P-318

Application of Gauss-Newton tomographic inversion to seawater velocity estimation

Uma Vadapalli *, *CSIR*

Summary

Any seismic survey consists of large number of profiles. Often different profiles are shot on different dates. In case of 3D marine seismic survey it is often that even adjacent profiles are shot on different dates, which can be days, weeks or even months. The two adjacent sail-lines can receive reflected rays from the same subsurface reflection point; this implies that the same reflection point will be recorded on different dates. The propagation velocity of seismic waves in sea water is a function of physical properties of sea water, especially temperature and salinity. These can vary seasonally, so can the seismic velocity in water. When two different sail-lines receive reflected rays from the same subsurface reflection point and the water velocity changes during the time gap between shooting these lines, reflection times will be affected by water velocity variations. But during seismic data processing all the traces corresponding to the same reflection point are processed with same water velocity, which is not valid in this case. This means the effects of water velocity variations are not taken into effect. If the water bottom reflection times of one line data are inverted to find velocity during the acquisition of that line and this velocity is used to generate final seismic image then the effects of water velocity variations could be removed. So in this study estimation of water velocity for each sail-line considering water-bottom reflections as an input to the Gauss-Newton tomography method is discussed.

Introduction

The propagation velocity of seismic waves in sea water can vary over time. According to a study (Scott et al., 2003) temperature variation of 10C can cause sea water velocity variation of 3m/s. If in the time interval between shooting different lines the water velocity has changed, then the recorded times of the same reflection point will be different. If a CMP gather is considered, since it is 3D case, it comprises seismic traces from different lines. Figure 1 shows how data comes from same reflection point to multiple sail lines. Figure 2 shows a small offset range (800-1000 m) of a CMP gather comprising data from three sail lines, trace color representing sail lines the data came from and peaks representing water bottom reflection. Reflection times are different for the same offset depending on which sail line it came from. This difference in travel time is proportional to water depth, in shallower water areas the time shift is negligible but in deeper water areas (6000ft to 10,000ft, after; Barley, 1999) it will be considerable. This dependence of reflection time on sail line is typical of water-velocity variation in the time interval when these data were acquired. If the CMP gather is stacked without correcting these time shifts, stack response will be poor and blurred. In 3D migration generally all the traces of a CMP gather are migrated (pre- stack migration) with single velocity, these causes over migration of some traces and under migration of others because each trace shot on different days will have different velocity. If water velocity specific for each sail- line is known, each trace could be migrated with its own velocity. Even 2D velocity analysis which is based on NMO correction can also provide velocity. But in case of complexly structured sea bottom, the move-out will not be hyperbolic in the CMP domain; this makes the velocity analysis uncertain.

Room No. 104, Main building, NGRI (CSIR), Uppal Road, Hyderabad.
Email: umageo.anm4@gmail.com

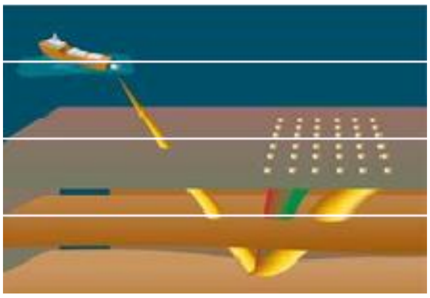


Figure1. Recording of data in multiple lines from same reflection point.

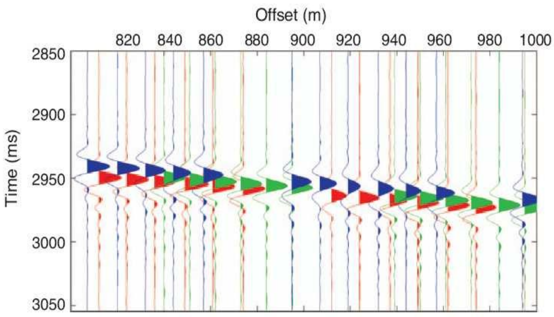


Figure2. Common-midpoint gather comprising data from three sail lines. Each trace color representing the sail line the data came from (After: Gerson Luis da Silva Ritter, 2010).

