CHAPTER 6

RESULTS AND DISCUSSION

The contents of this chapter have been broadly divided into two categories. The first category is the entire analytical and computational work done on the seismic data available in the form of SEG-Y files. This has formed the base for the generation of knowledge-base that helps in the analytical interpretation by the expert system.

The second category elaborated in this chapter discusses the work done in terms of generating the knowledge base for the manual interpretation of the seismic sections. The rules helping in interpretation of seismic sections through visual inspection have been designed, populated in the knowledge-base of the expert system. These rules have been chained and executed to build the interpretation.

6.1 ANALYTICAL WORK DONE ON SEISMIC DATA

This aspect consists of C++ programs for analytical techniques of horizon tracking mentioned in the earlier chapters.

The section below gives the descriptive examples of all three aforesaid techniques, showing the entire procedure and computations done in the present work using real seismic field data.

(i)   **Horizon tracking using cross-correlation technique:**

The processed seismic data in SEG-Y format have been taken to demonstrate the example. The process starts with picking up a user-defined marker, from the first trace of the reflected wave (such as maximum amplitude value in that trace), along with its corresponding time. Then a time window, of user-specified size, is selected, with marker in the center of the window. This is cross correlated with the adjoining trace, as demonstrated by an example described below:

Example 1: Cross-Correlation

Sample indicates 4 traces in a 8 ms time window with 2 ms time interval and the procedure of cross-correlation of adjoining traces.

Time (in ms)	Amplitude value			
	Trace #1	Trace #2	Trace #3	Trace #4
2520	25044.5	24868.5	25287.3	26741.1
2522	21835.8	20238.3	19365.4	19353
2524	6493.93	3729.11	1474.81	-421.381
2526	-12725.4	-15245.9	-17677.1	-20448.4
2528	-24779.9	-25230.7	-26096.8	-27959.9

For the ease of handling, this data was scaled by multiplying with 10^{-3} and truncated to 2 decimals in the following manner.

Time	Amplitude value			
(in ms)	Trace #1	Trace #2	Trace #3	Trace #4
2520	25.04	24.87	25.29	26.74
2522	21.84	20.24	19.37	19.35
2524	06.49	03.73	01.47	-00.42
2526	-12.73	-15.25	-17.68	-20.45
2528	-24.78	-25.23	-26.10	-27.96

Cross correlating trace #1 and trace #2: Assuming trace #1 amplitude values to be x_1…..x_n and trace #2 amplitude values to be y_1….y_n. (n= 5 in this case, as 8 ms window covers 5 amplitude values with corresponding times).

The equations for computation of cross correlation coefficients are given in Figure 5.1 in Chapter 5.

Substituting, the amplitude values, from traces #1 and #2, we obtain the 5 cross-correlation coefficients, a_1 to a_5 as follows:

(1) 6.4956 (2) 5.2209 (3) 1.9390 (4) -0.3242 (5) -0.0498

In order to eliminate noise, the maximum of the cross-correlation coefficients less than a threshold value has been discarded. The choice of the threshold value is somewhat arbitrary and has been taken to be 0.2 in the present case. In the above example, values from (1) to (5) are the computed cross-correlation coefficient values. The value at (1) 6.4956 is found to be maximum. This when compared to threshold value 0.2, is found to be greater, and hence the trace#2 is retained.

If the maximum value of cross-correlation coefficients had been less than 0.2, say 0.15, then we would have simply discarded trace#2 from further consideration.

Then the maximum amplitude from the window, on trace #2 has been picked as the marker and the same process repeated between traces #2 and #3. This process has been continued until all the traces have been examined in a given section. Figure 6.1 shows the correlated reflection horizon where the above explained technique was used using 2001 traces from a data sample. (These 2001 traces cover a distance of 50,025 meters adjoining the explosion).
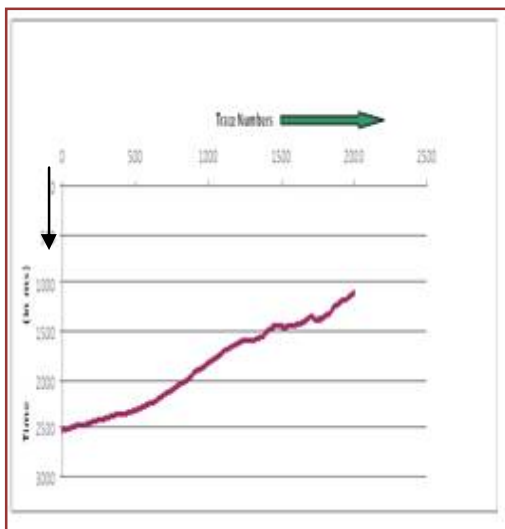
Figure 6.1: Correlated reflection horizon using 2001 traces.
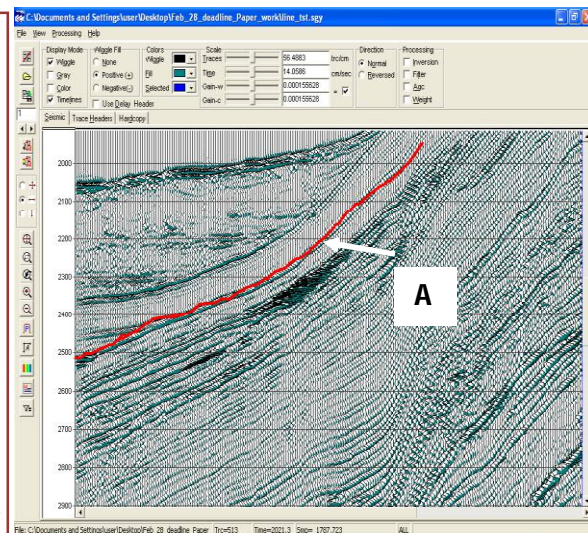
Figure 6.2: SEG-Y section corresponding to correlated horizon shown in Figure 6.1

The slope of line in Figure 6.1 corresponds to the slope of the reflecting horizon *A* in Figure 6.2. The correlated horizon attains a steep upward slope. This corresponds to an interface between two types of rocks.

A close similarity between the computed horizon and the one seen in SEG-Y snap, establishes the correctness of the approach. This represents a cross section of the geological structure along the line on which the reflected signals were recorded. Other markers can be picked up and followed in the same fashion. Several markers can be traced across the depth.

In the case shown above, the maximum amplitude along with its time was chosen, after inspecting the entire trace, in order to proceed with cross-correlation.

Another case demonstrated here is where the user can give a random time interval range, between which the trace of reflected wave is examined and maximum amplitude along with its time is taken as its starting point to begin cross-correlation.

Example 2: Cross-Correlation in a random time interval range

The same procedure has been repeated by providing a random range of time interval (by visually detecting the strong signal in SEG-Y section). The strong reflecting horizon *B* (see Figure 6.4) was found between the time interval range 3700 to 3800 ms. This time interval range was chosen and provided to the program.
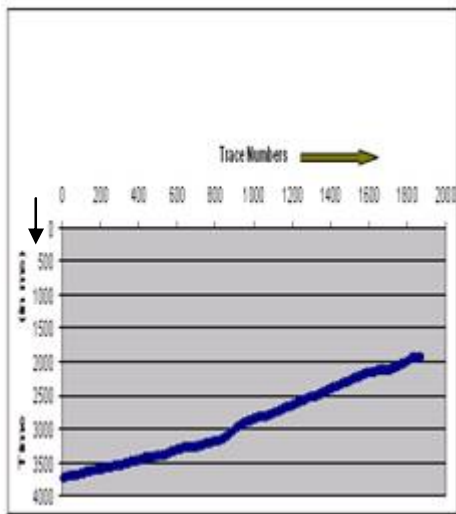

