Petroleum Technology Development Journal

Seismic Refraction Data Analyses: A Software for Low Velocity Layer Interpretation in the Niger Delta

By

G. I. Alaminiokuma* and T. Warmate**

Abstract
A software has been developed to carry out a near-surface seismic refraction data analyses using Visual Basic.net programming language. This software is based mainly on the principles of seismic refraction technique. The analyses include automatically plotting time-offset graphs employing the Least-squares approximation to fit the best lines, computing velocities from the reciprocals of the slopes of the lines and computing thicknesses employing the standard seismic refraction equations. The program was debugged and test-run on a Microsoft Windows 8 Operating System (OS) using a standard set of low velocity layer (LVL) data from nine seismic lines across the east-central Niger Delta. The results show that the weathered layer has a total thickness ranging between 12.20 m and 34.64 m; first weathered layer velocities ranging between 227.8 ms^{-1}, and 250 ms^{-1}, sub-weathered layer velocities between 428 ms^{-1} and 530 ms^{-1} and consolidated layer velocities between 1646.4 ms^{-1} and 1737 ms^{-1}. These results compare reasonably with those of other researchers in this same part of the Niger Delta. This remarkable closeness is an indication that the software is a reliable means for the fast, efficient and less strenuous analysis of near-surface seismic refraction data. It is an all-important tool needed by seismologist, geotechnologists and hydrogeologists.

Key words: Seismic Refraction, Low Velocity Layer, Visual Basic.net, Least-Squares Approximation and Software

INTRODUCTION

Before the advent of modern computers and software to analyze oil and gas data, seismic refraction data interpretation was conducted manually. This was characterized by time wastage, strenuous manual computational and iterative routines involved in obtaining numerical solutions, unnecessary repetitions involved in making modifications, lack of updating of previous works in the light of experience, and elusive results due to various simple blunders (errors) arising from manually plotting graphs: choosing line of best-fit, scaling of co-ordinates and tedious calculations.

Presently, various software for near-surface low velocity layer analysis are in use by most multi-national oil companies. Such software include the FACE static computer program used to process and interpret LVL properties[11], OYO Geospace Corp. uphole data analysis Software version 1.01b used to process and interpret LVL refraction data[2] and UDISYS used to interpret uphole data[3] among others. These expensive software are classified and unpublished.

---

*Department of Earth Sciences, Federal University of Petroleum Resources Effurun, P.M.B. 1221, Effurun, Nigeria.
**Geostrat International Services Ltd., 14 Manilla Pepple Street, D-Line, Port Harcourt, Nigeria.

[1] Eze, C. L., Okwueze, E. E., and. Uko, E. D. (2003). The velocity–thickness characteristics of the mangrove swamp low velocity layer (LVL) South central Niger Delta, Nigeria. Global Journal of pure and applied sciences, 9(3), 369-374.

[2] UGNL/IDSL JV (2005). Uphole Acquisition Report of Oguali 3D Seismic Survey, OML 16/22. pp 1-18.

[3] CNPC/BGP International Nigeria Limited. (2010). Determine the Optimum Shot Configuration (Charge Size and Depth) in OML 31. Pp 1-11

However, few attempts have been made by researchers in Nigeria to develop these packages for use. A computer program (NESURVELANA) was developed to analyze near-surface velocity seismic data employing the uphole method[4]. Similarly, a computer model (FAJSEIS) was also developed to compute low velocity layer correction parameters obtained by the uphole method[5]. These programs were developed mainly to interpret the uphole/downhole method which gives the direct vertical velocity and thickness profile in the area around the borehole. Results obtained by these researchers employing the uphole/downhole method were not laterally extensive therefore, needs to be complemented by the seismic refraction method which has a wide subsurface coverage in order to properly delineate hidden layers, blind layers, dipping beds and velocity reversals where they exist.

