SEG-Y ROUTINE FOR PROCESSING

SEISMIC DATA

BY

TARIQ JAVED
MSc. (GEOPHYSICS)
FEBRUARY 2004

DEPARTMENT OF EARTH SCIENCES
QUAID-I-AZAM UNIVERSITY
ISLAMABAD.

CERTIFICATE

This thesis of Mr. Tariq Javed is accepted in its present from by the Department of Earth Sciences as satisfying the thesis requirements for M.S.c degree in Geophysics.

Recommended by

Mr. Anees Ahmad Bangish ______________________
(Supervisor)

External Examiner ______________________

Dr. Zulfiqar Ahmad ______________________
(Chairman)
Department of Earth Sciences
Quaid-i-Azam University
Islamabad.

Dedicated to ....

My PARENTS

Who always pray for me.

ACKNOWLEDGEMENT

In the name Allah, the beneficent, the merciful,
"All praise is to Allah, we glorify Him,
seek His help and forgiveness and turn to Him".
I bear witness that Holly Prophet Muhammad (PBUH)
is His last messenger whose life is perfect model for humanity.

I offe my heartfelt gratitude to my honorable & respected supervisor, Prof. Anees Ahmad Bangish, for his appropriate guidance and constructive criticism throughout my M.Sc I want to pay my deepest gratitude to my parents, my brother Assad and Nadeem and my sisters for their prayer, patience, encouragement and their moral and financial support. I must also be thankful To all my class fellows for their sincere help, co- orporation and encouragement , during my MSc. The most fervent thanks to all those relatives and friends Faisal, Adnan, Tanveer, Aamir, Shahid, Saeed, Tahir, Akbar, Aamir & especially to my senior Farhan and Farhat Ali whose works & assistance has been an invaluable Source of light & inspiration to me.

TARIQ JAVED

MatSeis

It is the combination of two words Mat and Seis means that seismic data processing in Matlab. .MATLAB integrates mathematical computing, visualization, and a powerful language to provide a flexible environment for technical computing and largely used in the world for the data processing.

MatSeis is a M-file that allows the user to open a seg-y file in Matlab The raw data in seg-y format from the field can be fed into the Matlab without any expensive pre-processing. Along with the opening of the file, two types of the filters (Butterworth & Chebyshev) are applied separately with different orders for checking the data processing in Matlab.

The MatSeis is inexpensive, easy to operate and can be used in the field on a hand held PC. The M-file has been developed in MATLAB and it can be run on Linux, Unix and also on window XP. This M-file gives you facility of 3-dimentional view of the data and the facility is provided to enhance the view by changing the coordinate axis and by changing the number of traces from the seg file.

This project is achieved in short period of time and coding is provided to understand and the further upgrading.

The Hardware and System Requirements:

To run MATLAB, you must have certain hardware and software installed on your computer. The system requirements include:

- Linux, Unix or later Microsoft Windows like window 2000 and window XP.

- 486DX/66 MHz or higher processor (Pentium or higher processor recommended), or any Alpha processor running Microsoft Windows NT Workstation.
- A CD-ROM disc drive.
- VGA or higher-resolution screen supported by Microsoft Windows.
- 16 MB of RAM for Windows 95/98, 32 MB of RAM for Windows NT Workstation.
- A mouse or other suitable pointing device.

CONTENTS

CONTENTS

CHAPTER 1 INTRODUCTION

CHAPTER 2 Seismic Data Processing

CHAPTER 3 FILTERING

CHAPTER 1

INTRODUCTION

Seismic method

Seismic methods are considered active geophysical methods in world's oil exploration. In seismic surveying, ground movement caused by some source is measured at a variety of distances from the source. The type of seismic experiment differs depending on what aspect of the recorded ground motion is used in the subsequent analysis. We do not mean to imply by this statement that any seismic experiment can be done from a given set of observations. On the contrary, the two types of experiments described below have very different acquisition requirements. These acquisition differences, however, arise from the need to record specific parts of the Earth's ground motion over specific distances.

One of the first active seismic experiments was conducted in 1845 by Robert Mallet, considered by many to be the father of instrumental seismology. Mallet measured the time of transmission of seismic waves, probably surface waves, generated by an explosion. To make this measurement, Mallet placed small containers of mercury at various distances from the source of the explosion and noted the time it took for the surface of the mercury to ripple after the explosion. In 1909, Andrija Mohorovicic used travel-times from earthquake sources to perform a seismic refraction experiment and discovered the existence of the crust-mantle boundary now called the *Moho*.

The earliest uses of seismic observations for the exploration of oil and mineral resources date back to the 1920s. The seismic refraction technique, described briefly below, was used extensively in Iran to delineate structures that contained oil. The seismic reflection method, now the most commonly used seismic method in the oil industry, was first demonstrated in Oklahoma in 1921. A plaque commemorating this event was erected on the site by the Society of Exploration Geophysicists in 1971.

1.11 Refraction Seismology

Refraction experiments are based on the times of arrival of the initial ground movement generated by a source recorded at a variety of distances. Later arriving complications in the recorded ground motion are discarded. Thus, the data set derived from refraction experiments consists of a series of times versus distances. These are then interpreted in terms of the depths to subsurface interfaces and the speeds at which motion travels through the subsurface within each layer. These speeds are controlled by a set of physical constants, called *elastic p parameters* that describe the material.

1.12 Reflection Seismology

In reflection experiments, analysis is concentrated on energy arriving after the initial ground motion. Specifically, the analysis concentrates on ground movement that has been *reflected* off of subsurface interfaces. In this sense, reflection seismology is a very sophisticated version of the echo sounding used in submarines, ships, and radar systems. In addition to examining the times of arrival of these, reflection seismic processing extracts information about the subsurface from the amplitude and shape of the ground motion. Subsurface structures can be complex in shape but like the refraction methods, are interpreted in terms of boundaries separating material with differing elastic parameters .

Each of these techniques has specific advantages and disadvantages when compared to each other and when compared to other geophysical techniques . For these reasons, different industries apply these techniques to differing degrees. For example, the oil and gas.

Industries use the seismic reflection technique almost to the exclusion of other geophysical techniques. The environmental and engineering communities use seismic techniques less frequently than other geophysical techniques. When seismic methods are used in these communities, they tend to emphasize the refraction methods over the reflection methods.

Seismic methods are based on measurement of the travel times of short pulses or wavelets introduced by an impulsive force usually at the surface of the earth. The wavelet describes the perturbation of a representative point as a force is applied to an elastic medium of a given density. A sound wave in air or water is an example of such a process and the familiar echo sounding used on boats to determine water depth is perhaps the simplest application of a "seismic" method. The velocity of such an elastic pulse or wave is proportional to the elastic module (bulk and shear modules) and inversely proportional to the density. Seismic waves reflect off boundaries between media with differing density-velocity products and they refract at boundaries separating media of different velocities following Snell's Law (which was developed for the refraction of light).

It is observed that the velocity of seismic waves generally increases with depth on both a local and global scale. This means that a seismic wave propagating from a surface source will refract at interfaces so as to come back to the surface. This phenomenon was first discovered in earthquake seismology and led to the current picture we have of the internal velocity structure of the Earth. On a shallower scale the same behavior was employed to locate large guns in World War One. In the 1920s refraction methods were used to locate high velocity salt structures in the Gulf Coast region. A simple schematic of refraction paths (the path along which a seismic wavelet propagates) for a simple layered model and for a salt dome is shown in Figure 1.1

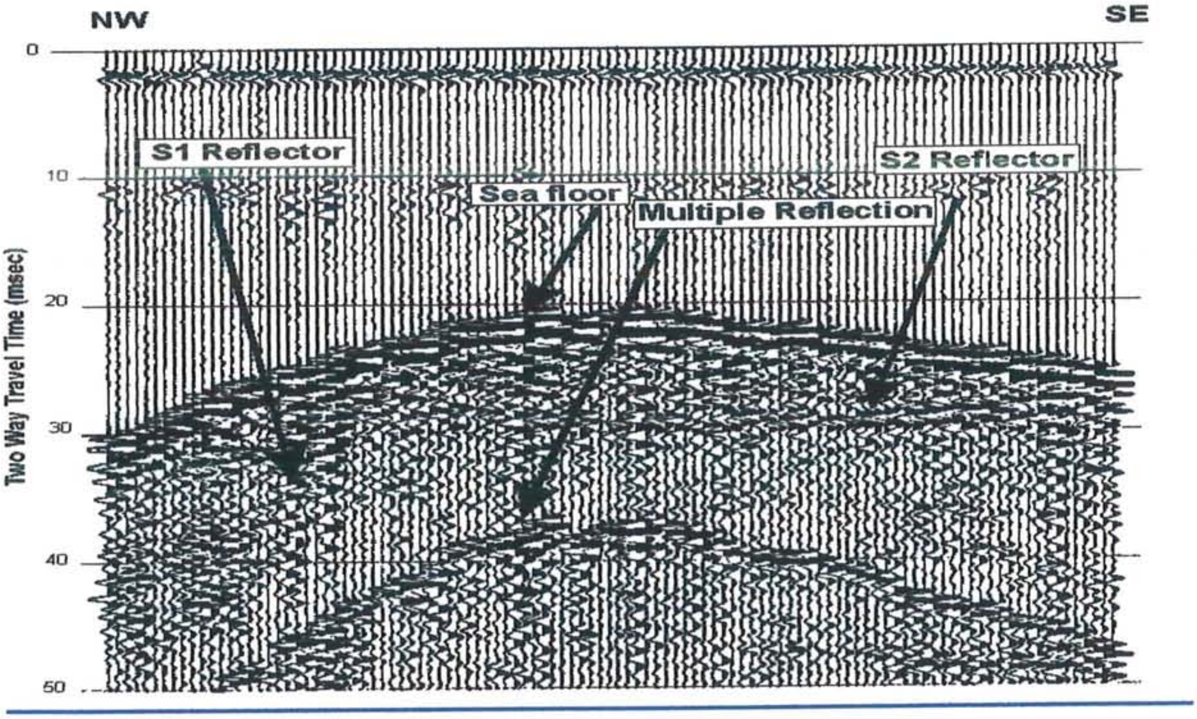


Figure 1.1

Seismic refraction is defined as the travel path of sound wave through an upper medium and along an interface (at a critical angle) and then back to the surface as shown in the figure below. The acoustic waves, like light waves, follow Snell's Laws of Refraction.

Seismic refraction surveys are commonly used to determine the thickness of unconsolidated materials overlying bedrock (overburden thickness) and depth to the water table. They are used for characterizing the geological framework of ground-water contamination studies and for assessing geologic hazards and archaeological studies.

Reflections from several layer boundaries to an array of receivers (geophones) are shown. This whole array is advanced along a profile so that essentially continuous reflection mapping is achieved.

. Reflections from several layer boundaries to an array of receivers (geophones) are shown in figures 1.2(a) and 1.2(b). This whole array is advanced along a profile

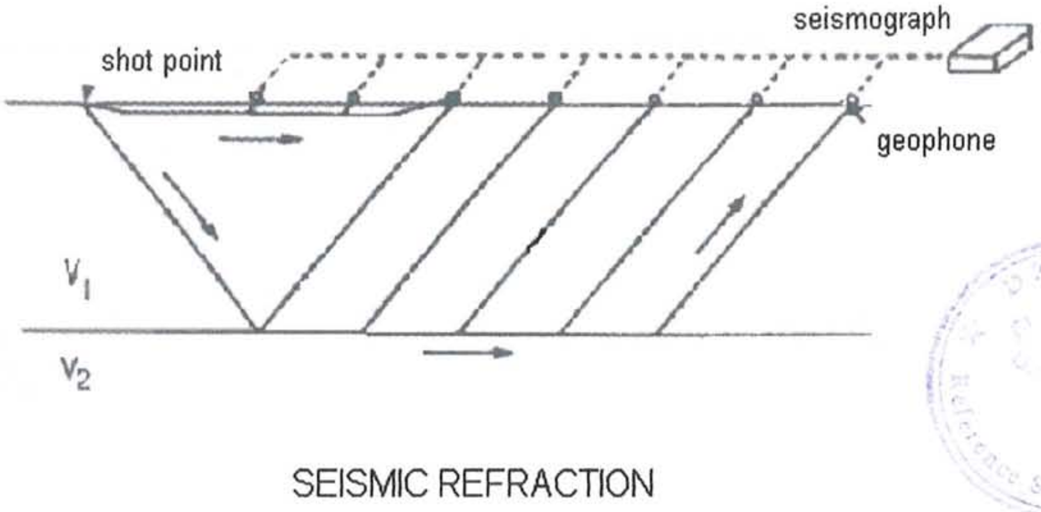


Figure 1.2(a)

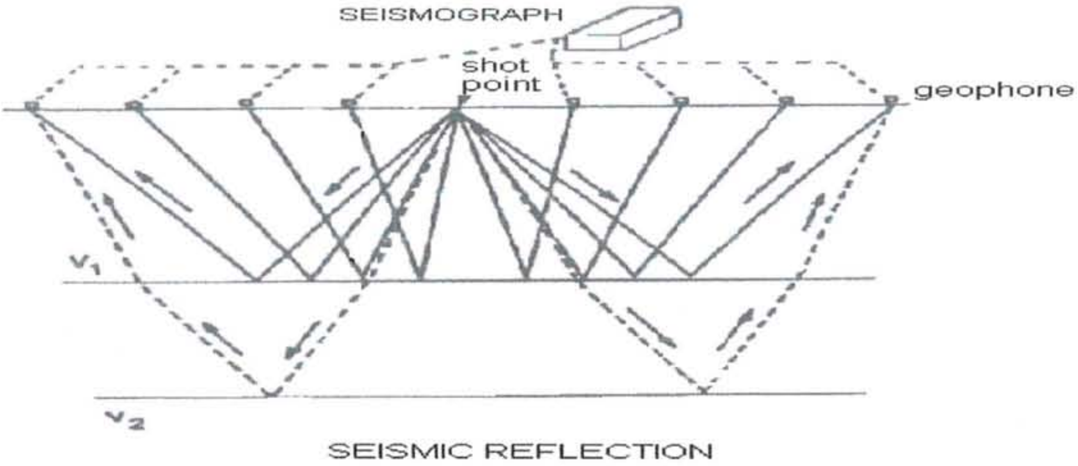


Figure 1.2(b)

AGS uses seismic refraction and reflection methods to map geologic and hydro - geologic features in the subsurface. The data can be acquired on land and in a marine environment. AGS uses seismic techniques to determine:

- Overburden of Landfill Thickness
- Bedrock Configuration in Two or Three Dimensions
- Stratigraphic or Lithologic Mapping
- Dip and Orientation of Geologic Layers
- Fault, Fracture, Weathered, and Shear Zone Presence
- Continuity of Confining Layers
- Depth to Groundwater
- Elastic Properties of Sediments and Bedrock
- Pathways of Groundwater Flow
- Void and Sinkhole Detection
- Vibration Monitoring

The seismic method is based upon the transmission of seismic waves into the subsurface and the recording, measurement, analysis, and interpretation of the resulting waveforms. Detailed cross-sectional images are generated that provide information on the depth and thickness.

1.2 Data acquisition systems

Seismic data acquisition units perform a number of functions:

1. Transmit the data from the recording devices to a central acquisition system. Many modem systems are called telemetry acquisition systems, because they transmit the data via radio or microwave signals to the recording unit.

2. Convert analog signals from the recording devices to digital form. Two parameters are important here: the sampling interval, and the digital precision (number of bits). The sampling interval is the time between successive digital samples. The digital precision represents the largest integer number that can be represented.