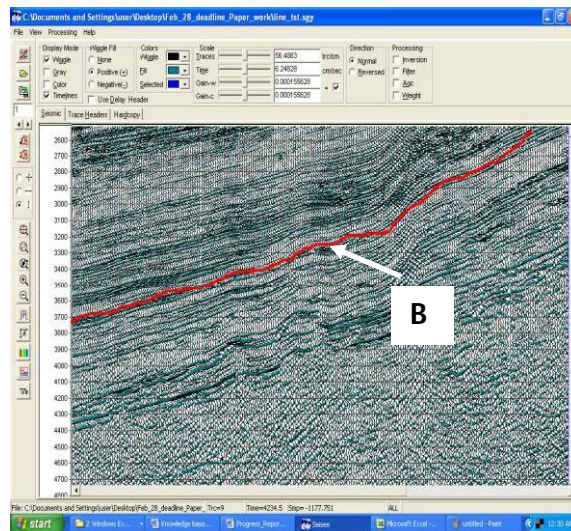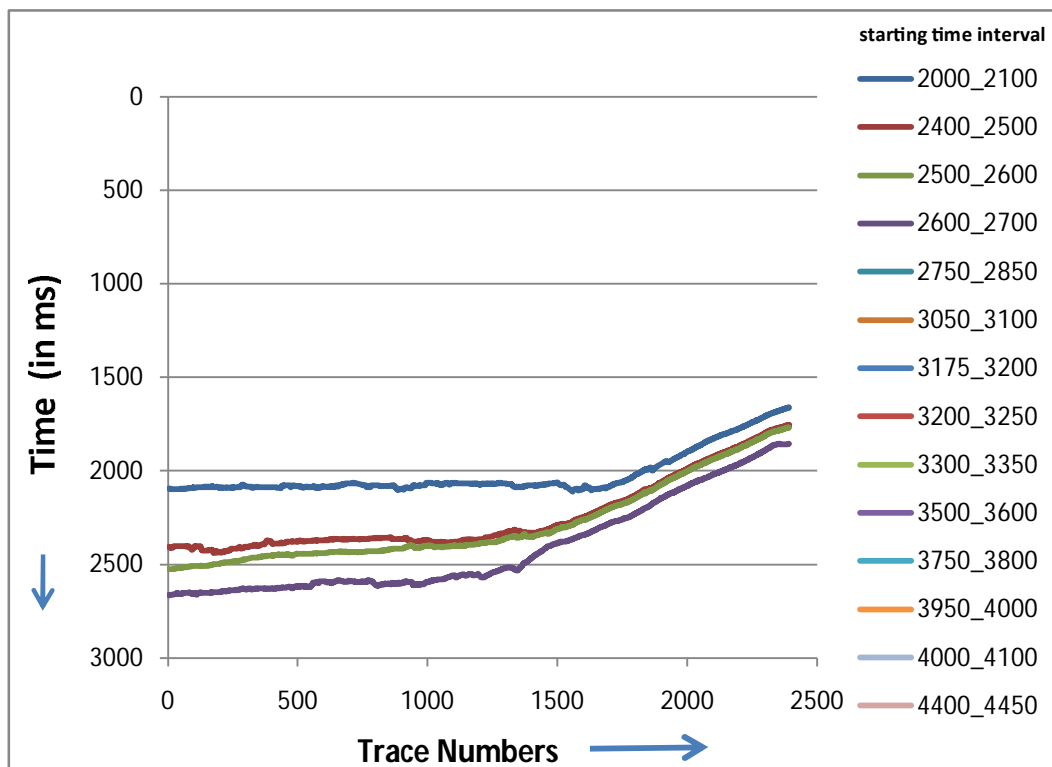**Figure 6.3: Cross-correlated reflection horizon using random time interval range**

Figure 6.4: SEG-Y section corresponding to correlated horizon shown in Figure 6.3.

Figure 6.3 shows the cross-correlated reflection horizon for the same SEG-Y data file used in Example 1.

The slope of the graph in both cases is seen steeply rising up from time interval 3700 to 2000 ms.

The complete procedure discussed above is built using C++ program codes, which can be invoked from Visual Basic front-end interface to execute and write their outputs to the output files which can be plotted using suitable plotting programs such as 'Ms Excel'. Using the resultant plots some interpretation rules have been developed and added to the rule-base component of the expert system which has been integrated with VB using Intelligence Server.

The cross-correlation technique was applied on several reflecting horizons each for three seismic sections of anonymous geological areas. Figure 6.5 shows 14 reflecting horizons that have been tracked using cross-correlation technique program. These have been compared with the corresponding SEG-Y snap shown in Figure 6.6.

Figure 6.5 Reflecting horizons tracked using cross-correlation technique corresponding to horizons seen in Figure 6.5
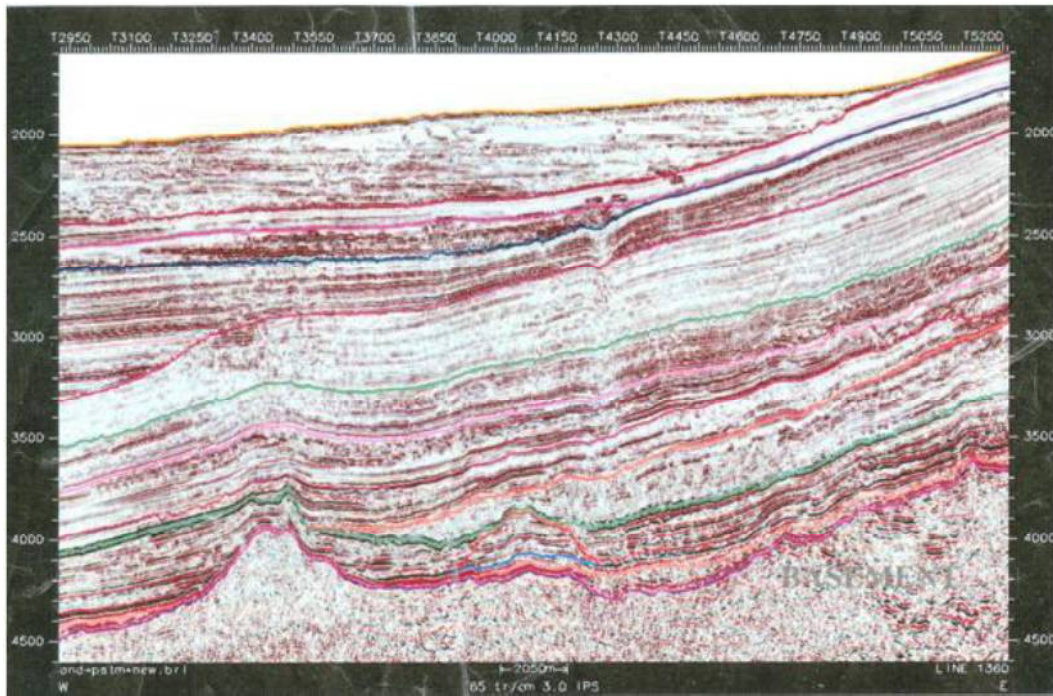
Figure 6.6 SEG-Y snap showing reflecting horizons