This work is another progressive effort to develop such software and stir up interest in geophysical software development in Nigeria. It is an automated technique to analyze a 2-way (forward and reverse shooting schemes) seismic refraction data obtained in the Niger Delta. It has been painstakingly developed following the standard processes for software development and considering accuracy and geological factors within the Niger Delta

METHODOLOGY

Field Data Acquisition

The data used to test-run this computer program was acquired by a Low Velocity (LVL) (Refraction) recording spread which consists of 12 SM4, 10Hz geophones laid on the surface but centered about the intersection (Figure 1). Total spread length is 70 m. The complete acquisition consists of 6 shots; 3 forward shots and 3 reverse shots made at surface intervals along clearly determined seismic lines. Shots were recorded at 75 m and 5 m off each end of the spread. To obtain deeper data, two additional shots both 210 m off each spread, were recorded. For each of the shots recorded, holes were drilled to the depth of 1 m and 5 to 10 caps (detonators) were used as the energy source. The detected signals are recorded on an OYO GEOSPACE MCSeis-160MxV5.42, a 24 trace portal digital recorder as 'wiggly' traces from which various first-breaks (first-arrival time) are picked. The first-arrival time, T is plotted against the geophone offset, X. In the case of a subsurface consisting of discrete homogeneous layers, this type of plot consists of linear segments.

Model Equations

Some mathematical equations which proved useful in solving the problem were employed in the program design. These are presented below:

Equations for Plotting the T-X Curves

Generally, the plot is mathematically given as:

$$T_n = \left(\frac{1}{V_i}\right) X_n + T_{ci} \qquad (1)$$

Where: $\left(\frac{1}{V_i}\right)$ = slopes of the various lines.

T_{ci} = Intercepts on the T-axis, i = 0, 1, 2. . . , n = number of data points for a particular line

[4] Alaminiokuma, G. I., Uko, E. D., and Israel-Cookey. C. (2007). Near-surface seismic velocity data: A computer program for analysis, Nigeria Journal of Physics, 19(2), 253-266.

[5] Fajana, A. O., Oladunjoye, M. A. and Olayinka, A. I. (2014). Computation of Low Velocity Layer Parameters by Computer Models. International Journal of Pure and Applied Sciences and Technology, 25(2), 46-62.

For Forward Shooting Scheme (f)

$$T_{f_n} = \left(\frac{1}{V_{f_i}}\right)\left(X_{f_n}\right) + T_{c\,f_n} \tag{2}$$

For Reverse Shooting Scheme (r)

$$T_{r_r} = \left(\frac{1}{V_{r_r}}\right)\left(-X_{r_r}\right) + T_{c\,r_r} \tag{3}$$

Equations for Computing the Slope

The mathematical equations for computing the slope of the various curves are given below:

Forward Shooting Scheme

$$Slope, S_f = \frac{\Delta T_{f_n}}{\Delta X_{f_n}} \ (s/m) \tag{4}$$

Reverse Shooting Scheme

$$Slope, S_r = \frac{\Delta T_{r_n}}{\Delta(-X_{r_n})} \ (s/m) \tag{5}$$

Mathematical Equations for Computing Velocities

The mathematical equations for computing the velocity of the various layers are given below:

Forward Shooting Scheme

$$Velocity, V_{f_i} = \frac{\Delta X_{f_n}}{\Delta T_{f_n}} \ (m/s) \tag{6}$$

Reverse Shooting Scheme

$$Velocity, V_{r_i} = \frac{\Delta(-X_{r_n})}{\Delta(T_{r_n})} \ (m/s) \tag{7}$$

Equations for Depth Computation

The equations for computing the depth as adapted from[6],[7] [6,7] are given below:

Forward Shooting Scheme

$$Depth, Z_{0_f} = \frac{t_{1_f}}{2}\left[\frac{V_{1_f} V_{0_f}}{\sqrt{V_{1_f}^{\ 2} - V_{0_f}^{\ 2}}}\right] + \frac{D_s}{2} \tag{8}$$

---

[6] Knox, W. A., (1967), Multi-layer near surface reflection Computation in Musgrave, A.W. Ed; Seismic reflection prospecting: Soc. of Expl. Geophysics 197 – 216.

[7] Dobrin, M. B. (1983). Introduction to Geophysical Prospecting. Mc Graw-Hill Book Co., London.3rd Ed., Pp. 294 – 300; 307 – 310.

$$Z_{1_f} = \left[\frac{t_{1_f} - 2Z_{0_f}\sqrt{V_{2_f}^{\ 2} - V_{0_f}^{\ 2}}}{V_{2_f}V_{0_f}}\right]\left[\frac{V_{2_f}V_{1f}}{2\sqrt{V_{2_f}^{\ 2} - V_{1_f}^{\ 2}}}\right] \quad (9)$$

Reverse Shooting Scheme

$$Depth, Z_{1_r} = \frac{t_{1r}}{2}\left[\frac{V_{1_r}V_{0_r}}{\sqrt{V_{1_r}^{\ 2} - V_{0_r}^{\ 2}}}\right] + \frac{D_s}{2} \quad (10)$$

$$Z_{1_r} = \left[\frac{t_{1_r} - 2Z_{0_r}\sqrt{V_{2_r}^{\ 2} - V_{0_r}^{\ 2}}}{V_{2_r}V_{0_r}}\right]\left[\frac{V_{2_r}V_{1r}}{2\sqrt{V_{2_r}^{\ 2} - V_{1_r}^{\ 2}}}\right] \quad (11)$$

Least-Square Approximation Equations

The Least-squares approximation equations are adapted from[8] to fit the best lines.