Theory

Tomographic inversion

The method presented here is based on tomographic inversion of seismic data. Inversion theory deals with estimating model parameters from the observed data. Tomography is travel time inversion, i.e. travel time is used as data to estimate model parameters like velocity, depth etc. For numerical implementation of tomography algorithm, the model space is divided into homogeneous cells (in 2-D) or blocks (in 3-D) (Berryman, 1990). It is convenient to develop tomography formulae in terms of wave slowness models, instead of velocity because pertinent equations are linear in slowness. Slowness is defined as reciprocal of the velocity. Then the travel time can be written as the product of path length and slowness.

t = LS... (1)

Where t is matrix of travel times, L and S are matrices of path length and slowness values in each cell. This is the basic equation to solve for linear inverse modeling. Here t is data matrix, **S** is model vector and **L** is geometry matrix. Suppose each cell of model space is assigned a value say s_j denoting the slowness value of the j^{th} cell or block let l_{ij} be the length of the i^{th} ray path through the j^{th} cell. Figure 3 shows a cell model of slowness. Here the ray is falling only in few numbers of cells only those cells will have l_{ij} values for other cells those values will be zero.

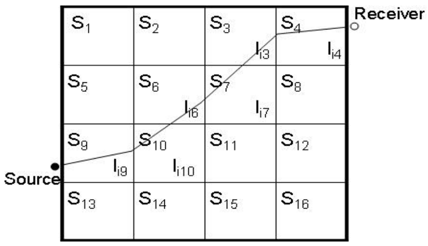


Figure3. Schematic illustration of ray paths through

Gauss – Newton algorithm

The Gauss – Newton algorithm is a method used to solve non-linear least-squares problems. This method uses iterative procedure to find minimum of a function. We follow the notations given by Aster et al. (2005) to describe the mathematical details of the method.

Consider a non-linear inversion problem in which the model parameters and the data are related through a non-linear system of equations G (m) =d. Where d is data (observed values), G (m) model's predictions (calculated values for assumed model). Goal is to find values of the model parameters that best fit the data in the sense of minimizing the residuals using L_2-norm. Consider a squared error function for 'm' number of measurements

$$f(m)=\sum_{i=1}^{m}\left(\frac{G(m)_i-d_i}{\sigma_i}\right)^2 \; i=1,2,\ldots,m \ldots\ldots\ldots\ldots\ldots(2)$$

As Gauss – Newton method finds minimum of a function so it can minimize the error function f(m), so it will be used to fit model to some data by minimizing the sum of squares of error between the data and model's predictions. For convenience let

$$f_i(m) = \frac{G(m)_i - d_i}{\sigma_i} \quad i = 1,2,\ldots,m \quad \text{and}$$

$$F(m) = \left[f_1(m)\ f_2(m)\ldots f_m(m)\right]^T$$

F(m) is matrix of residuals. Let us suppose that m*is approximately equal to true model. The necessary condition for m^* to be minimum of f (m) is that gradient of f (m*) is equal to zero i.e.

$$\nabla f(m^*) = 0.$$

When the gradient of function with respect to some value is zero then that value will give either local minimum value or local maximum value of the function. But in this case m^* will give only minimum of the function f(m) because m^* is approximately equal to true value and f(m) is residual function. Let m^0 be the initial assumed model, near to m^* and the difference between m^* and m^0 is Δm i.e.

$$\Delta m = m^* - m^\circ$$

$$\nabla f(m^\circ + \Delta m) = 0 \quad \text{Since} \quad \nabla f(m^*) = 0$$

After Taylor's series expansion and setting the approximate gradient equal to zero gives

$$\nabla^2 f(m^\circ)\Delta m = -\nabla f(m^\circ)\ldots\ldots\ldots\ldots\ldots\ldots(3)$$