Another way to express this is the system dynamic range, which is the ratio of the largest to the smallest nonzero number. The dynamic range is usually expressed using decibels (dB), which is 20log10 of the amplitude ratio. For example, the dynamic range of a 24-bit system is 138.5 dB.

1.21 Seismic sources

1. Sledge Hammer:

Advantages:

Low cost

Portable

Can stack until desired SNR is achieved.

Disadvantages:

Poor signal penetration (max ~ 20 m)

Moderate repeatability

2. Accelerated Weight Drop (Elastic Wave Generator):

Moderate cost and portability.
Better signal penetration than hammer, but less than dynamite or vibroseis.
Can stack until desired SNR is achieved

3. Dynamite:

Advantages:

Best signal penetration of any land source. As a result, used extensively in hydrocarbon exploration, as well as in crustal studies.

Disadvantages:

Heavily regulated - strict limits on handling, minimum distance from homes, bridges, pipelines, etc.
Pre-drilling of shot-holes is expensive.
Not repeatable.

4. Air guns:

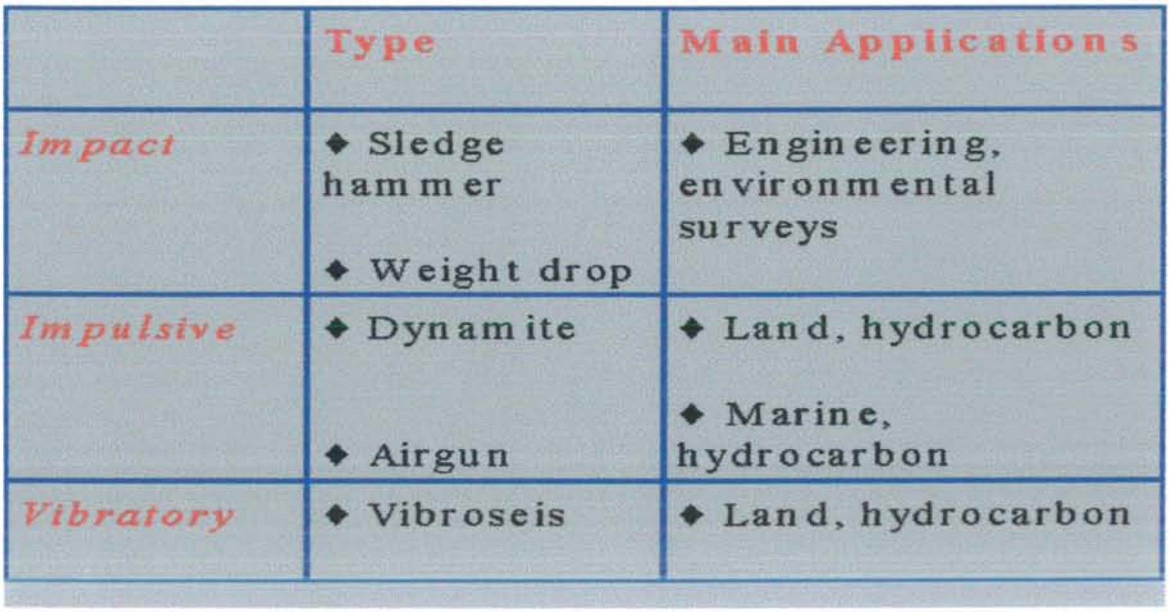
Air guns produce seismic waves by rapidly expelling a bubble of compressed air into the water. Air guns are now used in the vast majority of marine seismic surveys.
The air bubble undergoes an oscillatory pattern of expansion and contraction. Tuned air gun arrays are used to suppress this bubble pulse.
Signal penetration for air gun arrays approaches that of dynamite.

5. Vibroseis:

The vibroseis method is based on radar "chirp" technology.
Vibroseis vehicles are equipped with a large pad, a heavy reaction mass and hydraulic systems.

With the full weight of the vehicle on the pad, vibroseis units impart an oscillatory chirp into the ground.

Cross-correlation techniques are applied during data processing to simulate the response of an impulsive source.



	Type	Main Applications
Impact	◆ Sledge hammer ◆ Weight drop	◆ Engineering, environmental surveys
Impulsive	◆ Dynamite ◆ Airgun	◆ Land, hydrocarbon ◆ Marine, hydrocarbon
Vibratory	◆ Vibroseis	◆ Land, hydrocarbon

1.22 Seismic detectors

- **Geophones (natural frequency, critical damping)**

A simple device that measures the velocity of ground motion. It consists of a magnet suspended by springs inside a wire coil . The coil is affixed inside a rugged case, which usually has a spike at the base to provide optimal ground coupling. Often, the coil is suspended instead of the magnet.

Movement of the geophone case produces motion of the magnet relative to the coil. By Faraday's Law, this induces a current flow in the coil that is proportional to the velocity of the ground

- **Hydrophones (basic principles)**

1.23 Seismic recording devices

- Telemetry systems
- Analog to digital conversion
- dB scale

1.3 PROCESSING FLOW - OVERVIEW

1.31 Demultiplex
1.32 Trace Editing
1.33 Gather
1.34 Velocity analysis
1.35 Normal Moveout correction (Tnmo)
1.36 Mute
1.37 Statics Corrections
1.38 Stack
1.39 Migration
1.40 Depth Conversion

1.31 Demultiplex

It is equivalent to performing a matrix transpose. Demultiplexing in the geophysical sense is the unscrambling of multiplexed field data to trace sequential form .to the processor, demultiplex has come to mean a whole family of processing function that effect , or are affected by specific process of demultiplexing .Demultiplexing can be explained in the following six major steps :

A tape dumps and format determination
B Preliminary checks
C Binary gain recovery
D Vibroseis Correlation
E Summing
F Diversity stacking.

1.32 Editing

Editing is the process of removing or correcting any trace or records which, in their originally recorded form, may cause a deterioration of the stack. Individual traces may be affected by polarity reversals or by noise. Entire records may be contaminated by coherent noise or rendered unusable by misfires or auto fires. Trace inspection and editing is the step that require the most analysis time but it is critical to the development of clean good quality seismic sections. Several forms of editing are use throughout data processing which is categorized:

A Editing during demultiplex

B Muting

1.33 Gather

In the gather process, traces from different shot gathers are rearranged as common midpoint (CMP) gathers. The three traces in figure 1.3 have the same CMP gather, taken through the processing steps in subsequent illustration, comprises one 3-fold trace on the stacked seismic profile.

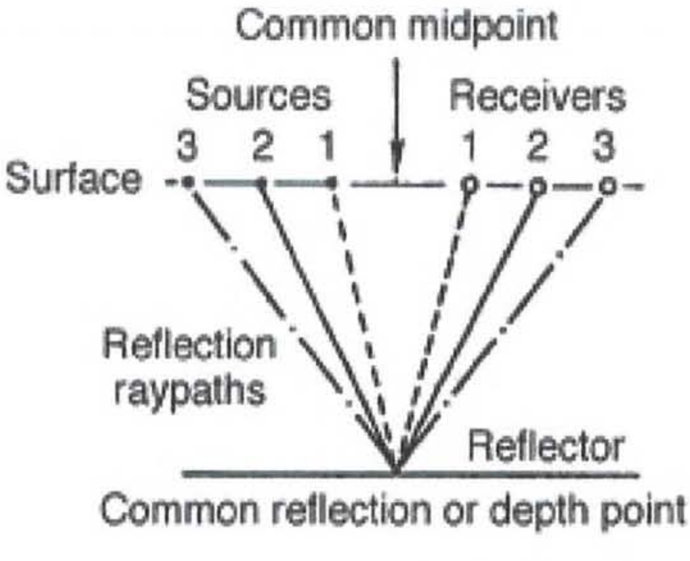


Figure 1.3

1.34 Velocity analysis

A reflection from a horizontal interface fallows a hyperbola according to :

Tt= to+Tnmo

Where

Tt= travel time from source to interface, to receiver

t o = T-axis intercept (time directly down to interface, straight back up to shot location)

Tnmo=hyperbolic increases in time with increasing distance from the source.

Velocity analysis determines velocities that best tune" primary reflections when traces are stacked. The process is commonly trial and error, whereby different normal moveout time (Tnmo) corrections are applied to traces in CMP gathers. A desired velocity removes Tnmo so that events that originally followed hyperbolas align horizontally. Upon stacking, primary reflections add in phase(constructively), while different types of noise add out of phase (destructively); the analysis thus yields stacking velocities.

1.35 Normal Moveout (Tnmo) Correction

1.36 NMO correct ions are determined not just for one event, but for several prominent reflections in a CMP gather. For a given t o and velocity, events originally falling along hyperbolas align after Tnmo corrections. Commonly, a deeper reflection is corrected for Tnmo with a higher velocity [Vrms(2)] than that used for a shallower Figure 1.4

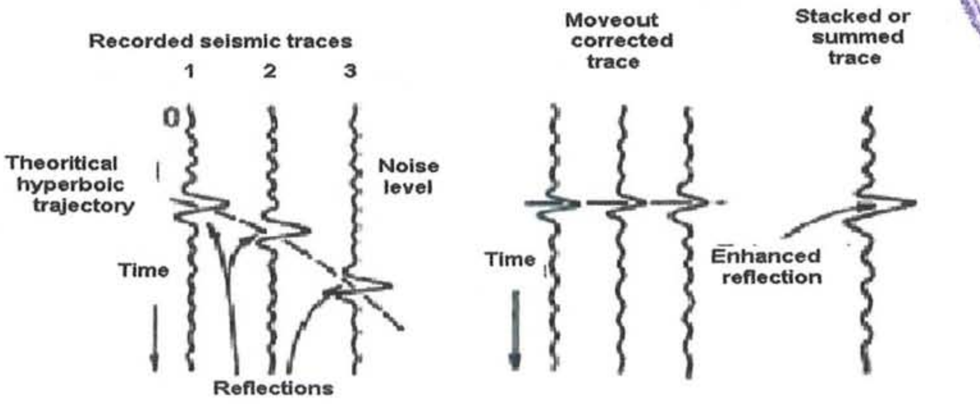


Figure 1.4

1.36 Mute: At distance from the source, the tops of records commonly have unwanted noise (direct P and S refractions, events distorted by Tnmo corrections), with few or no reflections. Rather then stack this noise together with reflections recorded near the source, a (commonly triangular) region of CMP gather traces are set to zero amplitude, or muted . On the tops of actual seismic reflection profiles, artifacts of muting can be seen as V-shaped "valleys" in regions where only far offsets comprise the stack traces.

1.37 Static Corrections:

On land, seismic source and receiver stations follow the topography. Corrections must be made for time delays at stations of high elevation, relative to lower stations. Time is added or subtracted to traces within CMP gathers, according to 1: the estimated near-surface velocity; and 2: the source and receiver elevations relative to a horizontal datum. After these Statics corrections are made, Tnmo corrected reflections should be more in Phase.

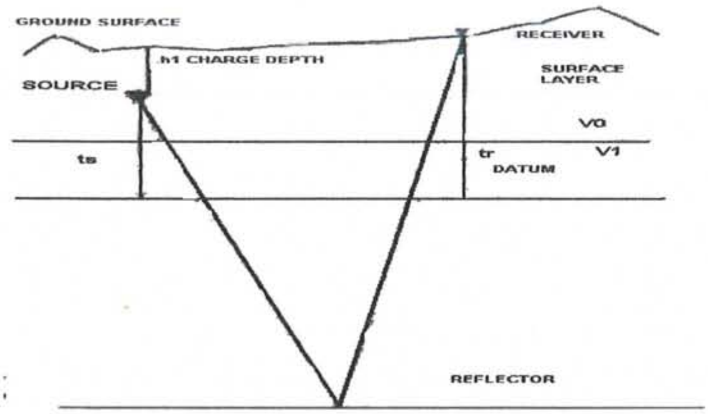


Figure 1.5

1.38 Stack:

After undergoing Tnmo, mute, and Statics corrections CMP traces are added (stacked) together. The fold of stack refers to the number of seismic traces that are combined to make one trace. Stacked traces are displayed side-by side to make an unmigrated section. The display mimics the situation that would result if each seismic trace were recorded at a common source/receiver position; thus, an unmigrated time section is also referred to as a "normal incidence" section.

1.39 Migration:

In two dimensions a reflected event could have come from any position along a seismic circle through the event, centered about the common source/receiver location. Migration spreads event along the potential locations on the semicircle; events on adjacent traces will add in phase at the true position of the reflector out of phase away from the true position. A migrated time section thus attempts to move event to their true horizontal positions relative to common source/receiver positions on the surface.

1.40 Depth Conversion:

The relative positions of events are distorted on time sections because of vertical and lateral changes in velocity. If velocities are well enough, the section can be converted from a vertical scale in two way travel time to time to a depth scale.

1.50 INTERPRETATION OF SEISMIC REFLECTION PROFILES

1.51 SEISMIC REFLECTION INTERPRETATION:

An explanation of subsurface conditions that led to the reflection of seismic waves back to Earth's surface.

Seismic reflection profiles are used extensively to map geological structures in the subsurface. They are especially useful when reflections can be tied to firm constraints provided by surface outcrop and drilling. Even when not strictly constrained, reflection geometries when provide information on gross structural form within the crust. Seismic profiles are thus valuable tools to unravel the tectonic history of a region, and to compare and contrast features associated with continental margins and plate boundaries.

Through resembling geological cross sections, seismic reflection profiles must be interpreted wisely, with appreciation for both the utility and pitfalls of the method.

1.52 APPERENCE OF STRUCTURES ON REFLECTION PROFILES

Patterns of events on seismic reflection profiles provide constraints on subsurface structure. Structural interpretation first requires appreciation of how simple geometries would appear on different types of seismic section displays. A common display is the unmigrated time section; the data may be processed further to yield migrated time and migrated depth sections.

An unmigrated time sections an attempt to display reflection data as if the source and receiver for each seismic trace were at the same surface position Figure 1.6

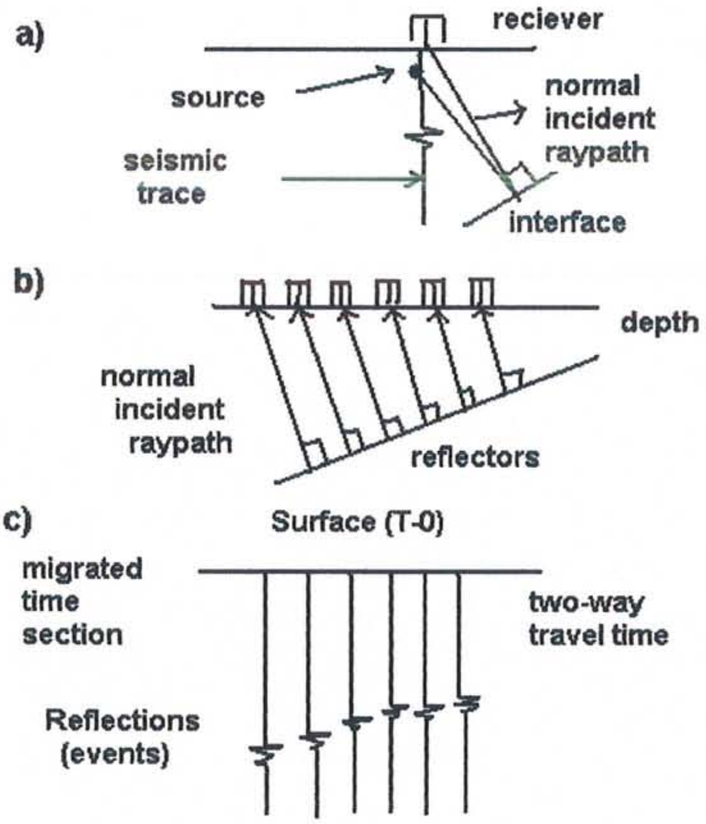


Figure 1.6

This situation requires that each seismic ray recorded at the surface be reflected normal (at a 90 degree angle) to an interface below the surface. An unmigrated seismic section is thus referred to as a normal incidence section. A reflection (event) appears directly below the position of the common source/receiver position, even if the reflection was not from directly below. The event appears at the time it takes the ray to travel to the reflecting interface, then back to the surface (two-way travel time).