For Lines of Best-fit

These equations are coded into the Visual Basic.net programming language for use by the computer to automatically plot the various LVL curves.

Forward Shooting Scheme:

$$T_{fc_i} = \frac{\left(\sum T_{f_n}\right)\left(\sum X_{f_n}^{\ 2}\right) - \left(\sum X_{f_n}\right)\left(\sum X_{f_n}T_{f_n}\right)}{N\sum\left(X_{f_n}^{\ 2}\right) - \left(\sum\left(X_{f_n}\right)\right)^2} = \text{Intercept Time} \quad (12)$$

$$\frac{1}{V_{f_i}} = \frac{N\left(\sum X_{f_n}T_{f_n}\right) - \left(\sum X_{f_n}\right)\left(\sum T_{f_n}\right)}{N\sum\left(X_{f_n}^{\ 2}\right) - \left(\sum\left(X_{f_n}\right)\right)^2} = \text{Slope} \quad (13)$$

Reverse Shooting Scheme:

$$T_{rc_i} = \frac{\left(\sum T_{r_n}\right)\left(\sum(-X_{r_n}^{\ 2})\right) - \left(\sum - X_{r_n}\right)\left(\sum(-X_{r_n})T_{r_n}\right)}{N\sum\left(-X_{r_n}^{\ 2}\right) - \left(\sum\left(-X_{r_n}\right)\right)^2} = \text{Intercept Time} \quad (14)$$

$$\frac{1}{V_{r_i}} = \frac{N\left(\sum - X_{r_n}T_{f_n}\right) - \left(\sum - X_{r_n}\right)\left(\sum T_{r_n}\right)}{N\sum\left(-X_{r_n}^{\ 2}\right) - \left(\sum\left(-X_{r_n}\right)\right)^2} = \text{Slope} \quad (15)$$

Algorithm Design

Step 1: Input values for T (*ms*) axis and X (*m*) axis respectively.
Step 2: Plot the points scatter for T (*ms*) on the vertical axis against X (*m*) on the horizontal axis.
Step 3: Identify the number of layers (lines on graph) plotted.
Step 4: Draw the lines of best fit for T-X graph.
Step 5: Compute the layers thicknesses (depths) and compute layers velocities.
Step 6: Display T-X plot, layers thicknesses and velocities.
Step 7: Stop.

Flow Chart

The Algorithms given above are visually represented in the flow chart (Figure 2).

---

[8] Spiegel, M. R., (1992). Theory and problems of Statistics, 2nd Revised Ed. S.I.Ed. – (Shaum's outline series), McGraw-Hill Book Co., London.

Source Codes in Visual Basic.net

Module for Plotting the Scattered Points

```
Private Sub cmdScatterediagran_Click()
 If txtpt.Text <> "" Then
  If (flag) Then Unload frmPlotGraph
    Me.Hide
    mm = getplotdata("X_Axis", "Y_Axis_Forward")
    frmPlotGraph.Show
    mm = plotxy_axis()
    mm = scatterdiagram()
    mm = getplotdata("X_Axis", "Y_Axis_Reversed")
    mm = scatterdiagram()
  Else
    MsgBox "error in data input..."
 End If
End Sub
```