$\nabla f(m)$ is gradient of f(m) and $\nabla^2 f(m)$ is Hessian of f(m). These can be expressed as follows

$$\nabla f(m^0) = 2J(m^0)^T F(m^0)\ldots\ldots\ldots\ldots\ldots\ldots(4)$$

$$\nabla^2 f(m^0) = 2J(m^0)^T J(m^0)\ldots\ldots\ldots\ldots\ldots\ldots(5)$$

(For derivation part refer to Parameter Estimation and Inverse Problems by Richard C. Aster et al, 2005)

Where J(m) is Jacobian of the function f(m), it is a matrix consisting of first partial derivatives of f(m) with respect to model parameters. In Gauss – Newton method Hessian is expressed in terms of Jacobian. This avoids the complexity of calculating second derivative.

$$J(m) = \begin{bmatrix} \frac{\partial f_1(m)}{\partial m_1} & \frac{\partial f_1(m)}{\partial m_2} & \cdots & \frac{\partial f_1(m)}{\partial m_n} \\ \frac{\partial f_2(m)}{\partial m_1} & \frac{\partial f_2(m)}{\partial m_2} & \cdots & \frac{\partial f_2(m)}{\partial m_n} \\ \vdots & & & \\ \frac{\partial f_m(m)}{\partial m_1} & \frac{\partial f_m(m)}{\partial m_2} & \cdots & \frac{\partial f_m(m)}{\partial m_n} \end{bmatrix}$$

Jacobian consists of as many number of rows as number of data points and as many number of columns as the number of model parameters. Substituting equations (4) and (5) in equation (3), it gives

$$J(m^0)^T J(m^0)\Delta m = -J(m^0)^T F(m^0)\ldots\ldots\ldots\ldots\ldots(6)$$

Δm can be calculated from the above equation. This is the first iteration for calculating Δm for an initial assumed model m^0.

In general iterative procedure for Gauss – Newton method can be described as follows

1) Consider an initial assumed model m^0

2) Solve the system for Δm

$$J(m^n)^T J(m^n)\Delta m = -J(m^n)^T F(m^n)\ldots\ldots\ldots\ldots\ldots(7)$$

3) $m^{n+1} = m^n + \Delta m$

4) Go back to step 2 until $\|\Delta m\|_2 < \varepsilon$

'ε' is very small number that controls the end of iterations.

In the above process Δm is found in each iteration by making the gradient of error function (f(m)) equal to zero (i.e. error is minimized) and L_2-norm of Δm which is the discrepancy between true and assumed model is also minimized successively in each iteration. So the calculated model m successively moves towards m* in each iteration and finally reaches m*, where the iterative process stops. Calculating Δm from equation (7) is linear inverse problem because

$$\Delta m = \left(J(m)^T J(m)\right)^{-1} J(m)^T\left(-F(m)\right)\ldots\ldots\ldots\ldots\ldots(8)$$

It is analogous to solution for over determined problem when J (m) is known. This reveals that Gauss – Newton method converts a non-linear problem to linear problem by assuming that initial assumed model is near to true model so that higher order terms of tailor's series expansion can be neglected.

Water velocity estimation by Gauss – Newton tomography:

The aim of this study is to find water velocity specific for each sail line in order to solve the problems caused by water velocity variations when data were acquired on different dates along different lines.

To estimate the true water velocity corresponding to a sail-line, consider the data from that line and use water bottom other reflection events. They could be easily picked and water-bottom reflector is mostly affected by water velocity variations compared to other deeper reflectors. Let the travel time of the seismic wave between the source (q) and receiver (r) reflected at the water bottom be described by the non-linear functional $T_{q,r}$ (m). The model vector m describes the medium where the wave propagates.

$$m = [z(x,y), V_q(x,y)]^T \ldots\ldots\ldots\ldots (9)$$

Where Z(x, y) describes the water-bottom surface and V_q(x, y) describes the water velocity associated with the source q. As usual in tomography water-bottom surface is discretized into 'n' points, to find depth of each point so that whole water bottom topography can be described throughout the profile. But the water column is not divided into cells with respect to velocity; here it is assumed that water velocity is constant along each sail line. This is not too much of simplification because when lateral velocity variations are expected, it is always possible to consider many portions of a sail line and this method can be applied to each portion. As usual in tomography slowness is used instead of velocity. Now the model vector modifies to

$$m = [z_1, z_2, \ldots, z_n, s]^T = [\mathbf{z}, s]^T \ldots\ldots\ldots\ldots (10)$$

There are n+1 number of model parameters to determine. Let us suppose that there are k measured reflection times ($\tau_{qi,\,ri}$, i =1, …, k). The model parameters, velocity and depth have to be estimated from reflection times. This is non-linear problem because velocity as well as depth is not linearly related to reflection times.