In another seismic section 8 reflecting horizons were tracked using cross-correlation technique. Figure 6.7 shows the tracked reflecting horizons and Figure 6.8 shows the corresponding SEG-Y snap.
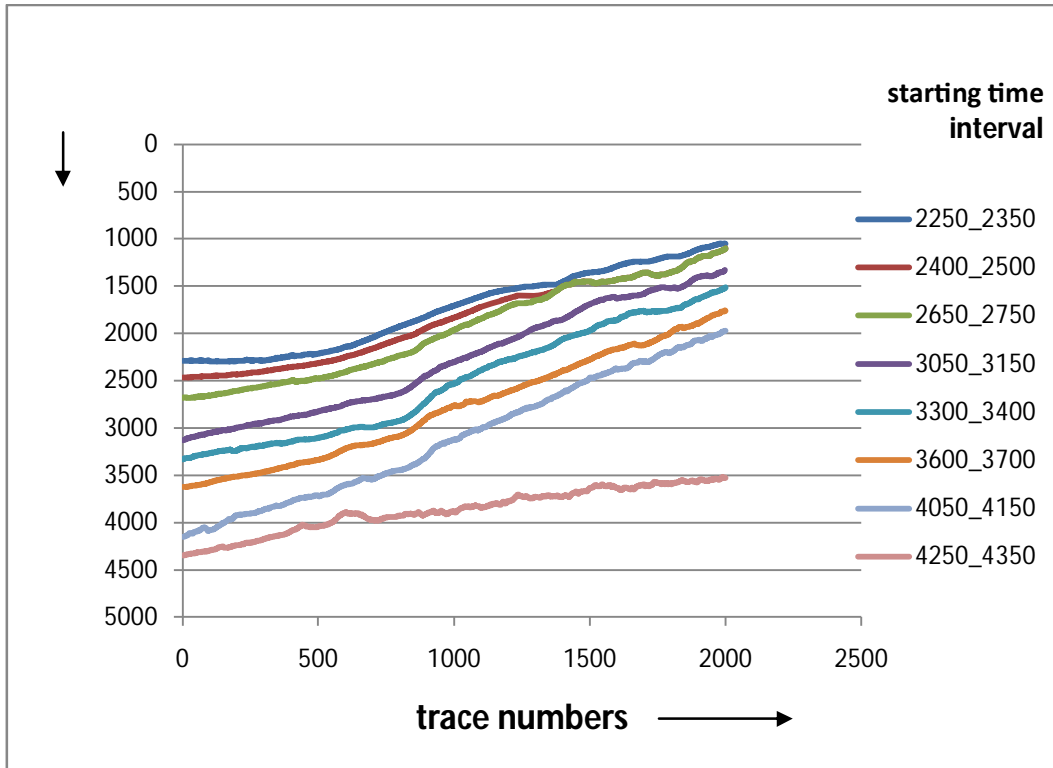
Figure 6.7 Tracked reflected horizons using cross-correlation technique
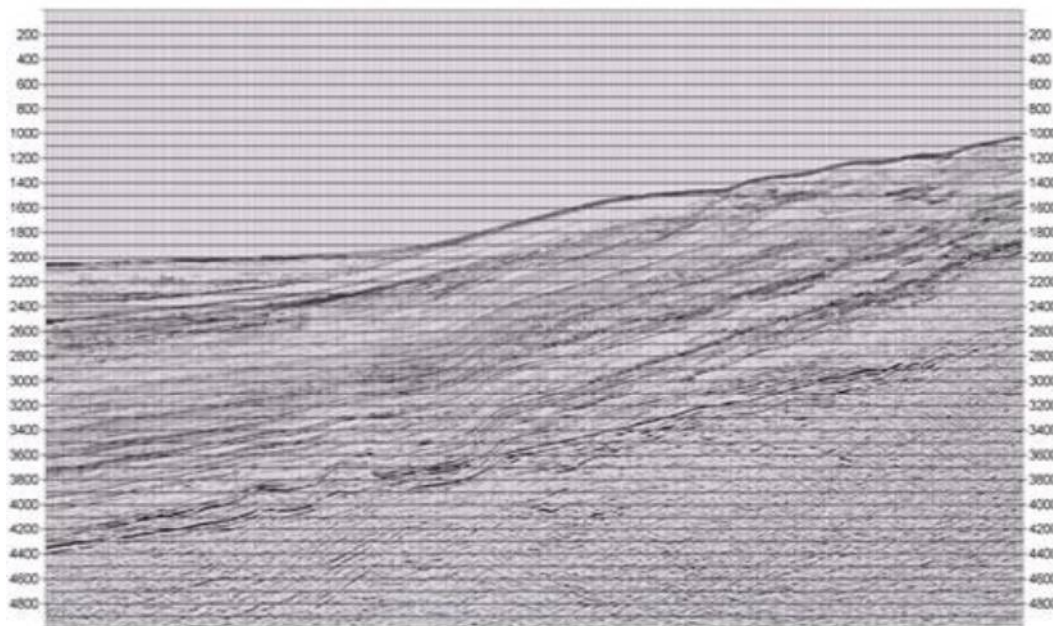

Figure 6.8 corresponding to horizons shown in Figure 6.7

Comparing the reflected horizons tracked using cross-correlation with their corresponding SEG-Y snaps, the main observation that was made was that the

cross-correlation technique works well for continuous horizons. On close observation, it can be seen that all the reflecting horizons, except the reflecting horizon 4000-4100 marked with light blue color in Figure 6.5 are found to be identical to the corresponding SEG-Y snap in the Figure 6.6. There is discontinuity/break in the above mentioned horizon (Figure 6.6), which is missing from cross-correlated horizon (Figure 6.5). The cross-correlation may fail when there are discontinuities such as faults, as will be discussed next.

We have already discussed in Chapter 5, how to compute complexes using Hilbert transformation. Procedure for calculation of instantaneous phase and reflection strengths, were also included in that chapter.

(ii)   Generation of complexes through Hilbert transform

The language of technical computing MATLAB has been used for the purpose of computation of complexes using Hilbert Transform of the given seismic trace data. MATLAB provides a built-in function 'Hilbert' for this purpose.

Firstly, the trace data for the given seismic section are put in a '.txt' file and loaded into MATLAB environment. Then file name is passed as a parameter to the 'Hilbert' function which supplies the complexes of the trace data as its output.

The following commands are typed in the command window of MATLAB for the same:

<div align="center">

load **il1360.txt**

ans=*Hilbert***(il1360.txt)**

</div>

Table 6.1 shows a sample of contents of file il1360.txt and Table 6.2 shows the computed complexes.

Table 6.1: A sample of input file to 'Hilbert'

Time (in ms)	Amplitude				
	Trace#1	Trace#2	Trace#3	Trace#4	Trace#5
4002	-0.37	-0.24	0.26	0.76	0.97
4004	0.32	0.53	1.05	1.49	1.63
4006	1.11	1.32	1.72	2	2.06
4008	1.77	1.92	2.14	2.2	2.13
4010	2.11	2.15	2.21	2.08	1.87
4012	2.04	1.96	1.91	1.69	1.37
4014	1.62	1.45	1.31	1.09	0.76
4016	1.02	0.76	0.54	0.36	0.11
4018	0.39	0.07	-0.22	-0.38	-0.5
4020	-0.15	-0.49	-0.83	-0.98	-0.97
4022	-0.59	-0.86	-1.19	-1.32	-1.23
4024	-0.9	-1.04	-1.29	-1.38	-1.24
4026	-1.08	-1.04	-1.18	-1.19	-1.03
4028	-1.1	-0.91	-0.9	-0.85	-0.72
4030	-0.97	-0.67	-0.53	-0.46	-0.4
4032	-0.72	-0.38	-0.12	-0.08	-0.13
4034	-0.41	-0.07	0.24	0.23	0.09
4036	-0.08	0.21	0.48	0.42	0.25
4038	0.23	0.43	0.57	0.48	0.37
4040	0.47	0.53	0.52	0.43	0.43
4042	0.63	0.52	0.37	0.33	0.45
4044	0.7	0.41	0.19	0.25	0.44
4046	0.68	0.26	0.07	0.24	0.43
4048	0.65	0.18	0.07	0.3	0.44
4050	0.67	0.29	0.25	0.47	0.52

Each complex can be seen as the combination of real and imaginary parts. For ex, the complex number at time 4002 ms for trace#1 is found to be -0.37-1.82i (Table 6.2), '-0.37' is the real part (Table 6.1) given as input and '-1.82i' is the imaginary part.