Modules for Plotting the Graph

```
Private Sub cmdPlot_Click()
 If txtpt.Text <> "" Then
  'bpp(counter) = txtpt.Text
  'counter = counter + 1
  gflg = 0
  If (flag) Then Unload frmPlotGraph
  Me.Hide
  mm = getplotdata("X_Axis", "Y_Axis_Forward")
  frmPlotGraph.Show
  mm = plotxy_axis()
  mm = scatterdiagram()
  mm = getplotdata("X_Axis", "Y_Axis_Reversed")
  'mm = plotxy_axis()
  mm = scatterdiagram()
  mm = getplotdata("X_Axis", "Y_Axis_Forward")
  mm = PlotLayerPoints(Int(txtLayer1sp.Text), Int(txtLayer1ep.Text)) '0, 15  0, 11
  tf(0) = myintercept: vf(0) = 1# / myslope
  mm = PlotLayerPoints(Int(txtLayer2sp.Text), Int(txtLayer2ep.Text)) '9, 22
  tf(1) = myintercept: vf(1) = 1# / myslope
  mm = PlotLayerPoints(Int(txtLayer3sp.Text), Int(txtLayer3ep.Text)) '18, 34
  tf(2) = myintercept: vf(2) = 1# / myslope

  'Reversed Layer
  gflg = 1
  ep1 = tn - Int(txtLayer3sp.Text) - 1
  sp2 = ep1 - (Int(txtLayer1ep.Text) - Int(txtLayer2sp.Text))
  ep2 = sp2 + (Int(txtLayer2ep.Text) - Int(txtLayer2sp.Text))
  sp3 = ep2 - (Int(txtLayer2ep.Text) - Int(txtLayer3sp.Text))

  mm = getplotdata("X_Axis", "Y_Axis_Reversed")
  'mm = PlotLayerPoints(Int(txtLayer1sp.Text), Int(txtLayer1ep.Text))
  mm = PlotLayerPoints(Int(txtLayer1sp.Text), ep1) 'Int(txtLayer1ep.Text)) '0, 15  0, 11
  tr(0) = tf(0): vr(0) = vf(0) '1# / myslope
  mm = PlotLayerPoints(sp2, ep2) 'Int(txtLayer2sp.Text), Int(txtLayer2ep.Text)) '9, 22
```

```
  'mm = PlotLayerPoints(Int(txtLayer2sp.Text), Int(txtLayer2ep.Text))
  tr(1) = tf(1): vr(1) = vf(1) '1# / myslope
  mm = PlotLayerPoints(sp3, Int(txtLayer3ep.Text)) 'Int(txtLayer3sp.Text) 18, 34
  'mm = PlotLayerPoints(Int(txtLayer3sp.Text), Int(txtLayer3ep.Text))
  tr(2) = tf(2): vr(2) = vf(2) '1# / myslope
  Ds = 0
  Zf(0) = (tf(1) / 2# * vf(1) * vf(0) / Math.Sqr(Abs(vf(1) ^ 2 - vf(0) ^ 2))) + Val(txtDs.Text) / 2
  Zf(1) = ((tf(2) - 2 * Zf(0) * Math.Sqr(Abs(vf(2) ^ 2 - vf(0) ^ 2))) / (vf(2) * vf(0))) _
  * (vf(2) * vf(1) / (2 * Math.Sqr(Abs(vf(2) ^ 2 - vf(1) ^ 2))))
  Zr(0) = (tr(1) / 2# * vr(1) * vr(0) / Math.Sqr(Abs(vr(1) ^ 2 - vr(0) ^ 2))) + Int(txtDs.Text) / 2
  Zr(1) = ((tr(2) - 2 * Zr(0) * Math.Sqr(Abs(vr(2) ^ 2 - vr(0) ^ 2))) / (vr(2) * vr(0))) _
  * (vr(2) * vr(1) / (2 * Math.Sqr(Abs(vr(2) ^ 2 - vr(1) ^ 2))))

  'frmPlotGraph.Line (x1 + 1000, y1)-(x1 + 1000, y1)
  frmPlotGraph.Line (originX + 500, originY - 5000)-(originX + 500, originY - 5000)
  frmPlotGraph.Print
  frmPlotGraph.Print Spc(200); "Layers"; Spc(5); "Time (sec)"; Spc(5); "Velocity (m/s)"
  frmPlotGraph.Print Spc(200); "======="; Spc(2); "==========="; Spc(2); "==========="
  frmPlotGraph.Print Spc(180); "Forward "
  frmPlotGraph.Print Spc(200); "Layer 1 "; Spc(5); Format(tf(0), "###0.00"); Spc(12);
Format((vf(0)), "#0.000")
  frmPlotGraph.Print Spc(200); "Layer 2 "; Spc(5); Format(tf(1), "###0.00"); Spc(12);
Format((vf(1)), "#0.000")
  frmPlotGraph.Print Spc(200); "Layer 3 "; Spc(5); Format(tf(2), "###0.00"); Spc(12);
Format((vf(2)), "#0.000")
  frmPlotGraph.Print
  frmPlotGraph.Print Spc(200); "Depth (m) = "; Spc(1); Format(Zf(0), "###0.00") '; Spc(12);
Format((vf(2)), "###0")
  frmPlotGraph.Print Spc(200); "Depth (m) = "; Spc(1); Format(Abs(Zf(1)), "###0.00") '; Spc(12);
Format((vf(2)), "###0")
  frmPlotGraph.Print Spc(180); "Reversed "
  frmPlotGraph.Print Spc(200); "Layer 1 "; Spc(5); Format(tr(0), "###0.00"); Spc(11);
Format((vr(0)), "#0.000")
  frmPlotGraph.Print Spc(200); "Layer 2 "; Spc(5); Format(tr(1), "###0.00"); Spc(11);
Format((vr(1)), "#0.000")
  frmPlotGraph.Print Spc(200); "Layer 3 "; Spc(5); Format(tr(2), "###0.00"); Spc(11);
Format((vr(2)), "#0.000")
  frmPlotGraph.Print
  frmPlotGraph.Print Spc(200); "Depth (m) = "; Spc(1); Format(Zr(0), "###0.00") '; Spc(12);
Format((vf(2)), "###0")
  frmPlotGraph.Print Spc(200); "Depth (m) = "; Spc(1); Format(Abs(Zr(1)), "###0.00") '; Spc(12);
Format((vf(2)), "###0")

  'mm = getplotdata("X_Axis", "Y_Axis_Reversed")
  'mm = displayval()

  'f2value = plotscatter()
  'Fvalue = Plotgraph()
 Else
  MsgBox "error in data input..."
 End If
End Sub
```