In this problem the objective function that has to be minimized is given by equation (11) and Δm has to be minimized by solving equation (12).

$$\phi(m) = \sum_{i=1}^{k} (T_{q_i r_i}(m) - \tau_{q_i r_i})^2 \ldots\ldots\ldots (11)$$

$$J^T(m^n)J(m^n)\Delta m = -J^T(m^n)(T(m^n) - \tau) \ldots\ldots\ldots\ldots\ldots (12)$$

$T_{qi,\,ri}$ are calculated travel times for an assumed model (m) and $\tau_{qi,\,ri}$ are observed travel times. The functional

φ (m)

can be minimized by the Gauss – Newton algorithm by using iterative procedure. To find Δm by solving equation (12) Jacobian is required.

In this problem Jacobian is calculated as

$$J_{i,j}(m) = \frac{\partial T_i(m)}{\partial m_j} \quad i = 1,\ldots,k \quad m_j = z_1,\ldots,z_n,s \ldots\ldots\ldots (13)$$

Since the water-bottom surface is discretized, the partial derivatives $\partial T_i/\partial z_j$ must be computed for all points z_j that influence the modeled time T_i. For a given ray reflection comes from a small zone called Fresnel zone, hence the propagation time would be influenced by only a few points z_j in the vicinity of reflection point. Then only those few points will have $\partial T_i/\partial z_j$ value and for all other points these values will be zero. The resulting Jacobian would be a sparse matrix with large dimensions (k rows and n + 1 column). This problem as many are in reflection tomography would be non-linear and ill-conditioned. A much simpler solution can be obtained by considering that the initial depth model i.e. water bottom surface can be created using previously migrated data (Ritter, 2010). One important assumption is that if the water bottom has small dips and is migrated with an incorrect velocity, it will be displaced in relation to its true position mainly in the vertical direction. This means a constant vertical displacement in the whole depth model throughout a sail line. Suppose Δs is the difference between true and incorrect slowness values it will cause a constant vertical depth shift (Δz) of the water-bottom surface. In this case, there will be only one depth parameter to estimate i.e. Δz instead of n depth locations and the slowness parameter to be determined is Δs. Now the Jacobian will consist of only two columns. The Jacobian to be computed and the variation of the model to be solved are,

$$J(z,s)=\begin{bmatrix} \frac{\partial T_i(z,s)}{\partial z} & \frac{\partial T_i(z,s)}{\partial s} \\ \vdots & \vdots \\ \frac{\partial T_k(z,s)}{\partial z} & \frac{\partial T_k(z,s)}{\partial s} \end{bmatrix} \qquad \Delta m = \begin{bmatrix} \Delta z \\ \Delta s \end{bmatrix}$$

The elements of the Jacobian are given by (Gerson Luis da Silva Ritter, 2010)

$$\frac{\partial T_i(z,s)}{\partial s} = d_i(z) \qquad \frac{\partial T}{\partial z} \cong \frac{\Delta t}{\Delta z} = \frac{2\cos\alpha\cos\theta}{V}$$