Table 6.2 Complex numbers corresponding to amplitudes given in Table 6.1

Time (in ms)	Amplitude				
	Trace#1	Trace#2	Trace#3	Trace#4	Trace#5
4002	-0.37-1.82i	-0.24-1.91i	0.26-2.09i	0.76-2.06i	0.97-1.82i
4004	0.32-1.99i	0.53-2i	1.05-1.92i	1.49-1.68i	1.63-1.4i
4006	1.11-1.83i	1.32-1.71i	1.72-1.42i	2-1.02i	2.06-0.702i
4008	1.77-1.28i	1.92-1.07i	2.14-0.681i	2.2-0.23i	2.13+0.109i
4010	2.11-0.475i	2.15-0.198i	2.21+0.184i	2.08+0.553i	1.87+0.841i
4012	2.04+0.38i	1.96+0.643i	1.91+0.99i	1.69+1.22i	1.37+1.37i
4014	1.62+1.05i	1.45+1.28i	1.31+1.58i	1.09+1.69i	0.76+1.67i
4016	1.02+1.45i	0.76+1.61i	0.54+1.85i	0.36+1.89i	0.11+1.73i
4018	0.39+1.55i	0.07+1.6i	-0.22+1.76i	-0.38+1.77i	-0.5+1.54i
4020	-0.15+1.45i	-0.49+1.35i	-0.83+1.39i	-0.98+1.37i	-0.97+1.15i
4022	-0.59+1.2i	-0.86+0.954i	-1.19+0.851i	-1.32+0.781i	-1.23+0.624i
4024	-0.9+0.853i	-1.04+0.499i	-1.29+0.281i	-1.38+0.164i	-1.24+0.0726i
4026	-1.08+0.444i	-1.04+0.0654i	-1.18-0.236i	-1.19-0.365i	-1.03-0.368i
4028	-1.1+0.00697i	-0.91-0.309i	-0.9-0.63i	-0.85-0.717i	-0.72-0.635i
4030	-0.97-0.385i	-0.67-0.587i	-0.53-0.867i	-0.46-0.89i	-0.4-0.743i
4032	-0.72-0.686i	-0.38-0.749i	-0.12-0.933i	-0.08-0.9i	-0.13-0.746i
4034	-0.41-0.863i	-0.07-0.795i	0.24-0.821i	0.23-0.772i	0.09-0.685i
4036	-0.08-0.928i	0.21-0.731i	0.48-0.593i	0.42-0.561i	0.25-0.59i
4038	0.23-0.872i	0.43-0.569i	0.57-0.322i	0.48-0.339i	0.37-0.478i
4040	0.47-0.731i	0.53-0.352i	0.52-0.0887i	0.43-0.171i	0.43-0.355i
4042	0.63-0.546i	0.52-0.157i	0.37+0.0477i	0.33-0.106i	0.45-0.26i
4044	0.7-0.347i	0.41-0.0289i	0.19+0.0392i	0.25-0.142i	0.44-0.197i
4046	0.68-0.199i	0.26-0.0428i	0.07-0.103i	0.24-0.239i	0.43-0.187i
4048	0.65-0.147i	0.18-0.212i	0.07-0.338i	0.3-0.365i	0.44-0.226i

(iii)   Calculation of 'Instantaneous Phase'

The calculation of 'Instantaneous Phase' $\theta(t)$ is done in Ms Excel by using IMARGUMENT( ). This function takes the complex number and returns the argument, which is an angle in radians. In order to convert the angle in degrees

for ease of operation, the angle in radians is multiplied with 57.3. Table 6.3 shows a sample of instantaneous phase $\theta(t)$, for data given in Table 6.2.

Table 6.3: Sample Instantaneous phases in degree for data given in Table 6.2

Time (in ms)	Instantaneous phase (in degrees)				
	Trace#1	Trace#2	Trace#3	Trace#4	Trace#5
4002	-101.499	-97.169	-82.915	-69.755	-61.948
4004	-80.871	-75.163	-61.331	-48.434	-40.662
4006	-58.765	-52.338	-39.545	-27.024	-18.819
4008	-35.876	-29.133	-17.654	-5.969	2.93
4010	-12.688	-5.262	4.76	14.89	24.217
4012	10.553	18.164	27.401	35.828	45.003
4014	32.952	41.44	50.341	57.183	65.535
4016	54.88	64.735	73.733	79.222	86.368
4018	75.882	87.501	97.132	102.124	107.995
4020	95.913	109.957	120.851	125.586	130.156
4022	116.19	132.043	144.441	149.4	153.112
4024	136.546	154.379	167.724	173.235	176.662
4026	157.664	176.415	-168.702	-162.96	-160.351
4028	179.65	-161.256	-145.019	-139.862	-138.6
4030	-158.363	-138.788	-121.447	-117.341	-118.305
4032	-136.395	-116.909	-97.336	-95.087	-99.893
4034	-115.42	-95.039	-73.71	-73.415	-82.521
4036	-94.934	-73.977	-51.016	-53.183	-67.041
4038	-75.23	-52.925	-29.465	-35.234	-52.262
4040	-57.265	-33.593	-9.681	-21.688	-39.545
4042	-40.917	-16.802	7.347	-17.809	-30.021
4044	-26.37	-4.032	11.658	-29.599	-24.121
4046	-16.313	-9.349	-55.804	-44.884	-23.505
4048	-12.744	-49.671	-78.305	-50.586	-27.189
4050	-13.674	-58.11	-67.587	-47.229	-31.53

(iv)   Calculation of 'Reflection strength'

The calculation of 'reflection strength' A(t) has been done in Ms Excel by using its IMABS( ). This function when supplied with the complex numbers returns the absolute value of the complex, generating the 'reflection strength' attribute. Table 6.4 shows reflection strength corresponding to data in Table 6.2

Table 6.4: Reflection strength for complexes given in Table 6.2

Time (in ms)	Reflection strength				
	Trace#1	Trace#2	Trace#3	Trace#4	Trace#5
4002	1.85855	1.92468	2.103	2.19776	2.06269
4004	2.01812	2.07315	2.18776	2.24268	2.15159
4006	2.13818	2.16197	2.23338	2.24699	2.17624
4008	2.18199	2.19622	2.24563	2.21197	2.13277
4010	2.16288	2.15911	2.21762	2.15229	2.0506
4012	2.07517	2.06268	2.15119	2.08354	1.94045
4014	1.93305	1.93518	2.05431	2.00756	1.83106
4016	1.77378	1.77591	1.92552	1.91981	1.73016
4018	1.60263	1.60539	1.77724	1.81227	1.62069
4020	1.45642	1.43448	1.62296	1.68055	1.50301
4022	1.33899	1.28427	1.46296	1.53397	1.37913
4024	1.24025	1.15334	1.32031	1.38972	1.24212
4026	1.16756	1.04205	1.2033	1.24482	1.09374
4028	1.10002	0.96095	1.09861	1.1117	0.95973
4030	1.04344	0.89051	1.016	1.00159	0.84398
4032	0.99436	0.84007	0.9406	0.90374	0.75679
4034	0.95523	0.7984	0.85498	0.80572	0.69132
4036	0.93146	0.76015	0.76299	0.70103	0.64104
4038	0.90201	0.71294	0.65449	0.58757	0.60411
4040	0.86935	0.63641	0.52751	0.46289	0.55772
4042	0.83358	0.54315	0.37306	0.3465	0.51984
4044	0.7812	0.41101	0.19401	0.2876	0.48227
4046	0.70852	0.2635	0.12419	0.33847	0.46902
4048	0.66639	0.27781	0.34487	0.47239	0.49451
4050	0.6895	0.5486	0.65589	0.69235	0.60981

(v)   Tracking Discontinuities

Consider a SEG-Y section shown in Figure 6.9 where a fault is known to be present between trace numbers 3450 and 3650. Let us try tracking a horizon across this interval starting at time instant 4092 ms where a strong reflection exists. We first use cross correlation and the resultant horizon is shown in Figure 6.10 by curve 1. Starting with the same time instant we next track the instantaneous phase across the traces and the resultant horizon is also shown in the same figure by curve 2.