Migration moves seismic events back to their true positions, as if each ray went straight down then back up. Figure 1.7

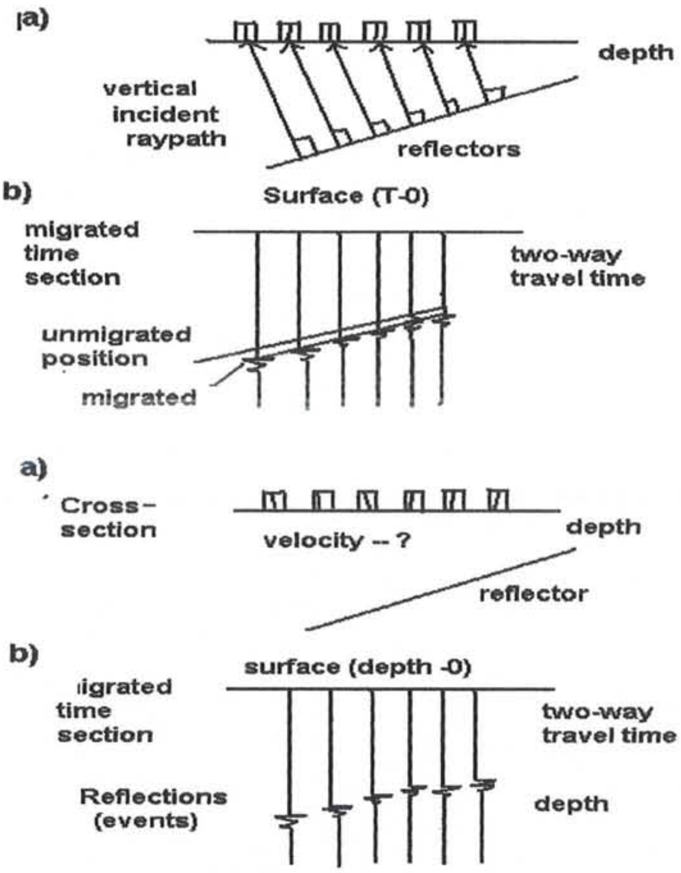


Figure 1.7

A migrated time section is thus referred to as a vertical incidence section. After migration the vertical scale can be converted from two-way travel time to depth, yielding a migrated depth section figure 1.7. (last a,b) For an average velocity(V) above the interface, the conversion of a point from two-way travel time(T) to depth(d) is:

$$d = (T/2) * V$$

Migration and depth conversion require accurate knowledge of the velocity and travel paths of seismic energy. Through interpretation requires viewing not just migrated time and depth sections, but also seeing how events appear before migration and depth conversion.

Four factors must be taken into account when viewing seismic profiles.

1. Geometric effects are due to reflections from interfaces that are not horizontal. Migration of the data attempts to put the events back into their proper orientation .

2. Velocity effects cause time shifts of some reflectors relative to others. A common example is "velocity pull-up," whereby events below high-velocity material appear shallow on time sections .Proper depth conversion removes velocity effects.

3. Raypath bending distortions occur as seismic waves penetrate complex structures. The bending is caused by changes in velocity across interfaces according to the Snell's law. The fourth effect is listed below

1. Geometric "Migration" Effects:

An unmigrated section has events plotted directly below the common source/receiver position, even through the actual reflecting point might not have been directly below, figure 1.8 The actual reflecting point could have been any point below the surface that is the same travel time away from the receiver. Thus, the event appearing on the seismic trace could have been from anywhere on the hemisphere (semicircle for two-dimensional structure) drawn through the event Figure 1.8 (a).

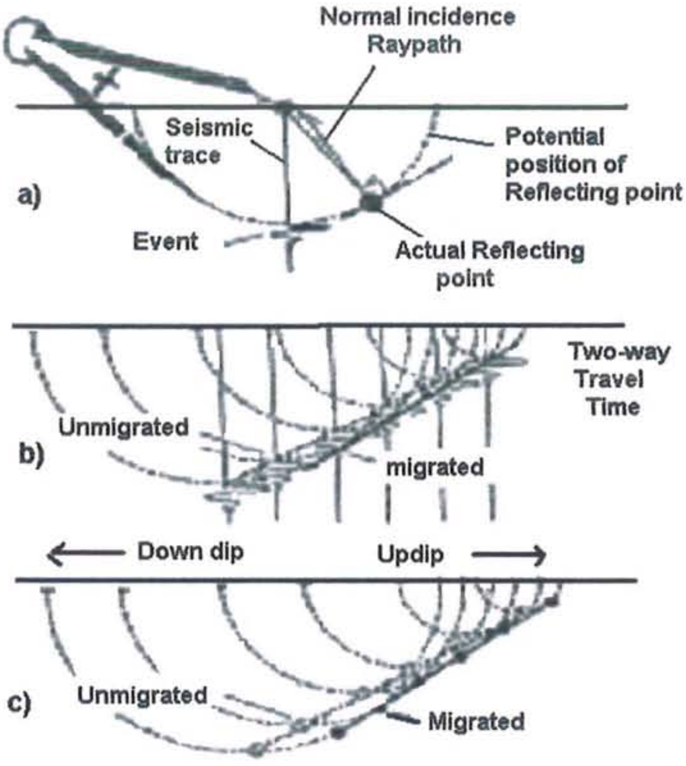


Figure 1.8 (a, b, c)

Migration of the linear reflection segment can be envisioned as swinging arcs through points along the event. The migrated position of the segment is defined by the line segment running tangent to each of the arcs above figure (1.8.b).Notice the relationships of the unmigrated and migrated positions of the event above figure 1.8.c. each point of the unmigrated event is downdip from the corresponding point of the migrated event, unmigrated points are deeper than corresponding migrated points, the unmigarated event is less steep than the migrated event, the unmigrated event is longer than the migrated event. Migrating a seismic profile thus tends to make reflecting segments move updip, shallow, steepened and become shorter. We have the examples like

i. Dipping Interface Connecting Two Horizontal Interfaces
ii. Very Small Sphere(Point Source Diffraction)
iii. Anticline

iv. Syncline

v. Series of Anticlines and Synclines

2. **Velocity Effects.**
3. **Raypath Bending.**
4. **Combination of Effects 1, 2 and 3.**
5. **Three Dimensional ("Sideswipe") Effects.**

CHAPTER 2

SEISMIC DATA PROCESSING

Seismic Data Processing

2.0 Introduction

Data Processing is an approach by which the raw data recorded in the field is enhanced to the extent that it can be used for the geological interpretation. To bring meaning to the recorded digital data, computer-based data processing is used. From ten to twenty programs are usually used in a processing sequence, and these are selected from a library of several hundred programs. Although many programs will be almost identical with other programs. For example one program may operate in time domain while other works in the frequency domain, yet they may yield almost similar results.(Zia-ur-Rehman,1989)

The selection of processing sequence for a given set of field data depends upon the following:

- Intrinsic Quality of Raw Data.
- Geological Environment.
- Personal preference.
- Cost.

According to Yilmaz (1987) there are three primary stages in processing seismic data. In there usual order of application they are

i Deconvolution

ii Stacking

iii Migration

2.1 BASIC STEPS INVOLVED IN SEISMIC DATA PROCESSING

Generally there are five basic steps in Seismic data processing shown below in tables(2.1 to 2.4)

1. Data Reduction.
2. Geometric Corrections.

3. Data Analysis and Parameter Optimization.
4. Data Refinement.
5. Data Presentation and Storage.

Some procedure is common to every processing sequence; others are used only occasionally or rarely. Some processes always come at the same stage of the sequence; others can be used in any of several stages. Some may be used more than once.

There is no standard processing recipe for all types of data.

DATA REDUCTION

PROCESS	PURPOSE	WHEN APPLIED
Demultiplexing	Put data into trace sequential order.	First
Cross Correlation	Vibroseis source energy compression.	In Demultiplex run .
Gain Recovery	Multiply data by binary gain codes	Second if required.
Editing	Remove bad records.	Third and also during processing otherwise.
Gain	Amplify weaker events	In the end of data Reduction and afterwards.

Table 2.1

GEOMATRIC CORRECTIONS

PROCESS	PURPOSE	WHEN APPLIED
Trace Gather(CDP Sort)	Arrange traces according to depth point	After data reduction or prior to stacking.
Static Correction	Vertical time correction.	Correct to at least a floating datum before NMO. May correct to final datum after stack.
Dynamic	Horizontal Time Correction	Before Stacking

Table 2.2

DATA ANALYSIS AND PARAMETER OPTIMISATION

PROCESS	PURPOSE	WHEN APPLIED
Band Pass Filter	Attenuate noise-having frequencies outside of signal band.	Best before stack and before NMO.
Notch Filter	Attenuate narrow frequency band noise with signal band.	Best before stack.
2-D Filter	Spatial band pass filter, attenuate random noise or events separable by dip.	Any time after data reduction, depending on type of events to be removed.

Deconvolution	Compress into pulse shape, attenuate reverberations.	Best before stack and before NMO.
Velocity Analysis	To optimize stacking velocity functions.	Preferably on deconvolved CDP fathers, after datum Statics applications.

Table 2.3

DATA REFINEMENT

PROCESS	PURPOSE	WHEN APPLIED
Mute	Deadens high amplitude noise spikes.	Before stack.
Stack	Attenuate random and other types of coherent noise.	After all geometric corrections and mute.
Residual Statics	Refinement of static corrections	Before final stack
Migration	Move events to true spatial position relative to short and receiver stations.	After stack, in some cases before stack.

Table2.4

2.2 DATA REDUCTION

Data reduction is the first and most important step in data processing flow. It includes following four categories of software programs:

1 Demultiplexing

2 Display

3 Editing

4 Amplitude Adjustment .

2.21 RECORDING REVIEW

Seismic field recording systems are designed to record data from several channel once. A typical seismic record consists of 24, 48, 96 data channels recorded simultaneously along with some auxiliary information channels. Yet a digital magnetic tape recorder is, in effect, a single channel device. It is therefore necessary to "multiplex" the channel before writing them on tape. The multiplexer scans data channels and write a sample for each channel in a digital format. It then repeats the cycle as many times as is necessary. On tape the beginning of each scan is mark with a "sync code" and at the very beginning of the record a "header" is written. The header marks the beginning of the record and also contains recording information such as gain and filter settings. The time between the beginning of a scan and the beginning of the next scan is the "sample interval"(often called sample rate). Figure 2.1 shows the more detailed diagram of multiplexed tape. There are seven channels multiplexed, channel 1 is the sync channel and channel 2 is the sync channel and channels 2 through 7 are data channels. It can be seen that the first scan consists of the first sample from each channel, and the second scan consists of the second sample from each channel, and so forth. In the multiplex format all of the channels are together in an orderly fashion. In order to process the data we must unmixed the samples, i.e. Demultiplex the data. Again Figure 2.1 shows the format of the data after Demultiplex. The data are now in trace order, i.e all the sample for each trace are together .Each trace begin with the trace header, which contains identification and information about the trace, followed by the data samples. The traces are separated by tape gaps which are areas of tape on which nothing is written. It would be possible to plot the samples of a trace graphically and recover the original waveform.

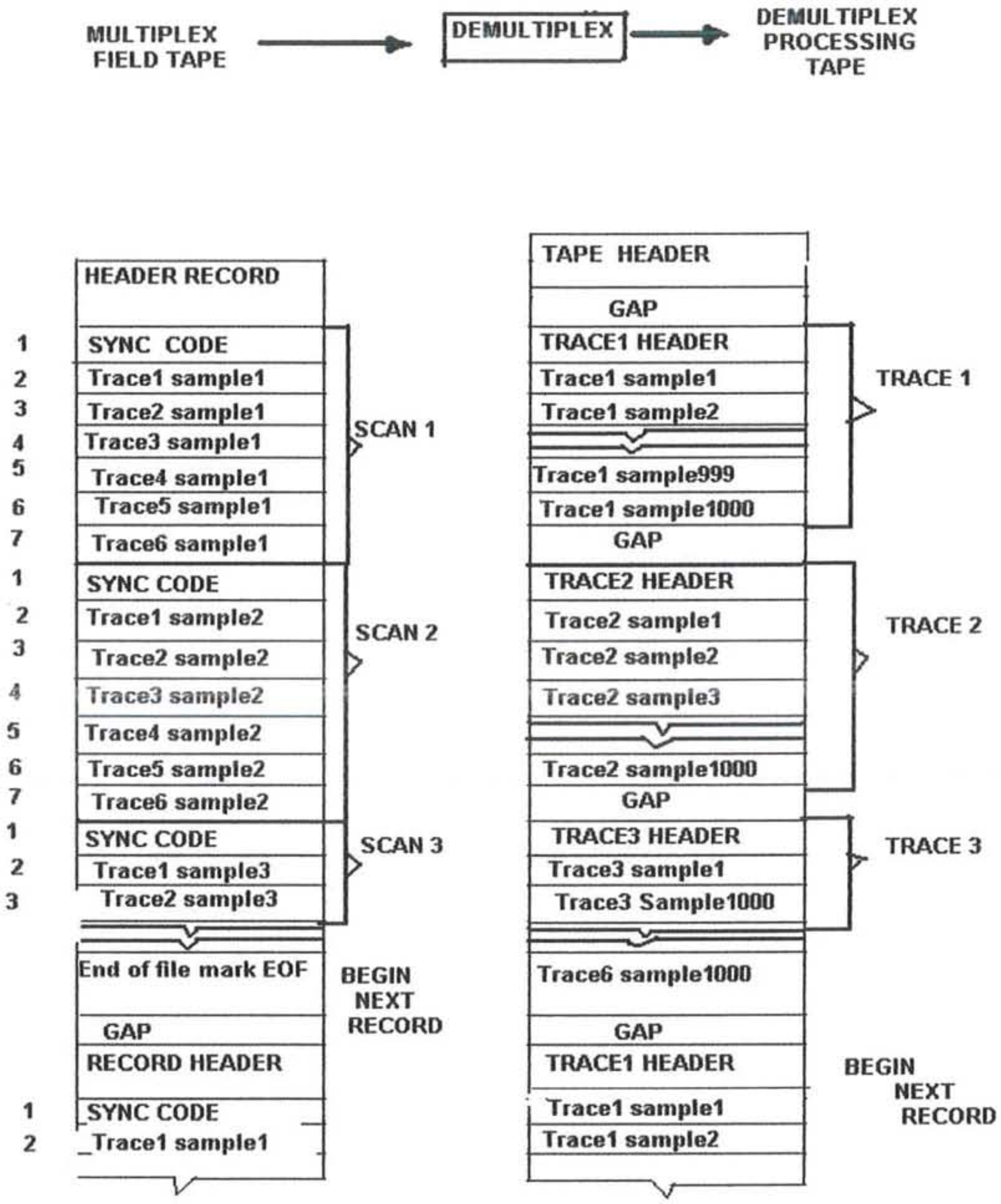

MULTIPLEX FIELD TAPE
DEMULTIPLEX
DEMULTIPLEX PROCESSING TAPE
HEADER RECORD
1 SYNC CODE
2 Trace1 sample1
3 Trace2 sample1
4 Trace3 sample1
5 Trace4 sample1
6 Trace5 sample1
7 Trace6 sample1
SCAN 1
1 SYNC CODE
2 Trace1 sample2
3 Trace2 sample2
4 Trace3 sample2
5 Trace4 sample2
6 Trace5 sample2
7 Trace6 sample2
SCAN 2
1 SYNC CODE
2 Trace1 sample3
3 Trace2 sample3
SCAN 3
End of file mark EOF
GAP
RECORD HEADER
BEGIN NEXT RECORD
1 SYNC CODE
2 Trace1 sample1
TAPE HEADER
GAP
TRACE1 HEADER
Trace1 sample1
Trace1 sample2
Trace1 sample999
Trace1 sample1000
TRACE 1
GAP
TRACE2 HEADER
Trace2 sample1
Trace2 sample2
Trace2 sample3
Trace2 sample1000
TRACE 2
GAP
TRACE3 HEADER
Trace3 sample1
Trace3 Sample1000
TRACE 3
Trace6 sample1000
GAP
TRACE1 HEADER
Trace1 sample1
Trace1 sample2
BEGIN NEXT RECORD

Figure 2.1

a). DEMULTIPLEX

According to Yilmaz, (1987) demultiplexing is the transposing of a big matrix in such a manner that the columns of resulting matrix can be read as seismic traces recorded at different offsets with common shot-points.

