'same');
   Synthetic.Odd(:,i) = conv(Wedge.Odd(:,i),...
      Input.Wavelet.Amplitude,'same');
end;
clear i;
toc

%%
disp('Extracting seimic window');
```

```
tic;
A = Input.Seismic.Traces;
B = Input.Seismic.Time.Vector;
C = Inversion.Window.Up/1000;
D = Inversion.Window.Down/1000;
Input.Seismic.Extracted.Time.Vector = (-C:Input.Seismic.Delta.T:D)';
D = repmat(Input.Seismic.Extracted.Time.Vector,1,numel(Input.Seismic.Inline));
E = repmat(Input.Guide.Time',numel(Input.Seismic.Extracted.Time.Vector),1);
F = E + D;
G = floor(F./Input.Seismic.Delta.T);
H = zeros(size(G));
for i = 1:numel(Input.Seismic.Inline);
    H(:,i) = A(G(:,i),i);
end;
Input.Seismic.Extracted.Traces = H;
Input.Seismic.Extracted.Time.Indexes = G;
Input.Seismic.Extracted.Time.Samples.NumberOf = numel(Input.Seismic.Extracted.Time.Vector);
clear A B C D E F G H i;
toc

%%
disp('Upsampling seismic data');
tic;
A = min(Input.Seismic.Extracted.Time.Vector);
B = max(Input.Seismic.Extracted.Time.Vector);
Input.Seismic.Upsample.Time.Vector = (A:Inversion.Delta.T:B)';
Input.Seismic.Upsample.Traces = zeros(numel(...
    Input.Seismic.Upsample.Time.Vector), numel(Input.Seismic.Inline));
for i = 1:numel(Input.Seismic.Inline)
    Input.Seismic.Upsample.Traces(:,i) =...
        interp1(Input.Seismic.Extracted.Time.Vector,...
        Input.Seismic.Extracted.Traces(:,i),...
        Input.Seismic.Upsample.Time.Vector, 'pchip');
end;
Input.Seismic.Upsample.Time.Samples.NumberOf =...
    numel(Input.Seismic.Upsample.Time.Vector);
clear A B i;
toc

%%
disp('Rearranging Wedges');
tic;
A = size(Wedge.Even);
B = zeros(A(1),2*A(2));
C = zeros(A(1),2*A(2));
for i = 1:A(2)
    B(:,1 + (i - 1)*2) = Wedge.Even(:,i);
    B(:,2*i) = Wedge.Odd(:,i);
    C(:,1 + (i - 1)*2) = Synthetic.Even(:,i);
    C(:,2*i) = Synthetic.Odd(:,i);
end
Wedge.Combined = B;
Synthetic.Combined = C;
clear A B C D i;
toc

%%
disp('Generating dictionary');
tic;
A = Input.Seismic.Upsample.Time.Samples.NumberOf;
B = size(Wedge.Combined);
C = A + B(1) - 1;
D = A + 2*Wedge.Offset.Samples;
E = zeros(C,B(2)*A);
F = zeros(C,B(2)*A);

for i = 1:A
    E(i:(i + B(1) - 1),((i - 1)*B(2)+ 1):(i*B(2))) = Wedge.Combined;
    F(i:(i + B(1) - 1),((i - 1)*B(2)+ 1):(i*B(2))) = Synthetic.Combined;
end;
G = sum(E((A + Wedge.Offset.Samples + 1):end,:));
Wedge.Dictionary = E(1:D,G == 0);
Synthetic.Dictionary = F(1:D,G == 0);
clear A B C D E F G i;
toc

%%
disp('Rescaling seismic');
tic;
A = size(Input.Seismic.Upsample.Traces);
B = size(Wedge.Dictionary);
C = zeros(B(1),A(2));
C((Wedge.Offset.Samples + 1):(Wedge.Offset.Samples + A(1)),:) =...
    Input.Seismic.Upsample.Traces;
Input.Seismic.Upsample.Extended = C;
D = max(abs(C));
Input.Seismic.Upsample.Scaling = D;
C = C./(ones(B(1),1)*D);
Input.Seismic.Upsample.Scaled = C;
Input.Seismic.Upsample.Time.Extended = ((1:B(1)) - 1)'*Inversion.Delta.T +...
    Input.Seismic.Upsample.Time.Vector(1) - Wedge.Offset.Seconds;
clear A B C D;
toc

%%
disp('Generating Ricker wavelet');
tic;
Wavelet.Ricker.Frequency = 50;
Wavelet.Ricker.Time.Length.Seconds = 0.2;
Wavelet.Ricker.Time.Length.Samples = Wavelet.Ricker.Time.Length.Seconds./...
    Input.Seismic.Delta.T;
Wavelet.Ricker.Time.Shift.Samples = round(Wavelet.Ricker.Time.Length.Samples/2);
Wavelet.Ricker.Time.Vector = ((1:Wavelet.Ricker.Time.Length.Samples) -...
    Wavelet.Ricker.Time.Shift.Samples).*...
    Input.Seismic.Delta.T;
A = - pi^2.*Wavelet.Ricker.Frequency.^2.*Wavelet.Ricker.Time.Vector.^2;
Wavelet.Ricker.Amplitude = (1 + 2.*A).*exp(A);
clear A;
toc