A comparison of the two curves with the SEG-Y section (Figure 6.9) clearly shows that while cross-correlation failed to track the horizon across the fault plane, the instantaneous phase identified the horizon correctly all the way. A closer look shows that the cross-correlation as it progressed across the fault plane, picked another strong reflection horizon and moved along this horizon which was actually, a horizon above the one we started to track but because of the rock movement downwards, after the fault it juxtaposed with the upper horizon before the 'fault'. Since, cross-correlation only correlates the strongest signal in the next trace it makes no difference which horizon it picks next. However, instantaneous phase essentially remains constant for a given horizon and hence tries to identify the same horizon across the fault plane.
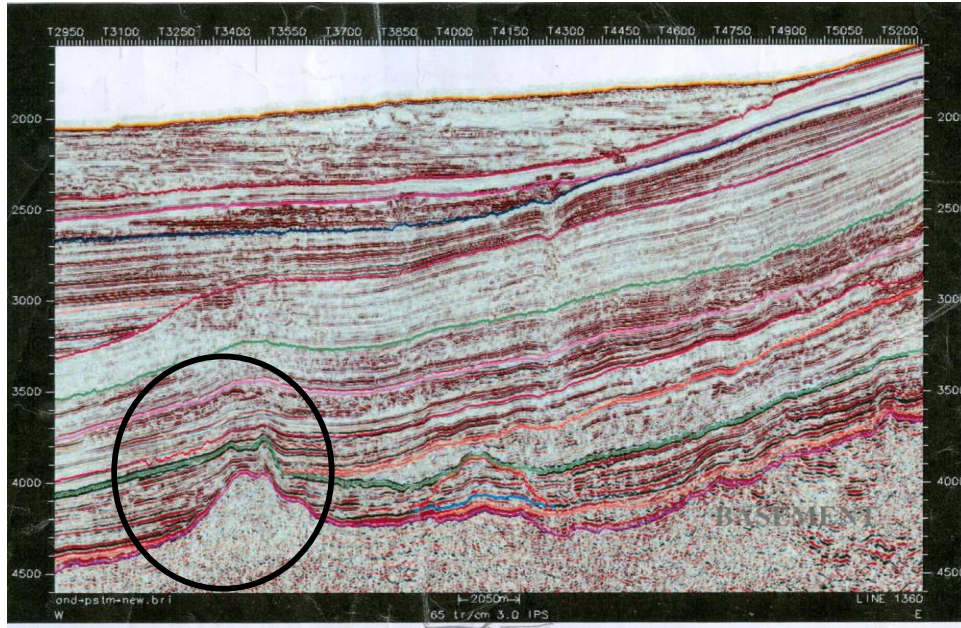
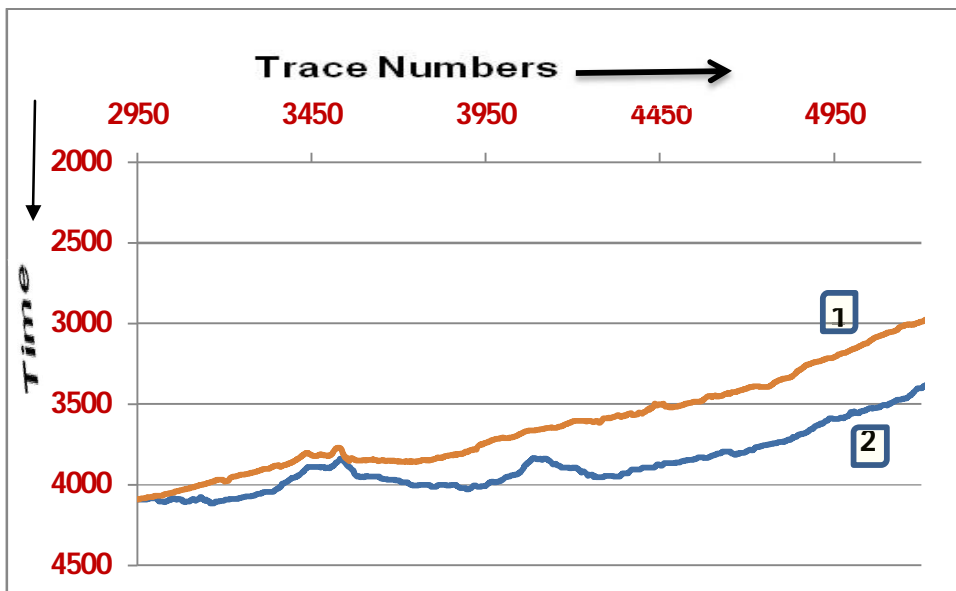**Figure 6.9: SEG-Y section showing a fault at bottom left**

Figure 6.10: Tracking across a fault plane using: cross-correlation curve 1, instantaneous phase curve 2.

(vi)   Tracking discontinuities using 'Reflection strength'

Reflection strength attribute was used to track the same horizon. The procedure used was similar to the one used with 'Instantaneous Phase'. The starting time interval of the beginning of the above tracked cross-correlation plot (horizon) was taken as the starting point and its corresponding reflection strength was identified.

Taking the window size of 7 time instants with the time of the chosen reflection strength as the centre, the nearest reflection strength was identified and the corresponding time interval was taken to be the centre for finding the next nearest match. This process was continued until all the traces were examined. The resultant plot is shown in Figure 6.11.

A comparison with Figure 6.9 and Figure 6.10 shows that 'reflection strength' too tracked the horizon across the fault plane correctly.
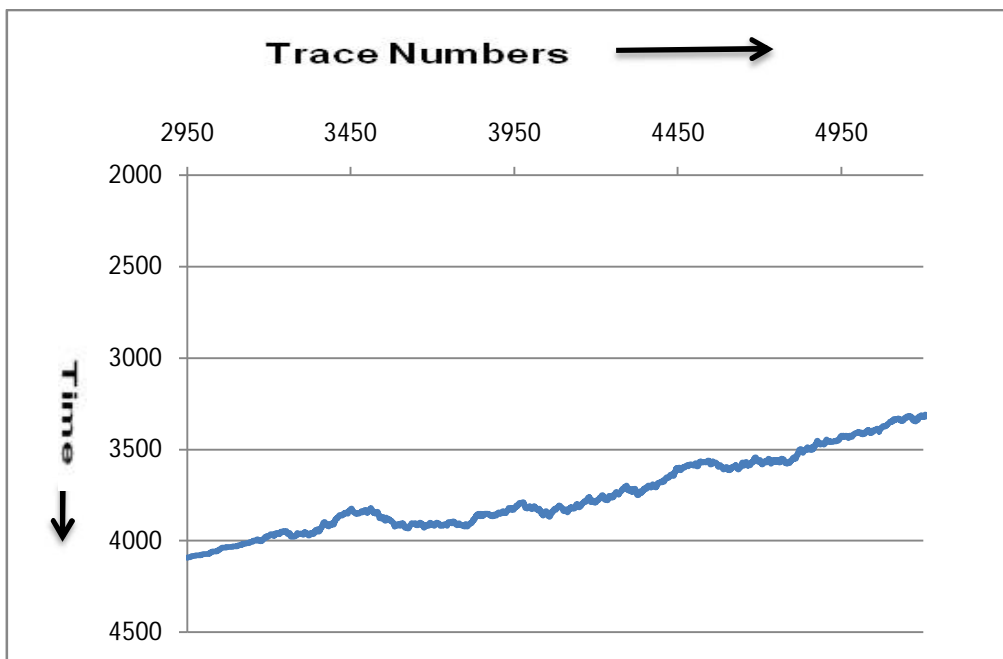

Figure 6.11: Tracking across a fault plane using 'reflection strength'.

(vii)   **Tracking fault by traversing in reverse direction (backwards)**

In the close vicinity of the fault plane, there is intense heat generated because of rock shearing and this leads to changes in the rock texture and composition. Because of this change in the nature of the rock, instantaneous phase and reflection strengths may also change making it difficult to delineate the precise location of the fault plane.

To be able to locate the fault plane, more accurately, the horizon was tracked from left to right until a small distance from the fault plane on the left. The same horizon was tracked in the reverse direction, from right to left, starting on the other end of the traces until a short distance from the fault plane on the right. The fault plane can be located more precisely in this interval. Figure 6.12 shows the reverse track.
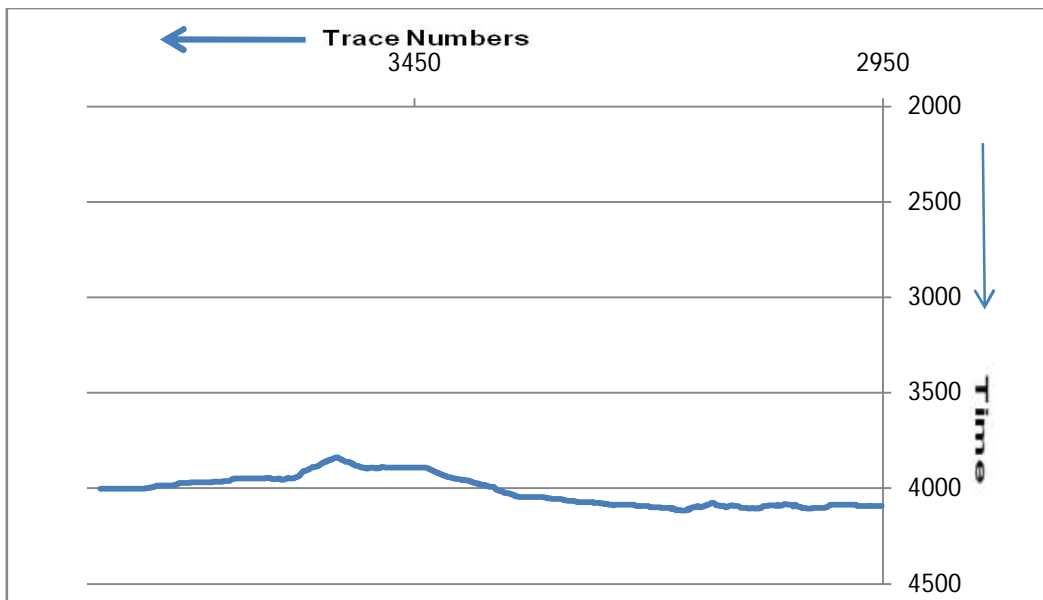
Figure 6.12: Horizon traced backwards using Instantaneous Phase.