Demultiplexing in the geophysical sense is the unscrambling of multiplexed field data to trace sequential form. To the data processor, Demultiplex has come to mean a whole family of processing functions that effect, or effected by the specific process of demultiplexing. Demultiplexing will be explained in the following six major steps:

I. Tape dump and format determination

II. Preliminary checks

III. Binary gain recovery

IV. Vibroseis Correlation

V. Diversity stacking

I. TAPE DUMP AND FORMAT DETERMINATION

A tape dump is a printed listing of the data on tape. A tape dump could be printed in binary but this would generate a large quantity of paper. It is much more convenient to convert binary numbers to "hexadecimal" for listing purposes. Hexadecimal is a base 16 number system using the characters 1,2,3,4,5,6,7,8,9,A,B,C,D,E and F . Four bits are required to express one hexadecimal number.

The hex characters are divided into groups of four for convenience. Each group then is a 16-bit word. There are sixteen groups or words per printed line. Figure 2.2 shows the same tape dump with notations. This is a dump of fixed point field data recorded on a 26-channel system. There are twenty four data channels, one sweep channel and one sync channel. The sync code is FFFF in hex, which is sixteen bits, all ones. As we can see in Figure the first twelve words in the record are header information. The first sync code marks the beginning of the first data scan. If we count the sync code as channel 1, then channels 2 through 13are data traces 1 through 12. Channel 14 contains the vibroseis sweep trace. Channels 15 through 26 are data traces 13 through 24. The next word is another sync code which marks the beginning of the next scan.

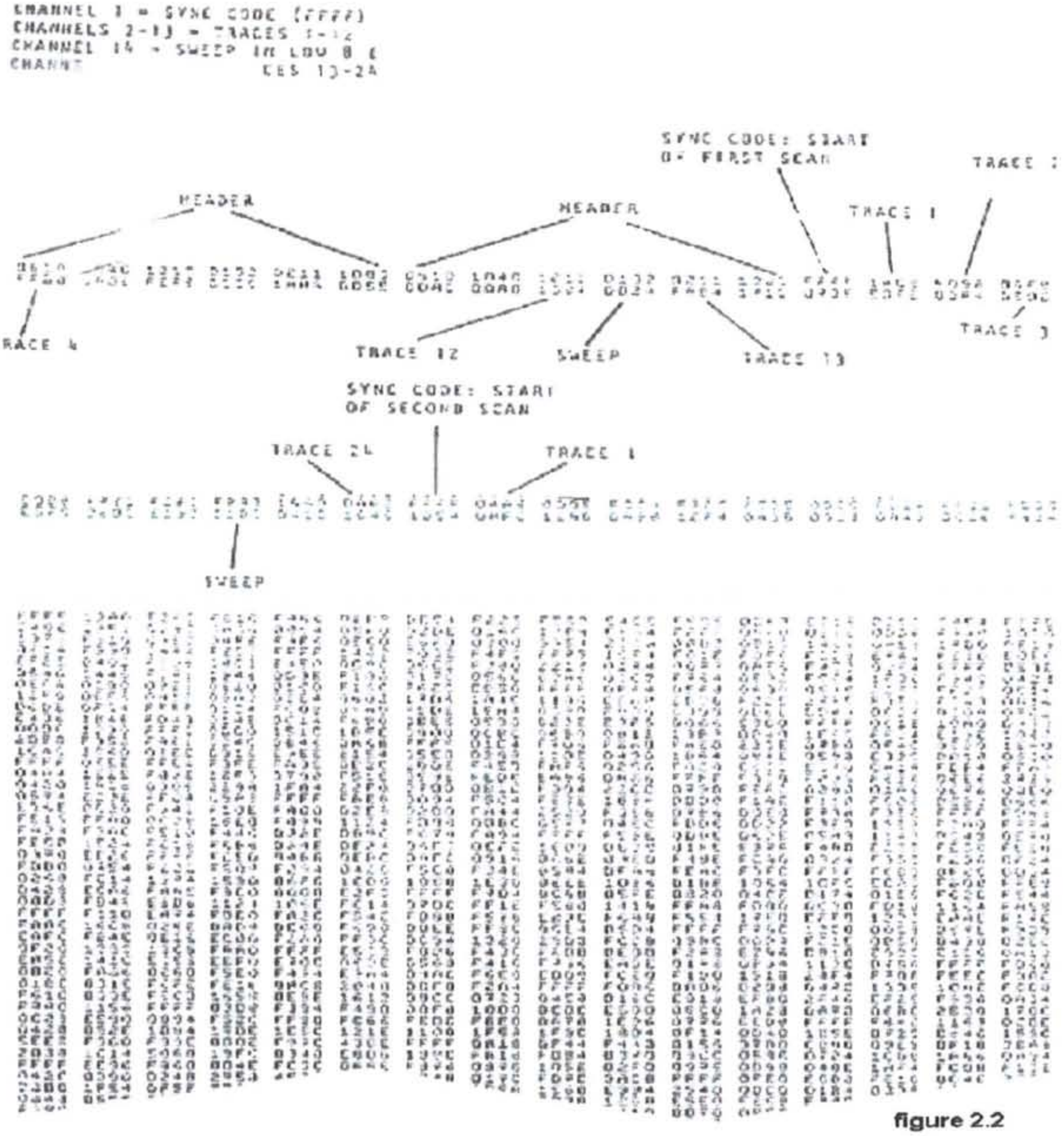

CHANNEL 1 = SYNC CODE (FFFF)
CHANNELS 2-13 = TRACES 1-12
CHANNEL 14 = SWEEP IN LOW 8 B
CHANN CES 13-24
SYNC CODE: START OF FIRST SCAN
TRACE 2
HEADER
HEADER
TRACE 1
TRACE 4
TRACE 12
SWEEP
TRACE 13
TRACE 3
SYNC CODE: START OF SECOND SCAN
TRACE 24
TRACE 1
SWEEP

figure 2.2

II. PRELIMINARY CHECKS

The processor has a tape format sheet, a hex-dump, a rim time printout from the computer and display of demultiplex data . With the help of this data following field tape errors can be diagnosed and removed.

1 Sync Errors

A sync code is placed at the start of each data scan during field recording. This code tends to vary with each format. A sync error occurs when the Demultiplex program cannot find a sync code. The number and the location of sync errors in the test run should be noted. If there are one or two sync errors on an occasional record, there is probably no cause for alarm. If 5-10 sync errors per record occur in shot data, a serious problem is indicated.

2 RATE ERRORS

A rate error occurs when the multiplexed data is coming to fast from the magnetic tape to the disk. This normally occurs when the wrong format has been specified or there are too many zeros ahead of the first sync frame.

III. BINARY GAIN RECOVERY

Modern digital seismic instruments use various gain ranging techniques to increase their dynamic range. This is necessary because of the very rapid decay of seismic signal strength with time Figure 2.3 shows the output of a geophone array when an impulsive source is used. Gain ranging instruments apply a gain to this signal before recording it and also keep a record of what gains were applied. Since the gains applied to signal are recorded along with the signal, the processing center can "de-gain" the data, that is, remove the gain applied by the field instruments. Basically a gain function is a smoothly time varying, gain applied to the data to bring the overall amplitude up to a fairly constant level. This is referred to as gain recovery.

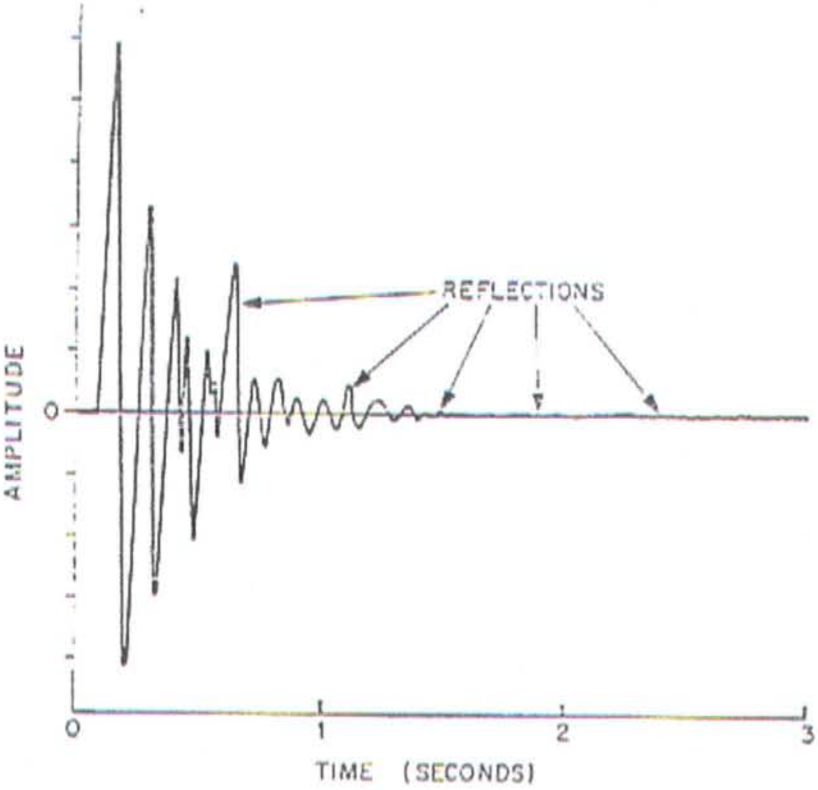


Figure 2.3

IV. VIBROSEIS CORRELATION

Most seismic methods use an energy source that generates a very short pulse. When these pulses are reflected and recorded they can be used directly to examine the subsurface structure. The signal generated in the vibroseis method is not a short pulse, but rather a "sweep" lasting some seven seconds or longer. The sweep is sent into the earth by a pad that is pressed against the ground surface and vibrated.

The sweep is transmitted through the earth and reflected as is any seismic signal, but on a vibroseis record, each reflection is as long as the input sweeps rather than being a pulse or wavelet as with an impulsive source. Each reflection is a near duplicate of the sweep itself and so the reflections in the vibroseis record overlap and are generally indistinguishable. To make the vibroseis record useable, we must "compress" the reflections into short wavelets. This is done by cross correlating the data with the original input sweep.

If we were to correlate a sweep with itself, the result would be a wavelet at time zero containing all of the frequencies in the sweep. This is an autocorrelogram of the sweep and is called a "Klauder wavelet". Similarly, when a vibroseis trace is correlated with a sweep, the result is that a wavelet is output every time a reflected sweep is encountered. On the correlated output record, each reflection has been "compressed" to a wavelet. Thus, correlated vibroseis data looks similar to impulsive source data and can be treated in much the same way.

V. DIVERSITY STACKING

Diversity stacking is an option that is available on most Demultiplex program. The diversity stack routine measures the amplitude of the trace, and then slides a window over the trace. The amplitude of each window is compared to the amplitude of the trace and each sample is automatically weighted according to the comparison before summing. If the window amplitude is higher than the trace amplitude, the window value is scaled down. Conversely, if the window amplitude is lower than the trace amplitude the window value is scaled up.

b). DISPLAY

The values obtained from the entire project depend largely on the way in which these displays are presented. They must accurately must accurately and concisely summaries the seismic information in a form which can be readily comprehended by the interpreter. Figure 2.4 illustrates in simplified from the way the data are displayed. The traces shown are produced by firing seven shots at successive locations along the line and recording each with a single geophone, right next to the shot. Although recording is not really done quite this way, the processing applied to the data makes it appear as if it were. The geophone generates an electrical signal in response to the vibration of the earth as the reflected energy returns to the surface. This signal is sent through a cable to the recording truck, where it is amplified and recorded on a tape which began moving at a fixed rate at the instant the shot was fired. This tape is then input to a series of digital processes and the resulting output tape is used to drive a plotter which moves a marker on a moving sheet of paper in response to the recorded signal. Timing annotation is put on the paper to tell us how much time elapsed between the shot and each reflection arrival.

At any point in the processing sequence the seismic analyst can display the data in wiggle trace or other modes.

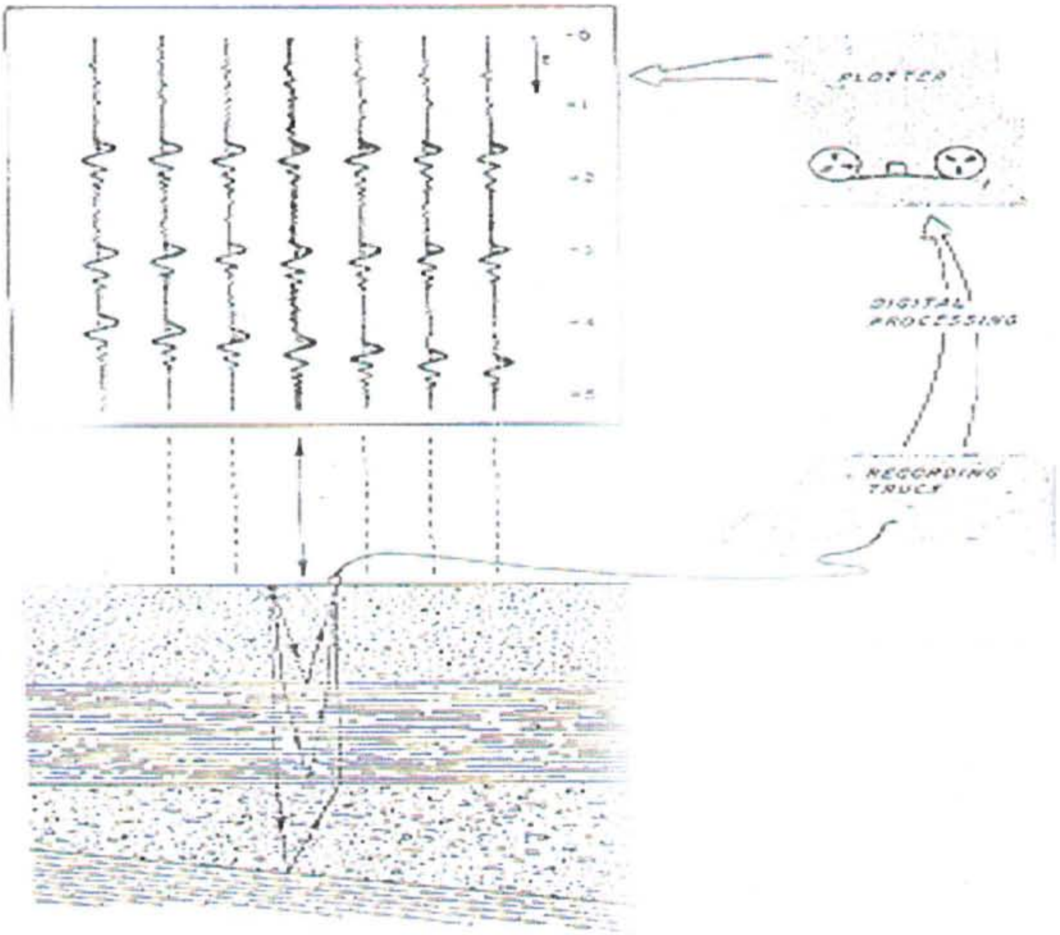


Figure 2.4

I. TRACE DISPLAY MODES

There are five conventional display modes that we can use

- Wiggles only
- Variable area wiggle

- Variable area only
- Variable density with wiggle
- Variable density without wiggle.