Module for Printing the Graph

```
Private Sub cmdPrtPlt_Click()
If (plotflag) Then
 frmPlotGraph.PrintForm
Else
  MsgBox "error in wrong output format..."
End If
End Sub
```

LVL Input Data

Table 1, Line No. 551/557 is a representative of the LVL data acquired in the East-Central Niger Delta. It shows the traveltimes for the forward and reverse shots.

RESULTS

The computer program has been designed as a user-interactive tool for seismic low velocity layer (LVL) interpretation following the various processes of composing high-quality software to implement numerical models above[9]. The program was debugged to eliminate syntax errors, Link errors, Run-time errors and Logic errors. It was test-run on a Microsoft Windows 8 Operating System (OS) to determine whether it fulfills the need for which it was designed.

The expected output is a typical display of time-offset graphs for both forward and reverse shooting schemes and the results showing the layers thickness and velocities (Figure 3 among others). The Program plots the first arrival times in milliseconds on the vertical axis and the geophone offsets in metres on the horizontal axis. The velocities, V_0, V_1 and V_2 for the various layers are accurately computed. Similarly, the various layer thicknesses, Z_0, Z_1 and the total weathering layer thickness, Z_w are also computed and displayed on the plot menu.

Table 2 is a summary of the results showing that the weathered layer has a total thickness which ranged between 12.20 m and 34.64 m; first weathered layer velocities ranging between 227.8 ms^{-1}, and 250 ms^{-1}, sub-weathered layer velocities between 428 ms^{-1} and 530 ms^{-1} and consolidated layer velocities between 1646.4 ms^{-1} and 1737 ms^{-1}. These results compare reasonably with those obtained by[10] in the same part of the Niger Delta.

CONCLUSION AND RECOMMENDATION

The various standard steps required in any software development were employed for the easy, fast and accurate analysis of the near-surface velocity data acquired in the Niger Delta by the standard geophysical technique. It has been simplified to allow for easy understanding and adoption by any Earth Scientist with minimal programming experience. Interpretation can be achieved in just a short period as computation time is reduced. Seismic refraction LVL interpretation which will normally take hours or even days can now be completed accurately within few minutes.

This software is recommended for seismologists for delineating the velocity and thickness of the weathered layer for making static corrections in 3D or 4D seismic data processing, geotechnologists

---

[9] Chapra, S. C. and R. P.Canale, (1998). Numerical methods for Engineers: with programming and software Applications. 3rd Edition, McGraw-Hill Companies, U. S. A.