6.2 DEVELOPMENT OF RULES FOR RULE-BASE

The development of the rule-base has been done in two broad categories, manual interpretation rules and analytical interpretation rules.

6.2.1 Manual Interpretation Rule-base

The manual interpretation rules are based on the information of the subsurface geology and geological structures that can be obtained by visually inspecting the sections. The details like 'whether the section is from land exploration or from an offshore exploration site?' can be ascertained. Further, the characteristics of reflections can provide clues regarding density of the horizons. Dark reflections are indicative of higher density and vice-versa. The shape of the reflections can provide information regarding the shape of the bed, for example, the reflections moving upwards in the seismic section are indicative of 'upward inclined beds'. The existence of slope reversal, may indicate presence of a structure and depending upon the change of slope, with respect to distance, it may be further discriminated into a syncline or an anticline. 'Break' in the reflections could be due to discontinuities. The presence of vertical shift in the horizon indicates that it could be a fault. Observing the direction of movement of the hanging block with respect to the foot wall along the fault plane indicates the type of fault whether 'thrust fault' or 'normal fault'.

Over 70 rules have been designed using 'Flex' and added to the knowledge-base in the form of a '.*ksl*' file. These rules have been added into '*rulesets*' which have been chained in forward chaining manner in the '*actions*' which execute the expert system. Rules gather their information using questions. About 35 questions have been designed for this manual interpretation part. The answers are to be provided by the user, by choosing from the given options. These options have been separately provided in the form of '*groups*'. Every question is given the

'because' clause that gets displayed to provide explanation to the user, regarding 'why the question is being asked?' and/or 'how it needs to be answered?'

A sample of the developed questions, groups, rules, rulesets, and actions under the manual interpretation aspect, is given in Table 6.5, Table 6.6, Table 6.7, Table 6.8 and Table 6.9 respectively.

Table 6.5 Sample of Questions for manual interpretation

```
question q4
What is the nature of reflections?;
choose one of grp_ref_nature
because The nature of reflections is a good indicator of
the nature of bed.


question q5
Following a particular horizon as we move from left to
right, how does it slope?;
choose one of grp_slope_direction
because The Direction of slope upwards or downwards is
indicative of whether the bed is inclined upwards or
downwards.


question q5_1
As the bed slopes upwards then what happens to the slope
with distance?;
choose one of grp_up_slope_further_change
because The further change in the upward slope may
elicite information about presence of structures of
interest.
```

Table 6.6 Sample of groups for answer choices for manual interpretation

```
group grp_ref_nature
'Horizontal', 'Inclined', 'Wavy', 'Disorderly'.


group grp_slope_direction
'Upwards','Downwards'.


group grp_up_slope_further_change
'Slope Does Not Change', 'Slope Changes Gradually',
'Slope Changes Sharply'.
```

Table 6.7 Sample of rules for manual interpretation

```
rule r1_q4
if ref_nature is q4 and
q4='Horizontal'
then
write('If the reflections are horizontal it is indicative
of horizontal beds').


rule r2_q4
if ref_nature is q4 and
q4='Inclined'
then
write('If the reflections are Inclined it is indicative
of Inclined beds').


rule r3_q4
if ref_nature is q4 and
q4='Wavy'
then
write('If the reflections are Wavy it is indicative of
subparallel beds').


rule r4_q4
if ref_nature is q4 and
q4='Disorderly'
then
write('If the reflections are disorderly it is indicative
of hommocky or contorted beds').


rule r1_q5
if ref_direction is q5 and
q5='Upwards'
then
write('If the horizon slopes upwards the bed is inclined
with a positive slope').


rule r3_q5_1
if upward_slope_change is q5_1 and
q5_1='Slope Does Not Change'
then
```

```
write('If the bed slopes upwards and the slope does not
change, then it indicates that there is no geological
structure in that section').
```

```
rule r4_q5_1
if upward_slope_change is q5_1 and
q5_1='Slope Changes Gradually'
then
write('If the bed slopes upwards and the slope shows a
gradual change, then it indicates that a geological
structure such as Fold or Anticline may be present').

rule r5_q5_1
if upward_slope_change is q5_1 and
q5_1='Slope Changes Sharply'
then
write('the slope changes sharply, then it indicate a
magmatic intrusion such as a salt dome or a thrust fault').

rule r2_q5
if ref_direction is q5 and
q5='Downwards'
then
write('If the horizon slopes  downwards the bed is inclined

with a negetive slope').
```

Table 6.7 Sample of rules for manual interpretation continued

Table 6.8 Sample of rulesets for manual interpretation

```
ruleset set_set4
contains r1_q3,r2_q3,r1_q4,r2_q4,r3_q4;
update ruleset by removing each selected rule;
when a rule misfires do true.


ruleset set_set6
contains r1_q5,r2_q5;
update ruleset by removing each selected rule;
when a rule misfires do true.


ruleset set_set7
contains r3_q5_1,r4_q5_1,r5_q5_1;
update ruleset by removing each selected rule;
when a rule misfires do true.
```

Table 6.9 Sample of actions generated for manual interpretation

```
action run_a6;
if q4='Horizontal' or q4='Inclined' or q4='Wavy' or
q4='Disorderly'
then
write(q4) and nl
and
write('The nature of bed is being chosen') and nl
and invoke ruleset set_set6 and run_a7
else
write('nature of reflection not chosen') and nl
end if.


action run_a7;
if q5='Upwards'
then
write(q5)
and write('The bed seems to be sloping upwards') and nl
and invoke ruleset set_set7 and run_a8
else
write('The bed seems to be sloping downwards') and nl
and
invoke ruleset set_set8 and run_a11
 end if.


action run_a8;
if q5_1='Slope Does Not Change'
then
write('The bed may be considered inclining upwards very
slightly without a significant change')
and nl
and invoke ruleset set_set13 and run_a18
else
write('There may be some change in slope') and
run_a9
end if.


action run_a9;
if q5_1='Slope Changes Gradually'
then
```

```
write('It can be termed as a bed which is gradually
inclining up ') and nl
 and
invoke ruleset set_set8 and run_a11
else
write('The bed is not gradually inclining upwards')
and nl and
run_a10
end if.
```

Table 6.9 Sample of actions for manual interpretation continued

```
action run_a10;
if q5_1='Slope Changes Sharply'
then
write('Sharp slope change may be due to magmatic
intrusion')
and nl and
invoke ruleset set_set9 and invoke ruleset set_set13 and
run_a18
else
write('This sharp slope change may be caused due to thrust
faults') and nl
and invoke ruleset set_set13 and run_a18
end if.
```

6.2.2. Analytical Interpretation Rule-base

The rules in this part of the rule-base are based on analysis done on seismic data. In the earlier section of this chapter it was described that the amplitude of the reflected wave is used in the cross-correlation technique and the continuous horizons are tracked. For discontinuous horizons, the computed derived attributes 'instantaneous phase' and 'reflection strength' are used.

Over 50 rules have been designed based on the above process and populated into this part of the rule-base. In the same manner, as done in the manual interpretation the *rules* are categorized and added to the *rulesets* and appropriately chained to be invoked by the *actions*. Questions have been designed to acquire responses from

the user, and the information obtained in this manner is used by the rules for firing. As the rules fire, the expert system proceeds towards building the interpretation, that is finally displayed to the user.