The first, wiggles only is the sort of display that comes out of the field camera monitor. Subtle features are hard to see, trace tangling can cause confusion, and zero crossings are not readily apparent. Generally a display of this sort is appropriate only for a preliminary comparative quality check of the data, of the sort of the field observer performs.

A sophistication of the wiggle only plots comes with blacking in one side, this is the variable area wiggles (v-a-w) plot .The full waveform is plotted, but this mode gives emphasis only to the blacked in peaks. Still the full waveform plotting is important when we undertake wavelet processing, and also when we want to pick and time an event properly.

A plot without wiggles but with variable area (v-a) only may be arranged to give equal emphasis to positive and negative excursions of the trace. Such a plot prints better than v-a-w plot. The problem with these modes is eyestrain, particularly for a person with astigmatism.

Variable density plots are used much less than the other modes for hard copy plotting. They are however, the norm for sections displayed on the cathode-ray tube of an interactive work station. Here, the degree of the shading is proportional to signal amplitude. Photographic processing must be done painstaking; however, as any change in amplitude affects the shading and perhaps, the appearance of event continuity.

A trace display mode which is achieving currency is the rectified variable area plot . Here, peaks (black) and troughs (gray) are plotted on the same side of the trace. For clarity of faults, these are very effective displays.

c). EDITING

Trace Editing is a process of removing or correcting any trace or record, which in their originally recorded form may cause deterioration to the stack. Several forms of editing are used throughout data processing which are categorized:

I. Editing during Demultiplex

II. Muting

I. EDITING DURING DEMULTIPLEX

A good rule to follow the editing process is; the sooner, the better therefore, demultiplex programs generally will contain edit features.

During the demultiplexing process there are two types of editing available:

a. Automatic editing

b. User selected editing

Automatic editing will reject traces which contain too many sync errors or rare too short. The programs usually allow the processor to specify the number of sync errors permitted or the number of samples required. The user selected editing is the one where the user must specify which traces or records will be omitted.

II. MUTING

If useless information is to be removed from the processing stream, it must first be identified and then blanked, which is called Muting. It is of two main types.

a. Initial Muting

b. Surgical Muting

d). AMPLITUDE ADJUSTMENTS

A gain recovery function is applied on the data to correct for the amplitude effects of wave front (spherical) divergence. This amounts to applying a geometric spreading function, which depends upon travel time, and an average primary velocity function, which is associated with primary reflections in a particular survey area. Gain is applied to seismic data for spherical spreading correction (Yilmaz, 1987).

Often AGC (automatic gain control) is applied to raise the level of the weak signals. AGC attempts to make amplitudes similar for all offsets, fro all times and for all mid points. A typical method of calculating the gain to be applied is to calculate the median or average amplitude within sliding windows down the trace, then to calculate the multiples needed to equalize the median values in all the windows.(Dobrin & Savit, 1988).

2.3 GEOMETRIC CORRECTIONS

A seismic trace on a field monitor shows reflected energy bursts from subsurface rock layer interfaces. We later measure the travel times from source down to reflect and back to geophones and use them together with average velocity information to compute depths to various reflectors. However, before we use these reflected energy bursts we must apply several corrections to compensate for geometric effects. These corrections include static corrections and NMO corrections applied on the trace gathered data.

a). TRACE GATHERING

According to the simple laws of reflection, the reflection point is located under the source-detector mid-point. By use of certain spread configuration, it is possible to record a number of shots such that the mid-point (i.e. the reflection point) is common to all of the applied spreads. Thus, with the suitable spread configuration (e.g. technique of expanding spread) we obtain a set of reflection signal which are all reflected from a common reflection (or depth) point. This is normally abbreviated as CRP or CDP set-up. Of course, the actual time travel varies from record to record since each is recorded with own offset. After correcting the observed travel times for all members of the set (normally referred to as the CDP gather) for the NMO and for the effects of near-surface irregularities, the CDP is summed together algebraically (i.e. stacked) to give an enhanced primary reflection.

Ever since the principle was outlined by Mayne (1962, 1967) it has been applied on a routine basis in the execution of seismic reflection surveys.

The improvement in the reflection signal brought about in the CDP stack depends largely on the accuracy of the NMO correction. The method is used effectively in the attenuation of multiples (Marr and Zagst, 1967). This is done on the basis that a multiple has a greater NMO than a primary reflection which occur at the same arrival-time. While other types of gathers are common source point gather, common receiver point gather and common offset gather

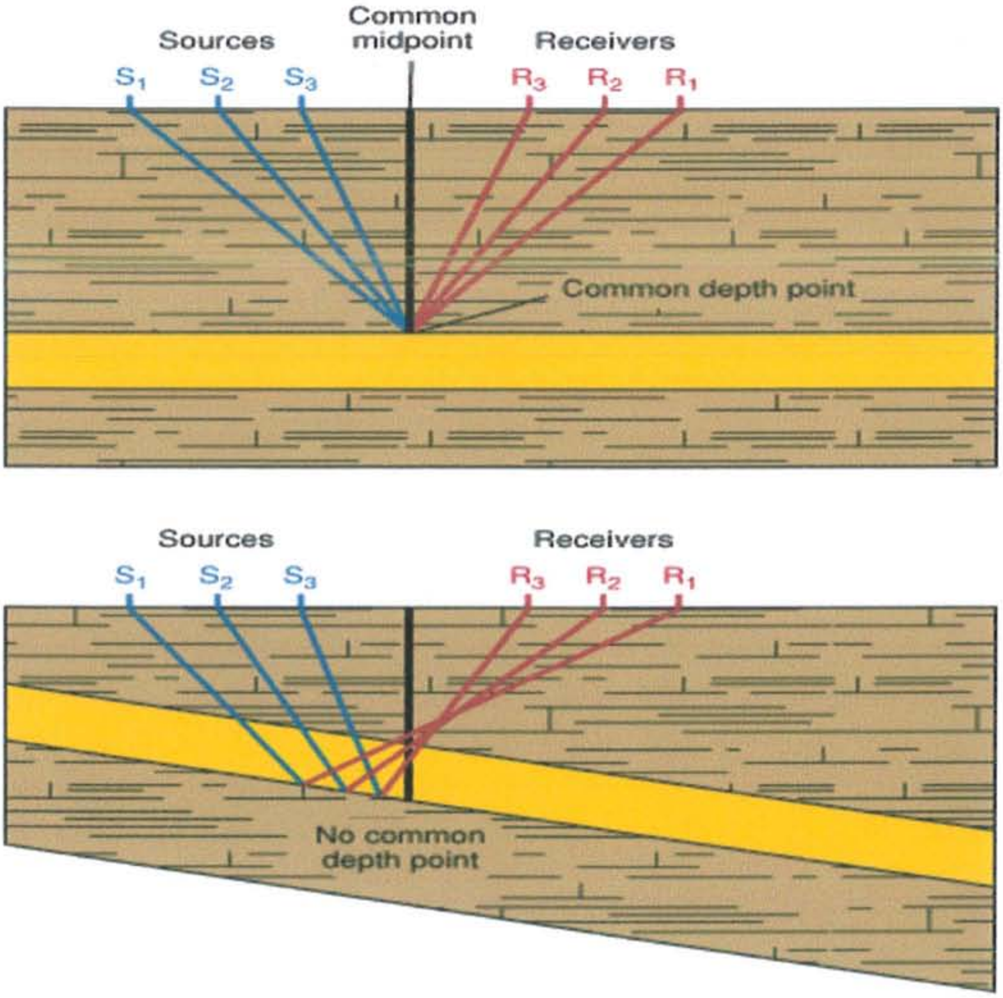


Figure 2.5

b). STATIC CORRECTIONS

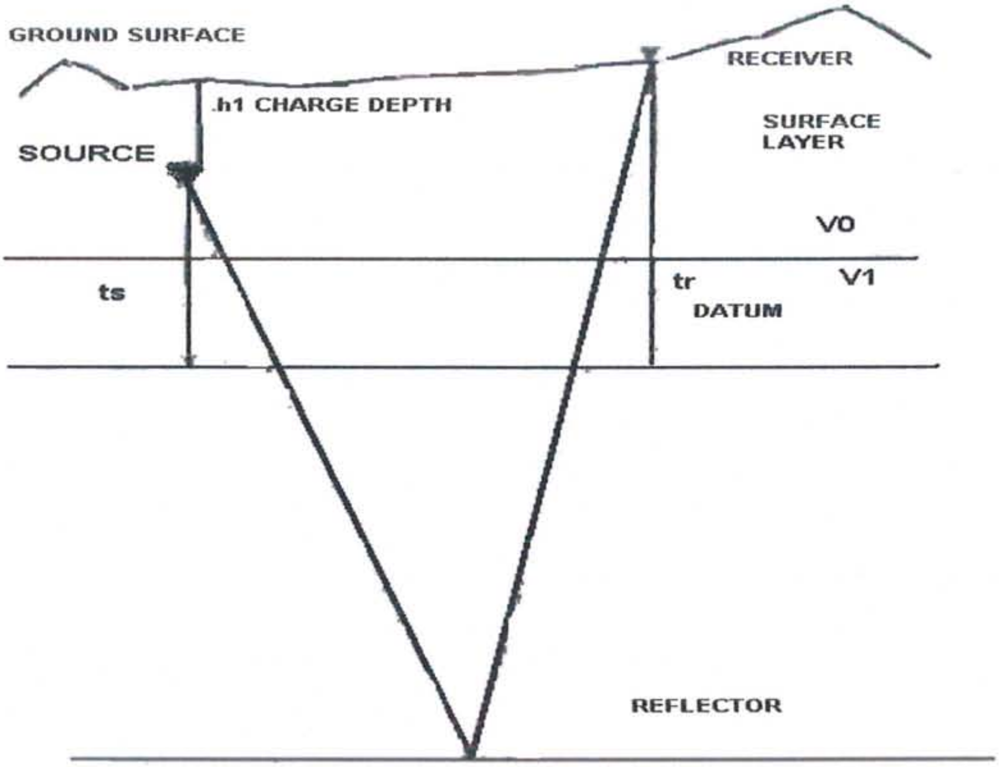
In order to obtain a seismic section which shows seismic reflectors representing the actual geological structure, the reflection times must be reduced to a defined reference time. This is normally taken to correspond to horizontal plane fixed at certain known elevation above mean sea level. Static correction is essential a time shift introduced to each trace reducing the observed reflection time to the chosen datum plane.

Although, it is customary to place the datum plane below the base of the variable-velocity surface layer, this is not always possible nor is it necessary. In any case, one can in principle place the datum at any reasonable position provided that the static correction is computed with sufficient accuracy.

The value of the total static correction (Δt) depends upon the following factors:

1) The perpendicular distance of the source from the datum plane.
2) The surface topography that is the perpendicular distance of the detector from the datum.
3) The velocity variation of the surface layer along the seismic line.
4) The thickness variation of the surface layer.

In computing Δt, it is usually assumed that the reflection raypath in the vicinity of the surface is vertical. Being close to the real situation, this assumption introduces only a negligible error. The total static correction Δt figure 2.6 is made up of two parts,



the static correction Δt_s and receiver correction Δt_r,

where :

$$\Delta t = \Delta t_s + \Delta t_r$$

The total-datum static correction (Δt) is normally computed in two stages. First, the source correction (Δt_s) is computed for all the sources points, then the receiver correction (Δt_r). The total correction is applied to each seismic trace by shifting the whole trace in time by an amount equal to the algebraic sum of the two corrections Δt_s and Δt_r.

In order to evaluate Δt_s , it is essential to know the surface elevation with respect to the adopted datum-plane, the source depth, and enough information about the surface layer. The surface elevation (E), shot-hole depth (h), thickness of the surface layer (s), and the velocities v_0 and v_1 are usually supplied by the field party. In application, Δt_s must be so calculated that both effects of topography and near-datum velocity abnormalities (weathered layer effects) are removed. Thus , when the datum plane is above the base of the surface layer or above the ground surface, Δt_s must compensate for the material which is lying immediately below the datum . This is achieved by mathematically replacing the anomalous v_0-layer by v_1-material .

The general formula for the Weathering Correction is :

WC = - dw (Vc –Vw) / (VwVc)

Where

dw = the depth of the weathered layer.

Vc = velocity in correct layer

Vc = velocity in weathered layer

Since **Vc > Vw** the weathering correction is always negative.

While for Elevation Correction the general formula is :

FOR SOURCE:

ECs = Ed – Es / Vc

FOR RECEIVER:

ECr = Ed – Er / Vc

Where:

ECs =elevation correction for source

ECr = elevation correction for receiver

Ed = datum plane elevation w.r.t surface

Es = elevation of the source w.r.t datum plane

Er = elevation of the receiver w.r.t datum plane

c). DYNAMIC CORRECTION

After the application of the static correction the reduced record time represent the two-way slant time where both the source and detector are effectively on the same datum. Now if we examine a reflection-event on all the contributes of one CDP, we find these events falling (in the ideal case) on a hyperbola. This is due to the dependence of travel time on the trace-offset, expressed by the NMO equation.

$$T_X^2 = T_o + X^2 / V^2$$

Where:

$T_o = 2h / V$ = zero offset travel time

When a proper dynamic correction (ΔT) is applied, the reflection travel time is reduced to two-way vertical time. Being for the same CDP, the reflection events will fall along a straight line producing a co-phasal alignment. Perfect alignment, i.e. zero residual ($\Delta^2T=0$) is achieved by proper choice of velocity which can be derived by special velocity analysis. Figure 2.6 shows the effects of static and dynamic corrections on the alignment of reflection arrivals observed in the traces which belong to one CDP.

The dynamic correction is carried out by computing ΔT per each trace sample. Thus a sample whose record time before the correction is T_X will be at time ($T_x - \Delta T$) after the correction. Since ΔT is a function of record time, T_x, different parts of the trace will be time-shifted differently, hence the correction is called dynamic (or time-dependent) to distinguish it from the time-independent or static correction.