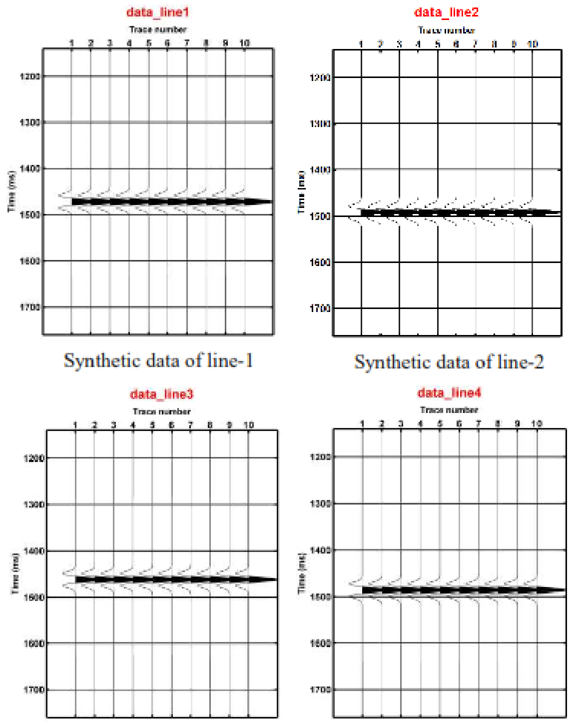
Where α is the angle between the normal to the surface and the vertical direction (i.e. dip), θ is the reflection angle. Once the Jacobian is known Δm can be calculated using equation (12) by following the steps of Gauss – Newton method.

The sequential procedure for this inversion process is as follows. Consider one line data. The first step in the inversion process is to define the initial depth model. For this time-migrate the data, use approximate seismic water velocity to migrate the sea bottom reflection. One could do a crude velocity analysis for this. Then pick the water- bottom reflection times from migrated data and apply constant-velocity vertical-ray time to depth conversion to create the depth model. Second step is choosing initial model velocity. Using these initial model parameters modeled reflections time (T_i(m)) and Jacobian has to be calculated. The third step is picking the water-bottom reflection for each trace this provides $\tau_{qi,\ ri}$. This step is on unmigrated data. Before picking water-bottom reflection times apply phase filter to convert the wavelet to zero phase so that maximum amplitude of the reflection corresponds to the reflection point in the sea bottom.

Repeat the whole process for each sail-line and estimate the velocity of seismic waves in sea water. Use the estimated velocities for pre-stack 3D depth-migration algorithm that takes into account the water velocity associated with each sail-line, this removes time shifts in the data caused by water velocity variations. Then stack the data, good stack response will be obtained.

A Simple Synthetic Example:

The Gauss – Newton method is applied to synthetic data set. To create the synthetic data set with water velocity variations, it is assumed that there is 10^{o}C temperature variation in water during the acquisition of four different sail-lines and with velocities of seismic wave in water as 1.495km/s, 1.475km/s, 1.505km/s, 1.480km/s and the density of sea water as 1.027g/cc for all the sail lines. The water bottom sedimentary layer is assumed to have 2.00km/s velocity and 2.1g/cc density. With these assumed model parameters the reflection coefficient for the water bottom surface corresponding to each sail line is RC_1= 0.4646, RC_2=0.4699, RC_3= 0.4620, RC_4=0.4685. The water bottom surface is assumed to be perfectly horizontal at a depth of 2.2 km. So the real depth model is Z=2.2 km. To condition the Jacobian matrix the units used are kilometers for distance and seconds for time.



Synthetic data of line-1

Synthetic data of line-2

Synthetic data of line-3

Synthetic data of line-4

Figure 4

This way, the numerical values of the columns of the Jacobian are in the same numerical range. It is assumed that each sail line consists of ten traces. Each trace has its own source and receiver, zero offset Normal incidence case is assumed so that convolution model could be used. Ricker wavelet (30Hz) is convolved with reflection coefficients to generate synthetic seismic trace. The picked one way travel time for the water bottom reflector for each sail line is

$$\tau_1 = 1.4720s,\ \tau_2 = 1.4920s,\ \tau_3 = 1.4620s,\ \tau_4 = 1.4860s.$$

All the ten traces of each sail line have same travel time because source and receiver are at same position for each trace and the reflector is horizontal. Figure 4 shows synthetic data created for four sail lines. However the reflector is at same depth beneath all the sail lines, travel times are different for each line because of water velocity variations.