This part of the expert system holding the analytical rules can be put to use only when the seismic data is available along with the seismic snap. Initially, the process of interpretation starts with the manual interpretation rules applied on a particular horizon and the resultant interpretation based on the visual inspection is built, and then the user is prompted to input his consent for verifying the obtained results with analytical interpretation rules from the rule-base.

The analytical interpretation invokes the appropriate application programs to apply analytical techniques on seismic data and generate the output plots which depict the tracked reflected horizon. The analytical interpretation rules are applied on these plots.

The rules such as, '*if the output of the cross-correlation technique is found to be similar to the corresponding horizon in the snap, then it indicates that the result of manual interpretation is been verified*', and '*if the horizon tracked using cross-correlation technique is found to be non-identical to the corresponding horizon in the snap, then it indicates that there is a discontinuity in the horizon*', and '*if there is a discontinuity then, the information regarding the existence of a shift in the vertical plane, needs to be found out*'.

Then the other set of rules like, '*if there is a vertical shift, then the discontinuity could be because of a fault*' and '*if the direction of the shift is upwards, that is, the hanging block of the fault, moves up in the vertical plane, with respect to the foot wall, then it could be the thrust fault*'.

The rules are thus fired, as per the chaining scheme framed during the design of the rule-base and the resultant interpretation is displayed to the user. This analytical component works to verify the results of the manual interpretation and

also further enrich the resultant interpretation, due to the contribution of analysis of seismic data.

The samples of questions, groups, rules, rulesets and actions that have been used in analytical interpretation have been presented in the Table 6.10, Table 6.11, Table 6.12, Table 6.13 and Table 6.14 respectively.

Table 6.10 Sample of questions based on analytical plots

```
question tech_part_p2_q1
In view of amplitude-time data being available it is
possible to analytically
compute some attributes to confirem the observations
already made on the horizon :
cross-correlation between adjacent traces is an analytical
technique to follow a
reflection horizon in the way as you see in the snap: would
you like to invite that
procedure now?;
choose one of grp_choice
because based on confirmation the developed procedure can
be invited and executed to confirm observations.



question tech_part_p3_q3
Choose the window size for cross-correlation;
choose one of grp_ccr_window_size
because the size of window for cross-correlation needs to
be supplied to the program for crosscorrelation technique.


question tech_part_p4_q2
compare this plot with corresponding horizon in the seismic
map: does the plot appear to be
identical to the horizon in the snap?;
choose one of grp_choice
because Comparison with the obtained plot would verify the
observation made by the earlier visual inspection.



question tech_part_p4_q3
Is the break/fault seen in the snap missing in the plot?;
choose one of grp_choice
```

```
because if the break seen in the snap is missing in the
plot, it indicates that cross correlation has not
been able to track the discontinuity.


question tech_part_p4_q4
To confirm the presence or absence of fault you may invite
other seismic attributes such as
Instantaneous phase or reflection strength: would you like
to see Instantaneous phase along the
horizon under observation?;
choose one of grp_choice
because Instantaneous phase is characteristic of a given
reflecting horizon and is almost constant,
hence is able to track the reflecting horizon better than
the cross correlation.
```

Table 6.10 Sample of questions based on analytical plots continued

```
question tech_part_p4_1_q1
Do you wish to see the obtained plot depicting the
identified horizon?;
choose one of grp_choice
 because Obtained plot can be displayed to be compared with
           the seismic snap examined earlier.
```

Table 6.11 Sample of groups for answer choices based on analytical plots

```
group grp_choice
'Yes','No'.

group grp_ccr_window_size
'window size 5 time intervals', 'window size 7 time intervals'.
```

Table 6.12 Sample of rules based on analytical plots

```
rule r1_tech_part_p2_q1
if prg_invite_ccr is tech_part_p2_q1 and
prg_invite_ccr = 'Yes'
then
write('The developed cross-correlation procedure would be run').


rule r2_tech_part_p2_q1
if prg_invite_ccr is tech_part_p2_q1 and
prg_invite_ccr = 'No'
then
write('We shall continue with manual interpretation through visual
inspection of the seismic snap').
rule r1_tech_part_p3_q3
if win_size is tech_part_p3_q3 and
win_size='window size 5 time intervals'
then
write('Crosscorrelation program with window size of 5 time
intervals is chosen...').


rule r2_tech_part_p3_q3
if win_size is tech_part_p3_q3 and
wind_size='window size 7 time intervals'
then
write('Crosscorrelation program with window size of 7 time
intervals is chosen....').
```

Table 6.12 Sample of rules based on analytical plots continued

```
rule r1_tech_part_p4_q2
if is_ccr_plot_identical is tech_part_p4_q2 and
is_ccr_plot_identical ='Yes'
then
write('We move on with manual interpretation  for other
horizons').


rule r2_tech_part_p4_q2
if is_ccr_plot_identical is tech_part_p4_q2 and
is_ccr_plot_identical = 'No'
then
write('There seems to be a break in the seismic snap which
is missing in the plot').


rule r1_tech_part_p4_q3
if break_in_snap_missed is tech_part_p4_q3 and
break_in_snap_missed='Yes'
then
write('The break seen in the snap is missing in plot').
```

```
rule r2_tech_part_p4_q3
if break_in_snap_missed is tech_part_p4_q3 and
break_in_snap_missed='No'
then
write('The sample may have to be redone, as similarity
between plot and seismic is missing and break is also
missing').


rule r1_tech_part_p4_q4
if use_inst_ph is tech_part_p4_q4 and
use_inst_ph='Yes'
then
write('Program using Instantantaneous phase attribute to
track break will be used').


rule r2_tech_part_p4_q4
if use_inst_ph is tech_part_p4_q4 and
use_inst_ph='No'
then
write('We shall resume with manual interpretation').


rule r1_tech_part_p4_1_q1
if see_ccr_plot is tech_part_p4_1_q1 and
see_ccr_plot='Yes'
then
write('Plot obtained after cross-correlation can be
viewed').
```

Table 6.12 Sample of rules based on analytical plots continued

```
rule r2_tech_part_p4_1_q1
if see_ccr_plot is tech_part_p4_1_q1 and
see_ccr_plot='No'
then
write('We may resume with manual interpretation').
```

Table 6.13 Sample of rulesets based on analytical plots

```
ruleset set_tech_part_p2_set1
contains r1_tech_part_p2_q1, r2_tech_part_p2_q2;
update ruleset by removing each selected rule;
when a rule misfires do true.
```

```
ruleset set_tech_part_p3_set1
contains r1_tech_part_p3_q3, r2_tech_part_p3_q3;
update ruleset by removing each selected rule;
when a rule misfires do true.


ruleset set_tech_part_p4_set2
contains r1_tech_part_p4_q2, r2_tech_part_p4_q2;
update ruleset by removing each selected rule;
when a rule misfires do true.


ruleset set_tech_part_p4_set3
contains r1_tech_part_p4_q3, r2_tech_part_p4_q3;
update ruleset by removing each selected rule;
when a rule misfires do true.


ruleset set_tech_part_p4_set4
contains r1_tech_part_p4_q4, r2_tech_part_p4_q4;
update ruleset by removing each selected rule;
when a rule misfires do true.


ruleset set_tech_part_p4_1_set1
contains r1_tech_part_p4_1_q1, r2_tech_part_p4_1_q1;
update ruleset by removing each selected rule;
when a rule misfires do true.
```

Table 6.14 Sample of actions generated during analytical interpretation

```
action main_tech_part_p2_a1;
do restart
and main_tech_part_p2_a2 and nl.


action main_tech_part_p2_a2;
do restart and
invoke ruleset set_tech_part_p2_set1 and
main_tech_part_p2_a3.
action main_tech_part_p2_a3;
if tech_part_p2_q1='Yes'
then
write(tech_part_p2_q1) and
```

```
write ('proceeding to execute program....')
else
write ('Lets continue with manual interpretation')
and
invoke ruleset set_set18
end if.


action main_tech_part_p3_a1;
do restart
and ask tech_part_p3_q1 and ask tech_part_p3_q2 and
invoke ruleset set_tech_part_p3_set1 and
main_tech_part_p3_a2.


action main_tech_part_p3_a2;
if tech_part_p3_q3='window size 5 time intervals'
then
write('The cross correlation program with window size 5
intervals shall be executed')
else
write('The cross correlation program with window size 7
intervals shall be executed')
end if.


action main_tech_part_p4_a1;
do restart
and main_tech_part_p4_a2 and nl.


action main_tech_part_p4_a2;
do restart
and invoke ruleset set_tech_part_p4_set2 and
main_tech_part_p4_a3.
```