To obtain the perfect dynamic correction, computation must be based on exact (i.e. minimum travel time) raypath. This means that the raypath must be determined by applying Snell's law and using the actual velocity distribution in the traversed medium. The apparent velocity which gives a proper (zero residual) correction is called stack velocity

2.4. DATA ANALYSIS AND PARAMETER OPTIMISATION

a). FILTERING:

Filtering is a process by which the frequency spectrum of a signal can be modified, reshaped, or manipulated according to some desired specification. It may entail amplifying or attenuating a range of frequency components, rejecting or isolating one specific frequency component, etc. the uses of filtering are manifold, e.g., to eliminate signal contamination such as noise, to remove signal distortion brought about by an imperfect transmission channel or by inaccuracies in measurement, to separate two or more distinct signals which were purposely mixed in order to maximize channel utilization, to resolve signals into their frequency components, to demodulate signals, to convert discrete-time signals into continuous-time signals and to band limit signals.

In Simple words filtering is an operation by which the amplitude and / or phase spectrum of a time signal are altered. Purpose of a filter is to enhance primary reflections by attenuating ambient and source-generated noise whose frequency spectra are separated sufficiently from that of coincident primary reflections.

Note: More detail of filtering is in chapter # 3.

b). DECONVOLUTION

INTRODUCTION:

Careful stacking of seismic data can remove most of the noises from the recorded earth signals. The signal may have been processed without filtering of the signal frequencies; nevertheless, the signal will have been modified by the natural filtering from transmission through the earth. Therefore the spectrum of the recorded signal will most likely suffer considerable attenuation of high, and some low, frequencies. Amplitude of the frequency components is important. A trace may have a very broad frequency band of reliable signal elements but those at low amplitudes can carry very little significant information. The frequency component must be restored to their original level if they are to provide the proper contribution of information to the seismic trace.

The process by which the attenuated elements are restored is another filter operation termed Deconvolution.

Deconvolution may also be defined as a certain process which counteracts a previous convolution action.

Although convolution follows the same commutative rule as multiplication, convolution is not multiplication. Neither is deconvolution division. A form of inverse convolution can be used to obtain the input function when the system transfer functions known, but this operation frequently fails in the presence of additive noise. A more practical process is to convolve the output signal (the seismic trace with the inverse of the transfer function to determine the input signal)

Output * 1.0 / Transfer Function = Input

While the operation 1.0 / Transfer Function

is not defined, i.e. there is no step-by-step equivalent of division fro handling functions or ordered arrays of numbers, the desired result can be obtained by computing another function which, when convolved with the Transfer Function will yield 1.0, the delta function ,or as close an approximation to it as required. This new function, when convolved with the output will "de-convolve" it to yield the input.

I. DECONVOLTUTION MODEL

All Deconvolution procedures may be related to some model of the process which yields the recorded seismic trace. A synthetic seismogram is an example of the type of modeling process used.

Basic assumption used to develop the model and the success of the deconvolution process is dependent upon the validity of these assumptions under real conditions. For example, the reliability of a process may be degraded by the presence of noise if a noise-free model was the basis of the development. Noise may be added to the model to compensate for this problem, but then the quality of deconvolution will depend upon whether or not the noise recorded in the field is indeed the same type as that used in the model.

Theoretical models, upon which the majority of deconvolution theory depends, assume several, if not all, of the following conditions:

1) The input signal is known

2) The transfer function is known

3) The output is broad-band

4) The noise is random white noise

5) The spectrum of the earth signal(reflection coefficients) is white

6) The output is stationary, i.e. does not change with time.

In reality the following conditions probably exist:

1) The input signal is not defined
2) The transfer function is not known
3) The unit impulse response is mixed phase
4) The output is narrow band
5) The noise may be periodic
6) The spectrum of the earth signal is not white
7) The output is time variant

The fact that such a large number of discrepancies exist has not deterred the pioneers in the field of signal processing. Many of the conditions vary, and unless the interpreter can recognize the situation existing for a particular area, and knows what processes are available, what assumptions were used in their formulation, and their effects on the data, he may not achieve the desired output. The difference between desired and actual output usually represents the effects of errors in the basic assumptions. Differences between theory and practice will nearly always be present to some degree because, as a rule, the basic assumptions are never quite correct.

c). VELOCITY ANALYSIS

The objective of the seismic survey method is a reliable measurement of the travel time between a signal source and a receiver. The results of a reflection survey are displayed on a time section and geophysicists map structure in units of reflection travel time. A geologist, on the other hand, is more interested in the depth of the horizons, and the thickness of the section components measured in units of meters or feet.

The key which links the time domain to the depth domain is velocity, which in itself can be a significant diagnostic tool for the determination of lithology.

The velocity determination and its relation to seismic data is in itself of sufficient interest, importance and complexity to fill an entire volume. All seismic field records contain the means for determining velocity information to some degree of reliability, but prior to the advent of digital processing an entire seismic survey could be completed without a velocity determination. Interpretation was done directly on original paper field records, continuity of horizon was carried from record to record by mapping near-trace times where dynamic effects are not significant, and the remaining traces were merely use to correlate continuity across the space between. Velocity determination was used primarily as an aid to depth estimations but often time structure maps were considered adequate.

Velocities can be obtained by direct bore hole measurements (velocity survey) or derived from seismic field records. The normal Moveout approximately follows the expansion of geometry if a right triangle so the familiar X^2, T^2 method of plotting trace-time squared against trace-distance squared was a fairly common early method. Occasionally, a rather time consuming expanded field profile was shot for more precise measurement of the velocity function, or direct measurement of velocity was accomplished by shooting a down hole well survey. A geophone lowered in the borehole measures the direct travel time from the surface to several depths

And the general formula used in determination of the velocity is:

$$X^2 = V^2 T_x^2 - V^2 T_\circ$$

$$V^2 = X^2 / (T_x^2 - T_\circ^2)$$

I. Type of velocities

Various types of seismic velocities are reported on seismic profile legends and in reports of seismic processing and interpretation.

Average velocity: The distance to an interface, divided by the one-way travel time to that interface, is the average velocity for the material above the interface.

$$V_{av} = z / t$$

Where :

V_{av} = average velocity above the interface

Z = depth from the surface to the interface

T = one-way travel time from surface to interface.

For a layered sequence above a reflector, the average velocity can be envisioned according to "V-shaped" rays that do not bend across interfaces; the average is time-weighted according to:

$$V_{av} = \sum V_i t_i \ / \sum t_i$$

Where:

V_i = seismic velocity of the i th layer

t_i= vertical, two-way travel time within the i th layer.

- **Root Mean Square (RMS) velocity (V_{rms})**

Snell's law describes bending as rays refract across an interface separating different velocities. Rays are refracted so that they travel more horizontally along high-velocity layers, compared to more vertical travel through low-velocity layers. Consequently, rays spend proportionally more travel time in higher-velocity layers. The root mean square (RMS) velocity is a weighted average; it accounts for the disproportionate travel time in high-velocity layers by squaring the velocities in the $V_i t_i$ term, then taking the square root of the averaged sum:

$$(V_{rms}) \ = \surd \sum V_i t_i \ / \sum t_i$$

where:

V_{rms} = root mean square velocity

V_i = seismic velocity of the i th layer

t_i = vertical, two-way travel time within the i th layer.

- **Stacking Velocity (V_{int})**

The interval velocity is the average velocity of the material between two interfaces

$$V_{Int} = \Delta z \ / \ \Delta t$$

Where:

V_{int} = average velocity between two interfaces

Δz = thickness of layer between the two interfaces

Δt = one-way travel time between the two interfaces.

Velocity information from seismic reflection surveys is in the from of stacking velocities determined during processing. The Dix Equation computes the interval velocity between reflecting interfaces by assuming that the stacking velocities are RMS velocities

$$V_{int} = \sqrt{}V^2{}_2\, t_2 - V^2{}_1\, t_1 \,/\, t_2 - t_1$$

Where :

V_{int} = interval velocity between upper and lower interface.

V_1 = stacking velocity for reflection from upper interface.

V_2 = stacking velocity fro reflection from lower interface.

t_1 = vertical two-way travel time to upper interface.

t_2 = vertical two-way travel time to lower interface.

2.5 DATA REFINEMENT

a). MUTING

If useless information is to be removed from the processing stream, it must first be identified and then blanked, which is called Muting. It is of two main types.

I. Initial Muting

II. Surgical Muting

I. INITIAL MUTING

It removes the large energy bursts associated with first arrivals. This type of muting is usually done later in the processing sequence. The initial muting function used is often a straight line which is chosen after examination of monitor records or trace gathers.

II. SURGICAL MUTING

Another set of interference encountered in seismic records is labeled "ground roll". These are the Rayleigh waves which propagated along the earth surface. There velocities are half to the velocities of the "p-waves". Some data may also contain "Air-waves". This is the energy that has traveled from the source to the receivers through the velocity of sound in the air (about 1100 ft/sec). The ground roll should be attenuated in the field by using proper recording filters and source detector arrays. However, if they are present in the recorded data necessary to apply a surgical Mute to zero out that portion of each trace containing the noise train.

b). STACKING:

Of all the advances in the reflection seismic technique, surely none is more important nor has greater impact than the invention of common-depth-point coverage, or simply, stacking.

In concept, the procedure is very simple. The distance between seismic source and receiver is varied symmetrically to survey the same subsurface point several times. The observations, corrected for geometrical distortion and adjusted to datum, are then summed to obtain the average.

The reason for its outstanding success is that it provides the most powerful method for separating noise from signals of the same frequency. With one exception, all other seismic processes either eliminate both signal and the noise, or merely modify the amplitude without changing the proportion. The exception is multichannel filtering, which is also able to separate signal from noise of the same frequency, but it is effective only in areas of relatively homogeneous dip and is of little value in areas of complex structure. Under proper conditions, stacking will remove several types of noise, but it is most effective against random noise.

Any signal, be it useful information or noise, can be analyzed by the Fourier transform into its component frequency element. The contribution of each frequency can be expressed in terms of a vector having a length equal to the amplitude, directed at the angle of phase. Random white noise has uniform amplitude for all frequencies, but the phase is totally random. Seismic traces represent the sum of signal and noise. Each frequency component is the vector sum of a random noise component and a signal component. For a twelve-fold stack, if all corrections have been properly applied, all twelve signal components of each frequency can be considered vectors having the same amplitude and phase. The twelve noise vectors will also all have the same amplitude, perhaps greater than the signal vectors, but the phase distribution will be random (figure 2.7).

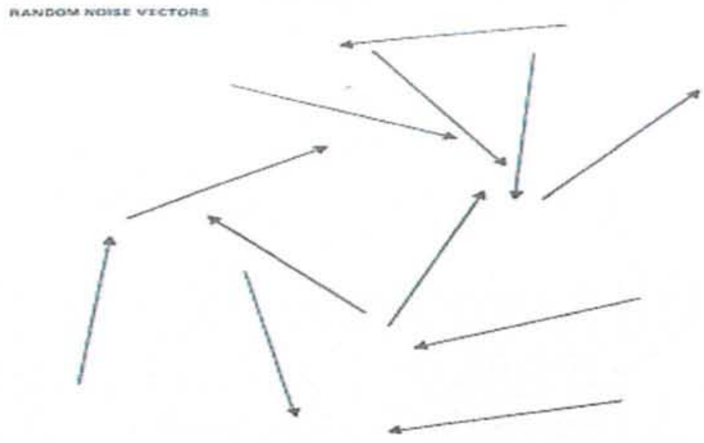


Figure 2.7

If the ratio of signal-to-noise is 1:4, the signal vector would sum with the noise, as in figure 2.8 to yield the results indicated by the heavy lines.

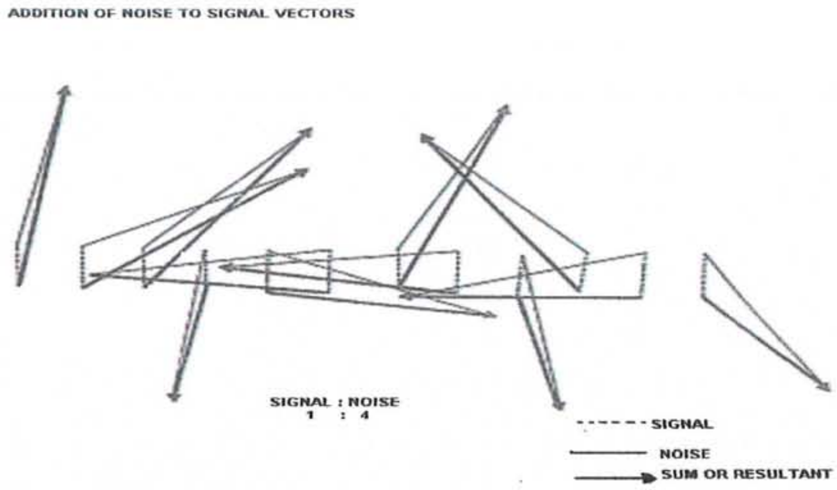


Figure 2.8

In the stacking process, the resultant vector of each trace is summed in turn by vector addition to produce the total resultant. In the addition, random noise components tend to sum to zero, while the signal components sum in phase to produce a resultant nearly twelve times the individual signal amplitude component, directed approximately at the signal phase angle (figure 2.9)

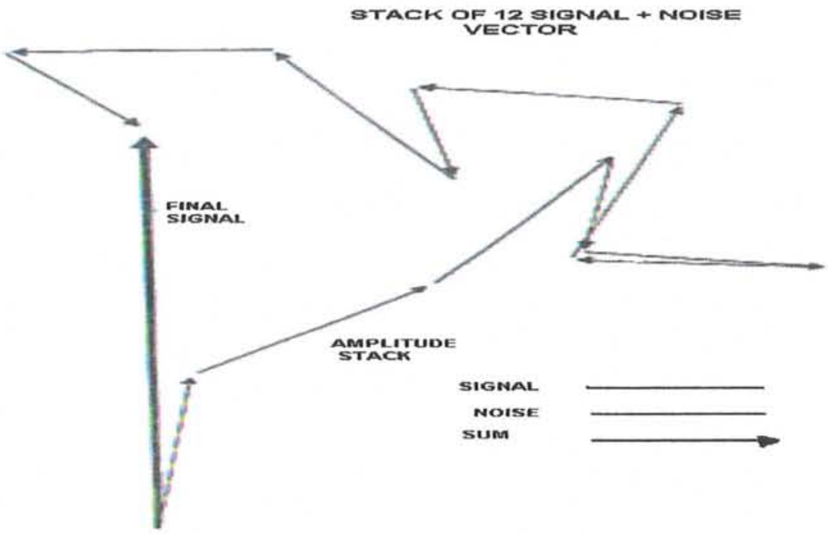


Figure 2.9

the method uses the power of statistics to produce relatively noise-free output. Naturally, if the noise has coherent components these too will sum in phase and be present in the output.

The assurance that the normal Moveout corrections and static corrections have been applied in an optimum manner is essential, otherwise the signal components will sum out of phase.

Over the years stacking has increased from 3-fold to the current 24-and 48-fold. Probably still greater coverage would be used if it were possible to increase readily the number of recording channels. But this is limited by the data rate to be recorded.

c). MIGRATION

INTRODUCTION

It is a well known, but often ignored, fact that the events which appear on a seismic section rarely correspond to their true position in space. Correspondence is possible only if the reflecting horizon is truly a horizontal plane surface. Under all other conditions the reflection must be migrated in order to position it properly in space. Reflected energy obeys physical laws and in the simplest dip case, the travel path back to the origin will be perpendicular to the dip of a reflecting bed, rather than vertically as it is plotted on the seismic cross section.

In the area of moderate dip, say up to 10 degree, the seismic section approximates the structure will sufficient accuracy to be acceptable. In areas of steep dip a reflection point plotted below the shot point may actually be removed laterally several hundreds of feet from the vertical line below the shot point. Even this problem may not be too serious in areas of fairly simple structure, because the error diminishes with decreasing dip and the geologically important crest of an anticline will be correctly located.