The initial depth model to be used is the same as the real one but vertically displaced by 0.02 km. Now the initial depth model to be used is Z_0=Z+0.02. The initial guessed velocities used for the four lines are

$$V_{i_1} = 1.473 km/s\ V_{i_2} = 1.454 km/s\ V_{i_3} = 1.482\ km/s\ V_{i_4} = 1.456\ km/s$$

Results:

With the chosen initial model parameters, the calculated one-way travel times; the distance traveled by the ray between source and reflection point for four sail lines are

T_1=1.5071s T_2= 1.5268s T_3=1.4979s T_4=1.5247s
d_1=2.2200km d_2=1.2200km d_3=2.2200km d_4=2.2200km.

Using these parameters the Jacobian described in the previous section is computed. The Jacobian matrix J_1, J_2, J_3 and J_4 for the four lines are given below. For all the lines second column of the Jacobian is same because initial depth model is same.

$$J_1 = \begin{bmatrix} 1.3577 & 2.2200 \\ 1.3577 & 2.2200 \\ 1.3577 & 2.2200 \\ 1.3577 & 2.2200 \\ 1.3577 & 2.2200 \\ 1.3577 & 2.2200 \\ 1.3577 & 2.2200 \\ 1.3577 & 2.2200 \\ 1.3577 & 2.2200 \\ 1.3577 & 2.2200 \end{bmatrix} \quad J2 = \begin{bmatrix} 1.3755 & 2.2200 \\ 1.3755 & 2.2200 \\ 1.3755 & 2.2200 \\ 1.3755 & 2.2200 \\ 1.3755 & 2.2200 \\ 1.3755 & 2.2200 \\ 1.3755 & 2.2200 \\ 1.3755 & 2.2200 \\ 1.3755 & 2.2200 \\ 1.3577 & 2.2200 \end{bmatrix}$$

$$J_3 = \begin{bmatrix} 1.3495 & 2.2200 \\ 1.3495 & 2.2200 \\ 1.3495 & 2.2200 \\ 1.3495 & 2.2200 \\ 1.3495 & 2.2200 \\ 1.3495 & 2.2200 \\ 1.3495 & 2.2200 \\ 1.3495 & 2.2200 \\ 1.3495 & 2.2200 \\ 1.3495 & 2.2200 \end{bmatrix} \quad J_4 = \begin{bmatrix} 1.3736 & 2.2200 \\ 1.3736 & 2.2200 \\ 1.3736 & 2.2200 \\ 1.3736 & 2.2200 \\ 1.3736 & 2.2200 \\ 1.3736 & 2.2200 \\ 1.3736 & 2.2200 \\ 1.3736 & 2.2200 \\ 1.3736 & 2.2200 \\ 1.3736 & 2.2200 \end{bmatrix}$$

With above Jacobians system of equations (12) is solved for Δm, which gives following Δz and Δs values.

Δz_1=-0.007043 Δz_2=-0.007022 Δz_3=-0.007193 Δz_4=-0.007805
Δs_1=-0.011515 Δs_2=-0.011334 Δs_3=-0.011832 Δs_4=-0.012614

These values are used for next iteration. After few iterations (number of iterations depends on how close the initial guess is to the true value), the iterative process has stopped and produced final Δz and Δs values. The corresponding estimated velocities and depths for each line are also given below.

Δz_1=0.0000003134 Δz_2=0.0000003445
Δz_3=0.0000003054 Δz_4=0.0000003771
Δs_1= 0.0000005158 Δs_2= 0.0000005595
Δs_3= 0.0000005058 Δs_4=0.0000006137
V_1=1.4953km/s V_2=1.4753km/s
V_3=1.5052km/s V_4=1.4798km/s
Z_1=2.2011km Z_2=2.2012km Z_3=2.2007km Z_4=2.1991km

Figure 5 is a plot between estimated and true velocities (sail lines are placed at 500m interval). Error plots are given in figure6. Data misfit versus iteration graphs are drawn for each line. It is obvious that Data misfit is decreasing with iteration for all the lines.