%%
disp('Running inversion');
tic;
A = size(Input.Seismic.Upsample.Scaled);
B = size(Synthetic.Dictionary);
Inversion.Coefficients = zeros(B(2),A(2));
Inversion.Trace = zeros(A);
Inversion.Spike = zeros(A);
Inversion.Time = zeros(A(2),1);
Inversion.Convolved = zeros(A);
for i = 1:A(2)
    Solved = 0;
    C = Regularization;
    while(Solved == 0)
        [D, E] = as_bpdn(Synthetic.Dictionary...
            ,Input.Seismic.Upsample.Scaled(:,i),C);
        if(E.stat == 0)||(E.stat == 6);
            Solved = 1;
        else
            C = C*2;
        end;
    end;
    Inversion.Coefficients(:,i) = D;
    Inversion.Time(i) = E.time;
    Inversion.Spike(:,i) = Wedge.Dictionary*D*Input.Seismic.Upsample.Scaling(i);
    Inversion.Trace(:,i) = Synthetic.Dictionary*D*Input.Seismic.Upsample.Scaling(i);
    clear D E;
    F = circshift(conv(Inversion.Spike(:,i),Wavelet.Ricker.Amplitude), -...
        Wavelet.Ricker.Time.Shift.Samples + 1);
    Inversion.Convolved(:,i) = F(1:A(1));
    disp([num2str(A(2) - i) '   ' num2str(Inversion.Time(i)) '   ' num2str(C)]);
    clear C F;
    if(rem(i,50000) == 0)
        save([Directory.Output 'MC13-Q_Volume2_' num2str(Regularization) '.mat'],'-v7.3');
    end;
end;
clear i;
```

```
toc

%%
A = max(Input.Guide.Time) + max(Input.Seismic.Upsample.Time.Extended) +0.1;
B = roundn((Inversion.Delta.T:Inversion.Delta.T:A),-3)';
C = zeros(numel(B),numel(Input.Seismic.Inline));
F = zeros(numel(B),numel(Input.Seismic.Inline));
G = zeros(numel(B),numel(Input.Seismic.Inline));
for i = 1:numel(Input.Seismic.Inline);
    D = roundn(Input.Seismic.Upsample.Time.Extended(1) + Input.Guide.Time(i),-3);
    E = find(B == D);
    C(E:(E + numel(Input.Seismic.Upsample.Time.Extended) - 1),i) = Inversion.Spike(:,i);
    F(E:(E + numel(Input.Seismic.Upsample.Time.Extended) - 1),i) = Inversion.Trace(:,i);
    G(E:(E + numel(Input.Seismic.Upsample.Time.Extended) - 1),i) = Inversion.Convolved(:,i);
end;
Inversion.Export.Time = B;
Inversion.Export.Spike = C;
Inversion.Export.Trace = F;
Inversion.Export.Convolved = G;
clear A B C D E F G i;

%%
WriteSegy([Directory.Output 'MC13-Q_InversionSpikeVolume2_' num2str(Regularization) '.sgy'],...
    Inversion.Export.Spike,'dt',Inversion.Delta.T,...
    'Inline3D',Input.Seismic.Inline,...
    'Crossline3D',Input.Seismic.Crossline);
WriteSegy([Directory.Output 'MC13-Q_InversionTraceVolume2_' num2str(Regularization) '.sgy'],...
    Inversion.Export.Trace,'dt',Inversion.Delta.T,...
                  'Inline3D',Input.Seismic.Inline,...
    'Crossline3D',Input.Seismic.Crossline);
WriteSegy([Directory.Output 'MC13-Q_InversionRickerVolume2_' num2str(Regularization)
'.sgy'],...
    Inversion.Export.Convolved,'dt',Inversion.Delta.T,...
    'Inline3D',Input.Seismic.Inline,...
    'Crossline3D',Input.Seismic.Crossline);

save([Directory.Output 'MC13-Q_Volume2_' num2str(Regularization) '.mat'],'-v7.3');
```

Vita

Elijah was born in Ibadan, an ancient and largest city in Nigeria. He completed his high school education from Wesley college of Science, Ibadan, Oyo-state and proceeded to Federal University of Technology Akure, Ondo-state, where he obtained a Bachelor's degree in Applied Physics with Electronics in 2008. He joined the Shell Nigeria Exploration and Production Company (SNEPCO) in 2009 as an intern under the mandatory 12 months youth scheme program in Nigeria. He continued to work as a geoscience technician for the Geophysics team. During this role, he developed a strong passion for seismic data processing and earth modeling. As a consequence, he joined the University of New Orleans graduate program to pursue a Master's degree in Geophysics from the department of Applied Physics. He was part of the team that won the 2015 AAPG Imperial Barrel Award competition, Gulf Coast section for University of New Orleans.