In more complex areas, where faulting, severe non-symmetrical folding, and sharp synclines are present, diffractions and double image reflections may be distorted to the point where the resulting seismic section bears little or no resemblance to the actual structure. This situation is not uncommon and it can be very difficult to interpret true structure completely from a section that has not been migrated properly.

Complicating the picture is the fact that diffraction patterns are recorded with the reflections. This phenomenon is most prevalent in faulted areas but is common to a certain extent to all. The termination of a reflection can grade smoothly from a reflection into diffraction. When this occurs all or part of the diffraction might be interpreted as dipping reflected energy. This is erroneous and may result in failure to recognize faulting, or in improper location of the fault plane, with the result that a well may be drilled on the wrong side of the fault. Synclinal folds with a radius smaller than the distance of the reflector to the surface produce an apparent structural inversion which can be mistaken for anticline or fault diffraction.

Prior to automatic migration by digital computer, seismic section dips could be transformed to an approximation of the true section only by manual migration. Two reflecting points on tan apparent dip segment can be timed, and if the velocity function is reasonably well defined, a series of rather lengthy calculation will yield the co-ordinates in space along the line of section of the points from which the reflection originated.

One common practice was to draw up a raypath chart using some relatively simple velocity function for the area. Reflections could be migrated by locating the appropriate reflection time and the amount of dip, expressed in two-way time, on the chart, and drawing in a dip segment tangent to the computed wave front. Some of these steps were automated when digital computers first came available, and routines to compute and draw migrated dip section (as illustrated in figure 2.10 were a welcome relief to interpreters.

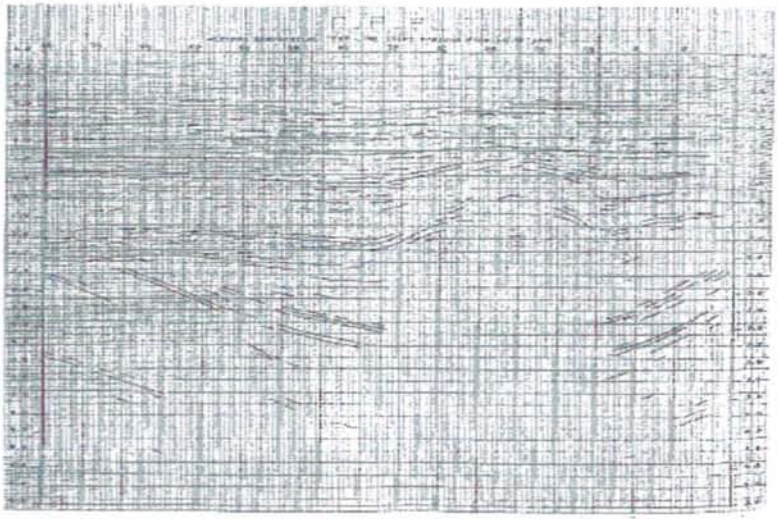


Figure 2.10

d). GENERAL PRINCIPLES

Assume a geological section homogeneous above the first reflector and a velocity function such that, for the purpose at hand, time and distance may be interchanged.

If the source and the receiver are placed at the same location A on the surface of the section, (figure 2.11)

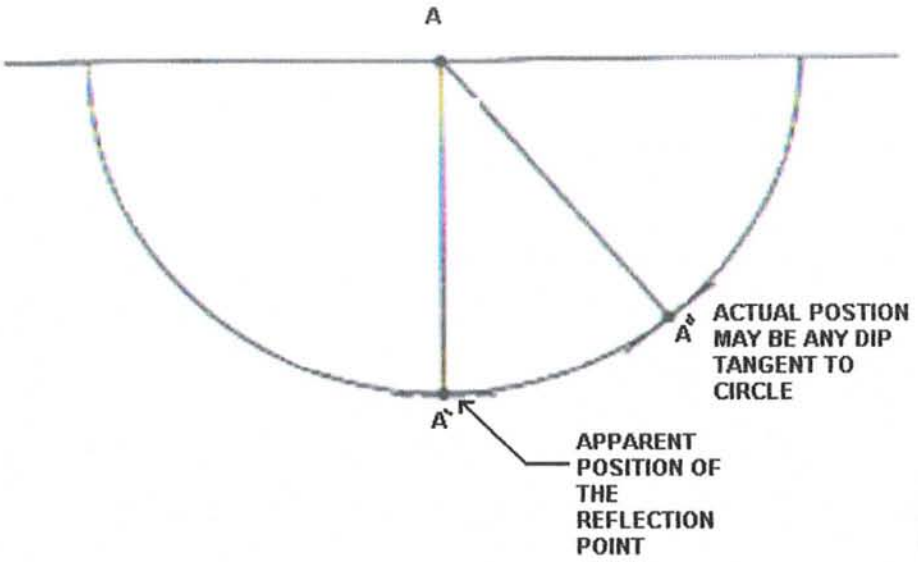

A
A''
ACTUAL POSTION MAY BE ANY DIP TANGENT TO CIRCLE
A'
APPARENT POSITION OF THE REFLECTION POINT

Figure 2.11

A seismic reflection received from the first reflection boundary would be plotted vertically below the source at point A´. In addition to a point directly below the receiver, the reflection could originate from any other point on the half circle having a radius AA´. Any reflector tangent to the circle could reflect energy back to the origin.

In fact, the radius of reflection not only describes a half circle but actually sweeps out a spheroid perpendicular to the line direction. Unless additional information is recorded, e.g. a cross spread perpendicular to the surface line, the lateral component will be unknown so only the component along the line of section can be considered.

A similar recorded from a second point B´ (figure 2.12)

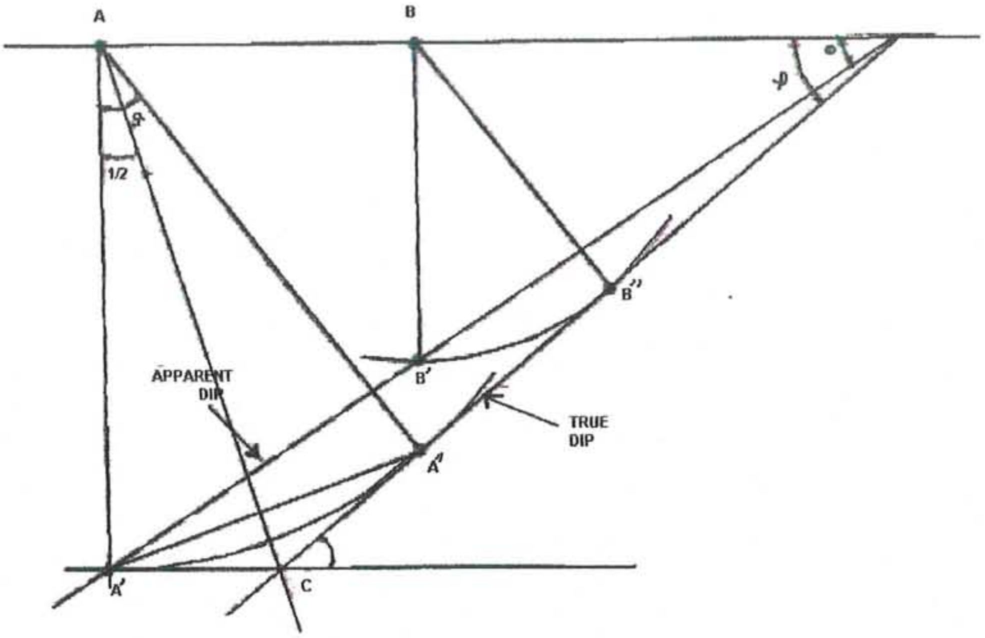


Figure 2.12

will produce the same results but, if the bed is dipping, travel time will be shorter and will appear at B. The horizon will then appear to have dip A´B´.

Under the special condition defined for our model, half-circles which represent the point from which A´ and B´ may originate can be struck off. A line drawn tangent to both will provide the plane A´´B´´ representing the true picture of the reflector in the line of the section. Note that the projection of the apparent dip and true dip intersect at the surface datum. If the apparent angle dip is ɵ and the true dip angle Φ then:

tan ɵ = sin Φ.

Under the condition outlined, it is not possible for any unmigrated seismic section to have a dip greater then 45 degree. A vertical reflector at B would have the reflection time equal to AB plotted vertically below A at A´ (figure 2.13)

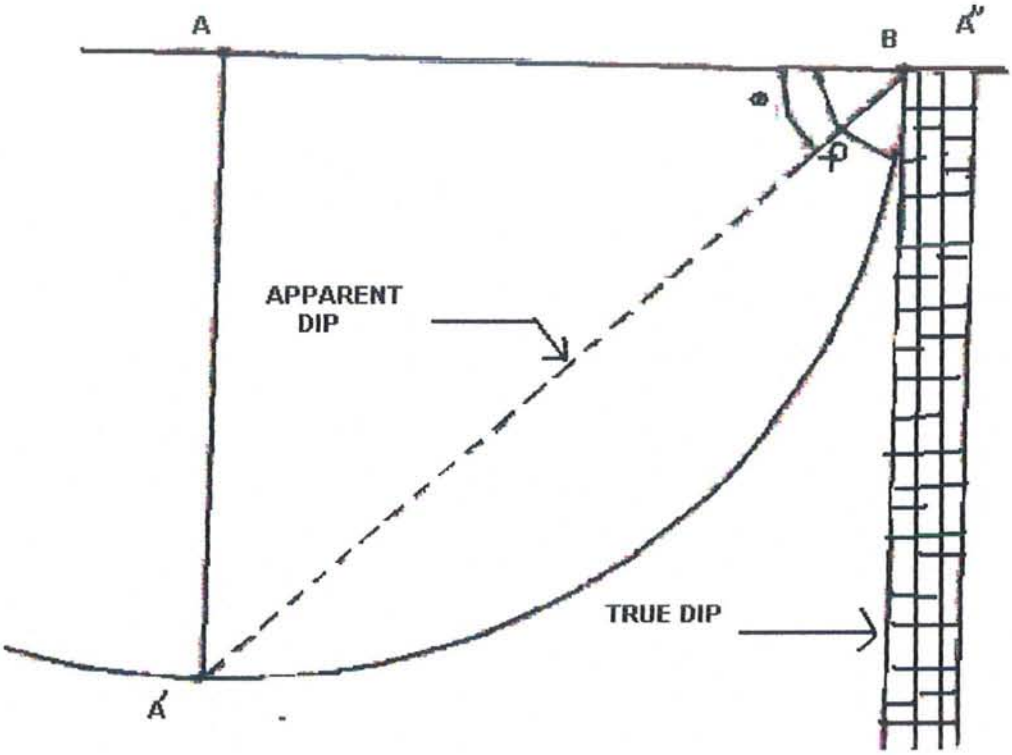


Figure 2.13

Thus AB and AA´ would be two equal sides of a right triangle, making the apparent dip 45 degree for a true dip of 90 degree. The angular relationship between apparent dip and true dip, tan θ = sin Φ confirm this, and show the angular difference between apparent and true dip to increase with increasing dip.

The continuous reflector will have a vertical trace plotted for each geophone position. To locate it properly in space, the reflection times recorded for all intermediate location could be swung on arcs to develop the tangents marking the position of the origin. If the intermediate locations are sufficiently close, the result of swinging the several arcs will be a continuous wave front. Figure(2.14)

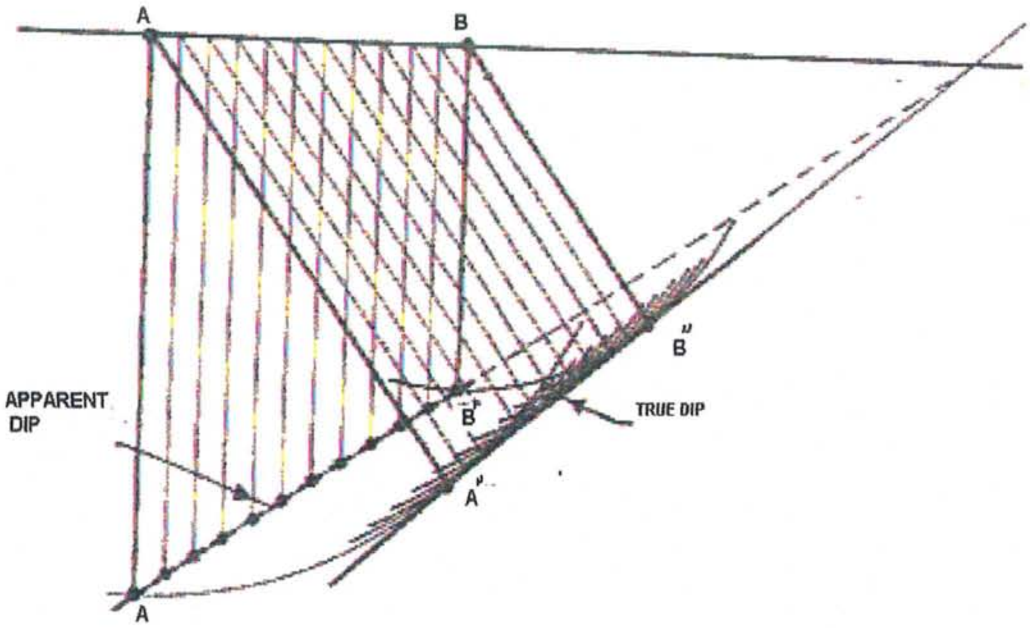


Figure2.14

This is exactly the principle expounded by Huygens. It is the basis of the original automated migration technique. The simple basic process of moving each reflected point to every other possible reflecting point, modified to consider actual physical condition, will provide an effective migration of the data, if the velocity has been adequately determined.

This procedure described assumes a source and receiver at the same position. In practice most receivers are located some distance from the source.

CHAPTER 3

FILTERING

FILTERING

3.1 Definition

Filtering is an operation by which the amplitude and / or phase spectrum of the time signal are altered or the purpose of a filter is to extract certain desired frequency component and attenuate all others. A filter can be designed and applied either in the time domain or the frequency domain. The results are same.

3.2 What is Digital Filter?

To extract a signal frequency from a trace in the time domain we convolve the trace with a cosine wave having the desire frequency. The cosine wave is the "filter operator".

The output of this filtering process will be a single frequency (figure 3.1) the amplitude and phase of eh output is equal to that of the same frequency component in the original trace. (figure 3.2) shows the same function in the frequency domain. The amplitude spectrum of the operator is very simple; a spike at the desire frequency. The phase spectrum is simpler yet; zero at all frequencies. This is the zero phase filters. To filter the data in the time domain we convolved the trace with the filter operator. To perform the same operation in the frequency domain we multiply the amplitude spectra, frequency for frequency and add the phase spectra. In the output trace spectra we see that the amplitude spectrum contains only the frequency we select and the phase spectrum is unchanged.(only the phase of the selected frequency (f) is significant since the amplitude of all other frequencies are zero). If we now convert the output trace spectra back to the time domain we will get a trace that is identical to the output trace.