[10] Uko, E. D., Ekine, A. S., Ebeniro, J. O. and Ofoegbu, C. O. (1992). Weathering structure of the east-central Niger Delta, Nigeria, Geophysics, 57, 1228-1233.

for determining the depth to bedrock for foundation studies and hydrogeologists for determining the top of the water table. Since it conducts a near-surface seismic data analysis for forward and reverse shooting schemes, it reveals dipping layers[11]; further research should be directed towards designing computer programs to automatically integrate interpretation by the uphole and the seismic refraction (LVL) surveys in a single software to properly analyze near-surface data in the Niger Delta.

FIGS AND TABLES

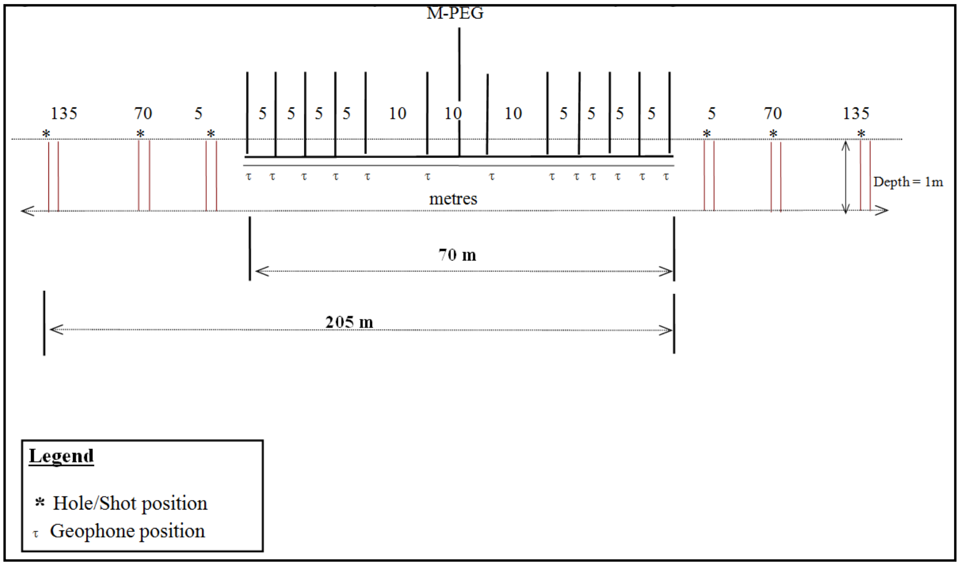

Fig. 1: Surface LVL Refraction Survey Geophone/Source array Diagram

[11] Telford, W. M., L. P. Geldart, P. E. Sheriff, and D. A. Keys, (1976). Applied Geophysics. Cambridge University Press, London.