In order to stop the iterative process appropriate ε value is required. To select the ε value one could follow this procedure. Any iterative process stops when the values of last two iterations are equal. For example consider two velocities 1.485km/s and 1.485002km/se which are approximately equal, then corresponding Δs=0.000001. Similarly Δz=0.000001 is considered. Then these values are used in Δm and 2-

norm of Δm is calculated. This has given ε =0.000001. This value is used to control the end of iterations.

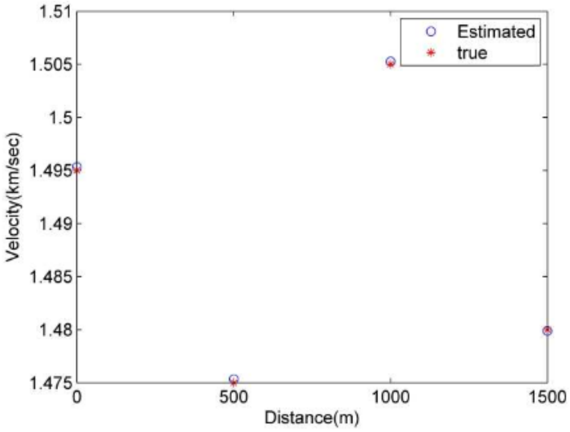


Figure5. Plot between Estimated and True velocities

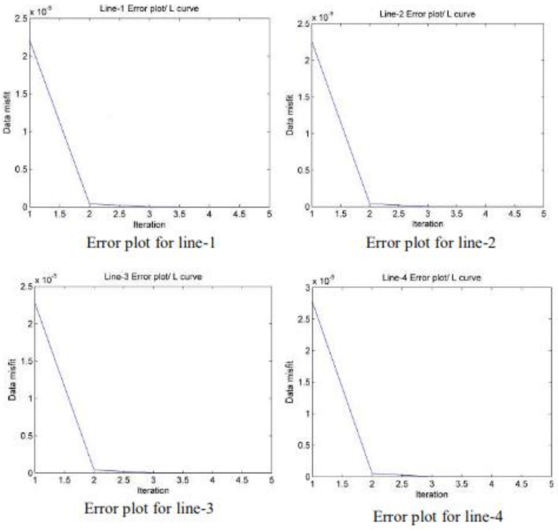


Figure6. Error plots for Data misfit

Conclusion

The whole exercise is focused on Gauss _ Newton inversion method. The technique presented in this study is used on synthetic data. Gauss – Newton method has minimized the discrepancies between modeled and measured travel times. So that two model parameters velocity and depth to the water bottom are obtained. The estimated model parameters are matching with the true model parameters used to create synthetic data set when the initial guessed model is near to the true model. Gauss – Newton inversion method, being Local Optimization method is converging to true model when the initial guessed velocity and the initial depth model are not too far from the true model parameters. In case of real data sets always it may not be possible to choose such an initial guess. So in my feature work I will focus on Global Optimization methods which will converge to global optimum. This has been a good learning exercise for me to implement the mathematical concept of gradient based inversion algorithm.

Acknowledgements

I am thankful to the Director, NGRI (CSIR) for permission to present this paper. My advisors Dr.V.P.Dimri and Dr. Ravi Srivastava made helpful suggestions and always encouraged me in improving this exercise. Thanks to Dr. Kirti Srivastava and Nimisha Vedanti for their suggestions and useful discussion.

Thanks to Parveen begum for her help in manuscript drafting. The author acknowledges the financial support from Royal Norwegian Embassy (New Delhi) to carry out this work.

References

Gerson Luis da Silva Ritter, 2010, Water velocity estimation using inversion methods: Geophysics, **75**, U_1- U_8.

Aster, R.C., B.Borches, and C.H. Thurber, 2005, Parameter estimation and inverse problems: Elsevier Science Publishers, Amsterdam.

Berryman, J.G., 1990, Lecture notes on Nonlinear and Tomography: I. Borehole seismic Tomography.

Scott MacKay, Jonathan Fried, Charless Carvill, 2003, The impact of water-velocity variations on deep water seismic data: TLE, April, 344-350.

Brain Barley, 1999, Deep-water problems around the world: TLE, April, 488-494.