Table 6.14 sample of actions generated during analytical interpretation continued

```
action main_tech_part_p4_a3;
if tech_part_p4_q2='Yes'
then
write('plot is identical to the horizon in snap') and
invoke ruleset set_set18
else
write('There could be a break seen in the snap which is missing in
the plot')
and invoke ruleset set_tech_part_p4_set3 and
main_tech_part_p4_a4
end if.


action main_tech_part_p4_a4;
if tech_part_p4_q3='Yes'
then
write('There is a break seen in the snap which is missing in the
plot')
and invoke ruleset set_tech_part_p4_set4 and
main_tech_part_p4_a5
else
write('The sample may have to be redone, because the seismic snap
and the plot are not identical
and at the same time even break is missing from the plot')
end if.


action main_tech_part_p4_a5;
if tech_part_p4_q4='Yes'
then
write('Program using instantaneous phase seismic attribute is being
called')
else
write('we may proceed with manual interpretation') and
invoke ruleset set_set18
end if.
action main_tech_part_p4_1_a1;
do restart
and main_tech_part_p4_1_a2 and nl.

action main_tech_part_p4_1_a2;
do restart
and invoke ruleset set_tech_part_p4_1_set1 and
main_tech_part_p4_1_a3.

action main_tech_part_p4_1_a3;
if tech_part_p4_1_q1='Yes'
then
write('plot will be displayed')
else
write('We will resume with manual interpretation')
and invoke ruleset set_set18
end if.
```

6.3 INTELLIGENCE SERVER

As mentioned in Chapter 5, the intelligence server is the package that allows the connection and communication between the front-end and 'Flex'. There are a set of files, which have been added to the directory holding the front-end. These work together, to make it possible to invoke the rule-base in 'Flex' through the front-end, finally leading to the interpretation.

6.3.1 Files added

Visual Basic, being an object-based language, allows for working with user-defined classes and objects. Intelligence server provides class definition of class in the form of a file by name, `'flex.cls'.` This file, when copied into the current directory holding the front-end, enables visual basic to create an object of 'Flex' class, call its methods and use its functionalities.

Two other files that have also been added are `utility.bas` and `int386w.bas`. The file `utility.bas` holds some utilities for 'Flex' to work in Visual Basic environment and the file `int386w.bas` holds the class definitions of intelligence server. These files when copied get stored into the front-end in the form of standard modules.

There are set of `.frm` files (the form modules) which also have been added. These are the dialog boxes which allow the 'Flex' questions to be displayed in a standard format in Visual Basic environment. The `edtbox.frm` is an edit box dialog which allows those questions to be asked which need the user to enter a text or numeric entry as their response. The `lstbox.frm` is a single choice selection dialog, which allows those questions to be asked which display a list options and allow the user to choose any one option as their response. The `mltbox.frm` is a multiple choice selection dialog, which allows those questions to be asked which display a list of options and allow the user to choose more than

one option as their response. Thus, these files allow for *questions* from 'Flex' to appear to the user, through Visual Basic interface.

There are a set of system files which have been added. One significant file is `flex.pc` which holds the flex run-time code. Other files are `int386w.sys`, `int386.ovl`, `int386w.dll` and `vint386w.dll`. The `int386w.sys` is the Prolog engine, `int386w.ovl` is the intelligence server runtime, `int386w.dll` is the intelligence server dynamic linked library and `vint386w.dll` is the visual basic interface to the intelligence server dynamic linked library.

The rules have been written in the 'Flex' files which have the extension '.ksl'. In the present case, the entire set of manual interpretation rules (over 70 in number) have been put into a single ksl file and the analytical interpretation rules have been put into 6 files with around 10-12 rules in each.

6.3.2 Methods of 'Flex' class

There are six methods (functions) `'LoadFlex()'`, `'InitGoal()'`, `'RunGoal()'`, `'ExitGoal()'` and `'HaltFlex()'`.

For the front-end to call the 'Flex' rule-base the following code has been written:

```
Dim flx as new Flex
```

This is the first step where an instance of the Flex class is being created and initialized.

Then the 'Flex' object *flx* is used to call the methods of 'Flex' class. `LoadFlex()` loads a 'Flex' instance into memory. `InitGoal()` is the

method which takes the name of the *.ksl* file as its input parameter and loads it. **RunGoal ()** method is given as its input parameter, the name of the *action* in the *.ksl* file which needs to be executed. **ExitGoal ()** method ends the execution after all the rules have fired and **HaltFlex()** unloads the 'Flex' instance from memory.

Figure 6.13 shows one of the interpretation forms. The Figures 6.14, 6.15, 6.16, 6.17 and 6.18 show the code snippets that have been typed in the form for invoking the interpretation.

Figure 6.14 shows instantiation and initialization of 'Flex' object '*flx*' in the *General Declarations* section of the interpretation form. Figure 6.15 shows the code written in the *click( )* event of the *'Call Expert System'* command button. Figure 6.16 shows the code written in the *load( )* event of the form. Figure 6.17 shows the code written in the *unload( )* event of the form. Figure 6.18 shows the code written in the *click( )* event of the *'Help'* command button of the form.
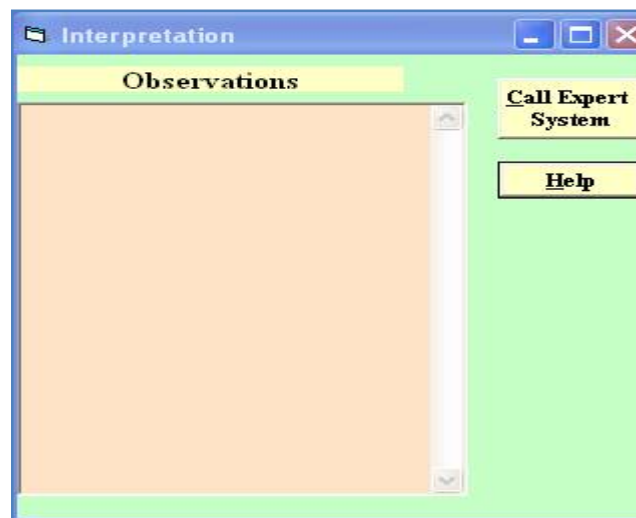

Figure 6.13 Interpretation form

```
General Declaration
Dim flx As New flex
```

Figure 6.14 *General declaration* **area showing 'Flex' object instantiation and initialization**

```
Private Sub CallExpertSystem_Click()
        txtObservation.Text = ""
    txtObservation.Text = flx.RunGoal("main. ")
                End Sub
```

Figure 6.15 Code in *click( )* **event of** *Call Expert System* **button**

```
Private Sub Form_Load()
            flx.LoadFlex ("")
flx.InitGoal("reconsult_rules( 'manual_int_flex_coding4.ksl' ). ")
            flx.CallGoal
            flx.ExitGoal
                End Sub
```

Figure 6.16 Code in *load( )* **event of form**

```
Private Sub Form_Unload(Cancel As Integer)
                flx.HaltFlex
                End Sub
```

Figure 6.17 code in *unload( )* **event of form**

```
Private Sub help_Click()
    MsgBox "This part of the module calls expert system to ask
questions to the user," + Chr(13) + " and solicits responses to
proceed with interpretation." + Chr(13) + "Press the Run button
            to start !", vbOKOnly, "Flexeg"
                    End Sub
```

Figure 6.18 Code in click event of Help button

6.4 DEPLOYMENT OF EXPERT SYSTEM

The expert system has been packaged, using *'package and deployment wizard'* feature of Visual Basic, into installable software. This wizard determines which files need to be distributed with the software, compresses them all into a cabinet file (*.cab file*), that acts like a setup file. This setup file can be used to install the software.

In the present case, the developed software has been packaged into a setup file by name *SeisExpert.cab*. A *readme.txt* file has been designed to hold the basic installation steps, and has been packed on the same CD as the software. SeisExpert can be deployed on to any windows environment along with its dependency files.