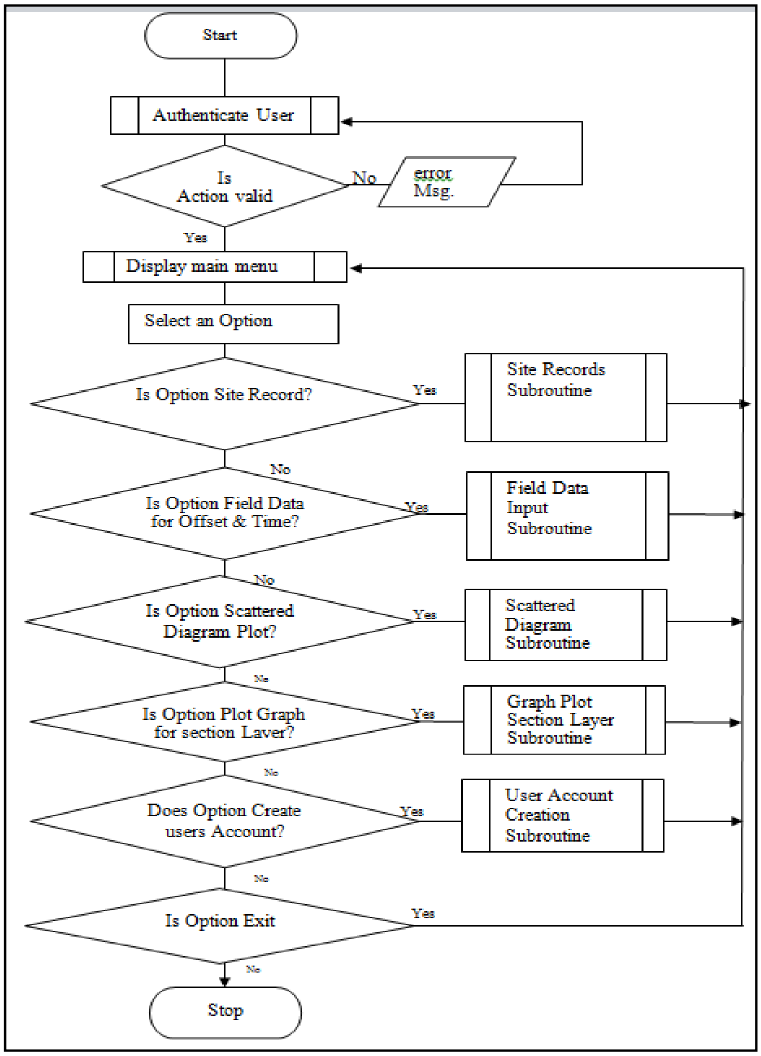

Fig. 2: Main Flow Chart

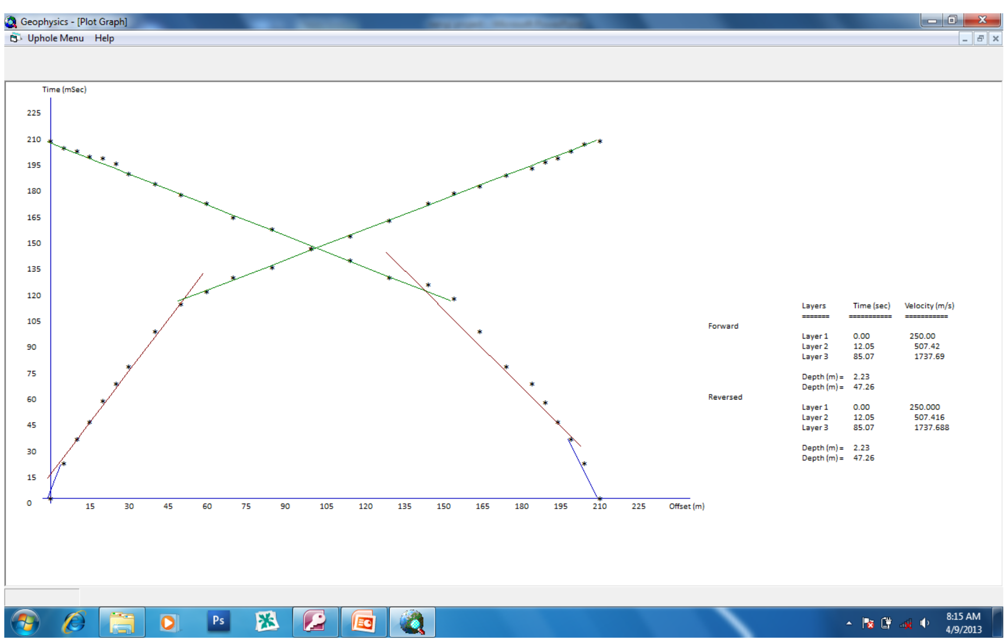

Fig. 3: Typical Display of Time-Offset Graphs (Forward and Reverse) for Line No. 551/557

Table 1: Typical Travel-time Data for Line No. 551/557

Channel No.	Travel-time (ms)	
	LVL (Forward)	LVL (Reverse)
1	20	206
2	34	202
3	44	200
4	56	197
5	66	196
6	76	193
7	96	187
8	112	181
9	119	175
10	127	170
11	133	162
12	144	155
13	151	144
14	160	137
15	170	127
16	176	123
17	180	115
18	186	96
19	190	76
20	194	66
21	196	55
22	200	44
23	204	34
24	206	20

Table 2: Summary of Results

Line No.	Velocity (ms^{-1})			Thickness (m)	
	First Weathered Layer V_0	Second Weathered Layer V_1	Consolidated Layer V_2	First Weathered Layer Z_0	Second Weathered Layer Z_1
247/253	250	466.21	1725.06	2.2	18.8
323/329	238	428.02	1663.00	1.7	18.4
397/403	250	438.87	1655.00	1.5	12.4
551/557	250	507.42	1737.00	2.4	23.6
623/629	250	500.00	1690.00	2.4	21.9
697/703	250	518.00	1695.00	2.3	23.0
773/779	250	477.42	1646.38	1.8	20.5
847/853	238	506.60	1486.76	2.1	32.5
924/929	228	530.30	1666.80	1.8	10.4