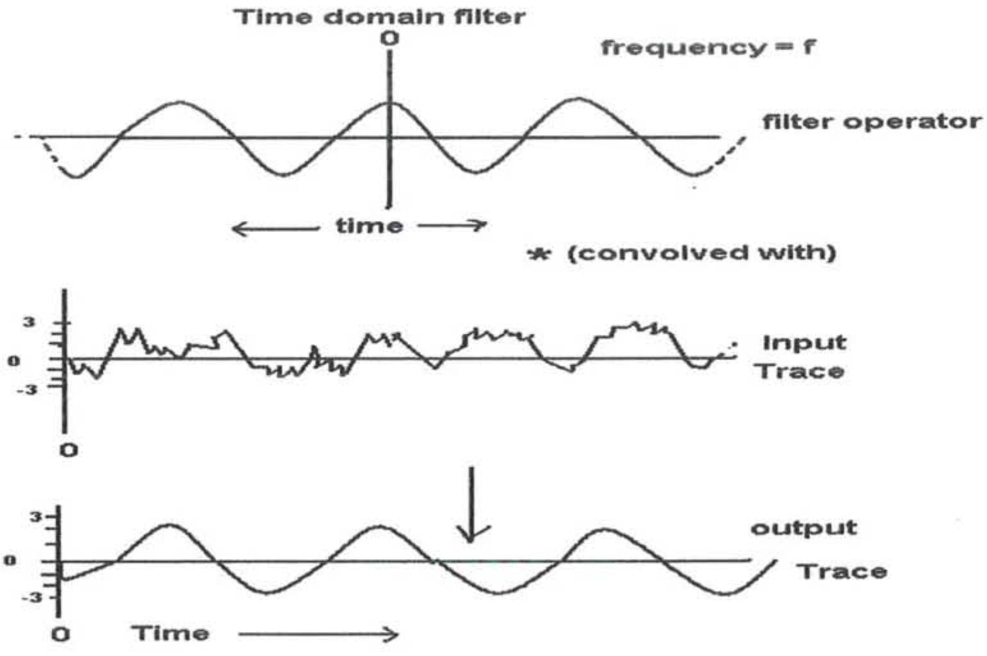


Figure 3.1

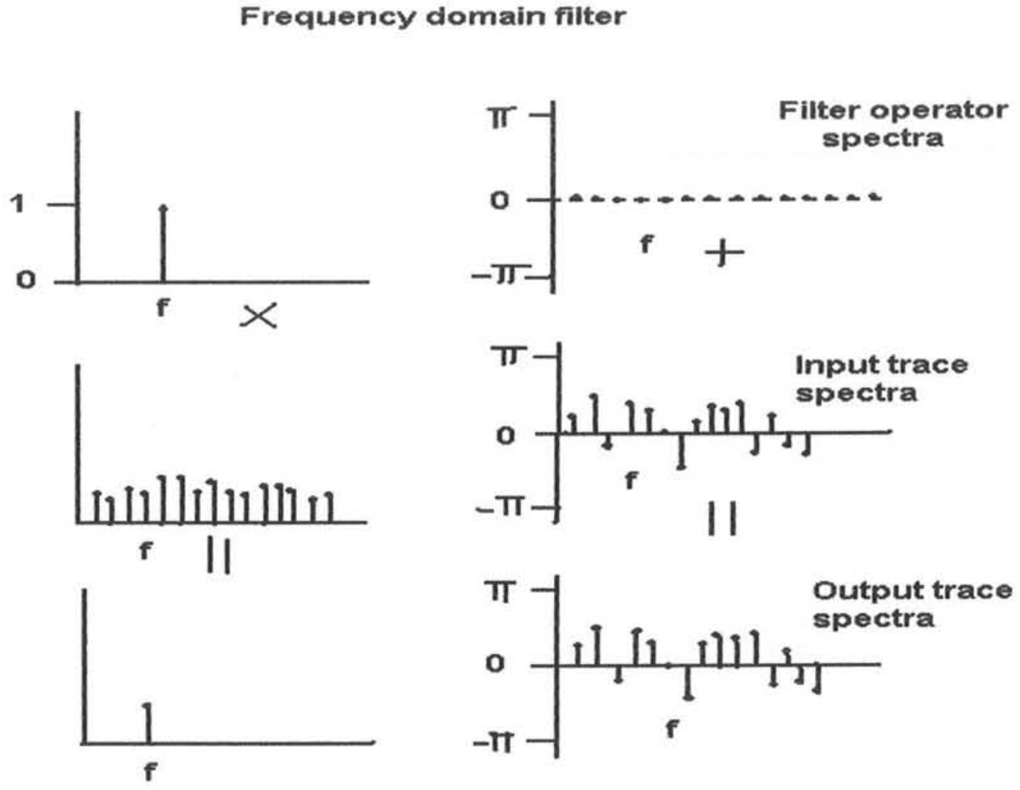


Figure 3.2

3.3 Reason for the Digital filtering.

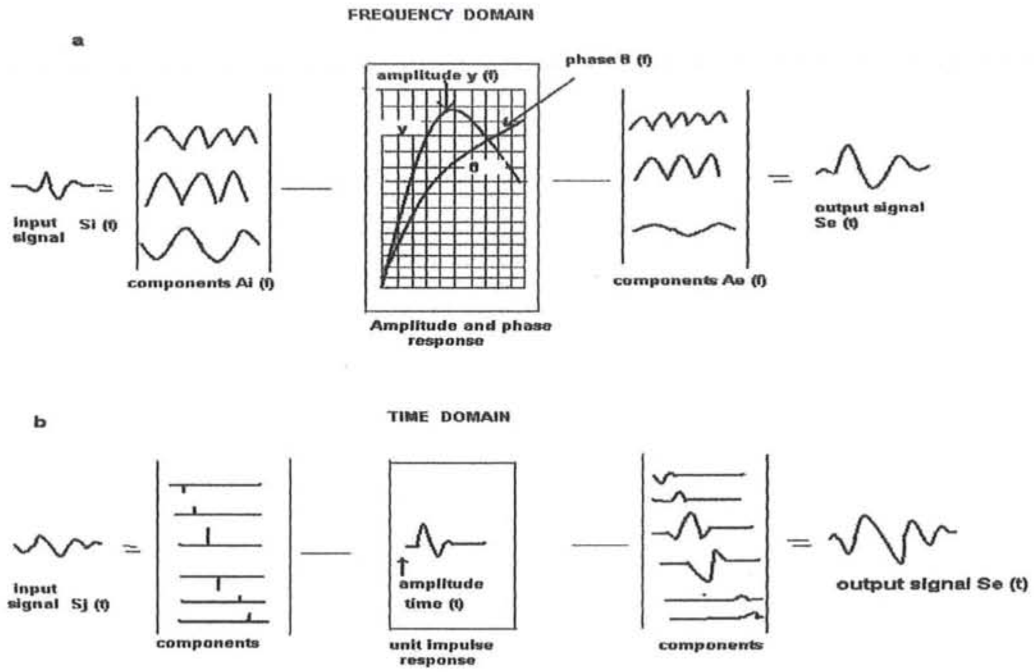
The basic reasons that motivate all filtering operations performed on the geophysical data are the desire to reduce unwanted noise and to enhance the quality of the signal. Since the variety of the signals is quite large and the type of the noise one must contend with the both numerous and diverse, it is easy to see why the filtering operations used on geophysical data are themselves an electric multitude.
Filters may be characterized in two ways which are identical mathematically.

3.4 Frequency domain

3.41 Time domain

In frequency domain the filter is characterized by its amplitude and phase spectra whereas in time domain it is characterized by a unit impulse response signal. The latter is simply the Fourier transform of the filer amplitude and phase spectra.

The frequency domain description of filtering is illustrated in figure



A filter input signal S_i (t) may be replaced by a sequence of co sinusoidal signals (spectral components);

$$S_i\ (t) = \sum f\ A_i\ (f)\cos\ [\ 2\pi ft + \Phi_i\ (f)]$$

The filter then modifies the amplitude spectrum A_i (f) and phase spectrum Φ_i (f) of input signal, resulting in an out put signal S_e (t).

$$S_e\ (t)\ =\ \sum f\ A_i\ (f)\ Y(f)\ \cos\ [\ 2\pi ft + \Phi_i\ (f) + \Phi\ (f)]$$

Where Y (f) and Φ (f) are the amplitude and phase spectrum or response, respectively, of the filter. Thus, the amplitude spectrum of the output signal A_e (f) is simply the product of the input signal amplitude spectrum and filter amplitude response.

$$A_e\ (f) =\ A_i\ \ Y(f)$$

And the phase spectrum of the output signal is simply the sum of the input signal phase spectrum and filter phase spectrum response.

$$\Phi_e\ (f) =\ \Phi_i\ (f) + \Phi\ (f).$$

In figure above only three of a large number of input signal spectral components are shown. The amplitude and the phase of each are modified by the filter, resulting in the spectral components of the output signal. The output signal then is given by the sum of these components.

$$S_e\ (t) = \sum_f A_e\ (f)\cos\ [\ 2\pi ft + \Phi_e\ \ (f)]$$

The time domain description of filtering is illustrated in figure above (b). The filter input signal is considered composed of a sequence of impulses ("spikes"); that is, simply the amplitudes of the digitized signal. The filter replaces each impulse with its impulse with its unit impulse response signal Y(t),

$$Y(t) = \sum_f Y(f) \cos [\, 2\pi ft + \Phi\ (f)],$$

With the same polarity and a peak amplitude proportional to that of the impulse. The output signal then is the sum of these impulse response signals,

$$S_e(t) = \sum_t Si\ (t - T)\ Y(t)$$

The operation is called "Convolution", which has many applications in seismic data analysis and interpretation; notably, in the construction of synthetic seismic traces and sections. An abbreviated from of equation above is;

$$S_e(t) = Si\ (t) * Y(t)$$

Where the asterisk indicates convolution.

NOTE: The filter may be either phase distortion or phase distortion less.

3.42 TYPES OF FILTER

The capacity of any signal to carry information can be measured in a manner analogous to the volume of a container. Just as volume is the product of height times width times length, information capacity of seismic signal is related to a product of amplitude, frequency bandwidth, and length of the signal. A dynamite energy source pulse produces an input signal having considerable amplitude,(height), and bandwidth, but very short length. A Vibroseis energy source compensates for its limited amplitude by extending the input sweep signal over a relatively long length.

One of the most obvious and most common filtering operations is the acceptance of a certain band of frequencies in the data, with frequencies outsides this band being rejected. Such a filter (known as a ***Band-pass filter***) might be used if the frequency of the desired signal are known to be confined to a certain range(i.e. the signal is band-limited) where as outside this range the data contains mainly noise. An example of a band-limited signal is in

Vibroseis data for which the frequency range of the input energy is known to lie for all practice purpose, within a certain band (e.g. 10-40 Hz)

Digital bandpass filters are fairly easy to design and apply. A digital band-pass operator is, in essence, nothing more than the zero phase sum of a collection of frequencies extending over the bandwidth which is to be retained. Cross-correlation of any signal frequency with a signal will extract the frequency and that alone from the signal, since the output from the cross-correlation operation can have no frequencies which are not common to both the signal and the operator. Cross-correlation (or convolution) of several summed frequencies with the signal, will extract those frequencies and reject the rest

Reducing the bandwidth of a signal, as may be done with a band-pass filter, also reduces its capacity to carry information. A reduction in bandwidth not only limits amount of information but also to some degree, determines the type of the information that can be carried. Yet, in the years prior to the advent of stacking and digital processing, application of a band-pass filter was practically the only procedure available to eliminate noise from seismic data.

For the opposite situation, where one knows the frequency range of the noise, one may wish to use a band-reject filter, if the rejected band does not contain a significant amount of the desired signal. A special case of this kind of filter is provided by the 60 Hz noise components found in some seismic data whose origin is the local power lines in the vicinity of the data-recording area. Normally this 60 Hz component is removed in the field by a very narrow band-reject filter (a so called ***notch filter)***.

When using potential field data such as magnetic surveys to define large anomalies in the subsurface, it is usually desirable to remove high frequency noise. In this case one applies smoothing filters which pass only long wavelengths. These are the analog of time filters which pass only low frequencies ***(low pass filters)*** since it is the inverse of the wavelength (k= 1/L, where L = wavelength and k = wave number) which is the spatial variable corresponds to the frequency.

A familiar problem associated with seismic data is the presence of so called " multiples", unwanted signals whose origin is the reverberation of the seismic signal within a water layer or between several layers of the earth. The strength, character and period of these multiples vary considerably from one area to another and even within a particular line. Hence, a filtering process whose purpose is to attenuate such multiples must be a relatively sophisticated operation as it must decided upon the characteristics of the filter to apply on the basis of the particular data on which it is working. Thus, these filters (called ***deconvolution filters***) are of some more complicated nature than most of the other seismic filters.

3.43 PRE – PROCESSING FILTERS

For seismic data acquired by standard field techniques two types of preprocessing filtering are important : time filters which effect the bandwidth and spectrum , and spatial filters , which determines the apparent wavelength of the signals which are recorded. Here we discuss only the time filters

a). TIME FILTERS

The analog signal which is digitized will have important limitations on its frequency spectrum which are determined by the electronics as well as by the response function of the phone.

Generally, the high frequencies are removed from analog signal in order to prevent aliasing upon digitization, while the very low frequencies which might be present in the original seismic signal are not recorded due to the insensitivity of geophone (hydrophone) at low frequencies.

b). ALIAS FILTER

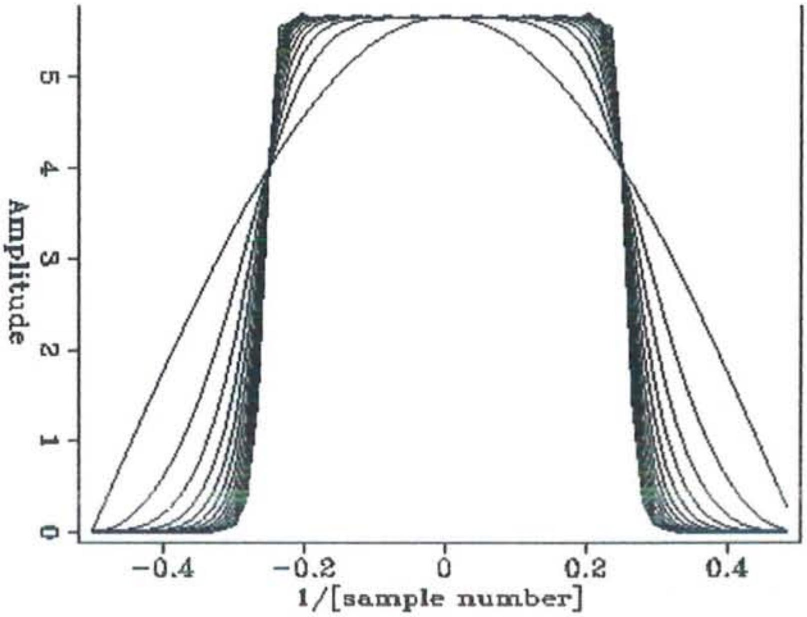
The frequency above the Nyquist frequency (fN=1/2 dT where dT = sample rate) cannot be measured. If there are such frequencies present in the data, they will be recorded as if they had a lower frequency; we say the higher frequencies appear aliasing in the digital signal. In order to prevent this effect from marring the quality of the seismic data, the high frequencies are electronically removed from the signal before it is digitized. Since fro most seismic work dT = 4 milliseconds, fN = 125 Hz and one must be sure to cut out in the field all frequencies above 125 Hz.

c). LOW CUT FILTERS

In addition to the high-cut filters used to prevent aliasing, low –cut filters are placed on the analog signals for a number of reasons. One consideration is the basic response of the geophones and hydrophones used to record seismic signals. These should ideally have a flat response as a function of frequency , i.e. they should record all frequencies without distorting the input frequency spectrum. In general the geophone and hydrophones approximate a flat frequency response curve for frequencies above a certain lower limit. This lower limit varies with the particular instruments, but it is generally on the order of 8 Hz.

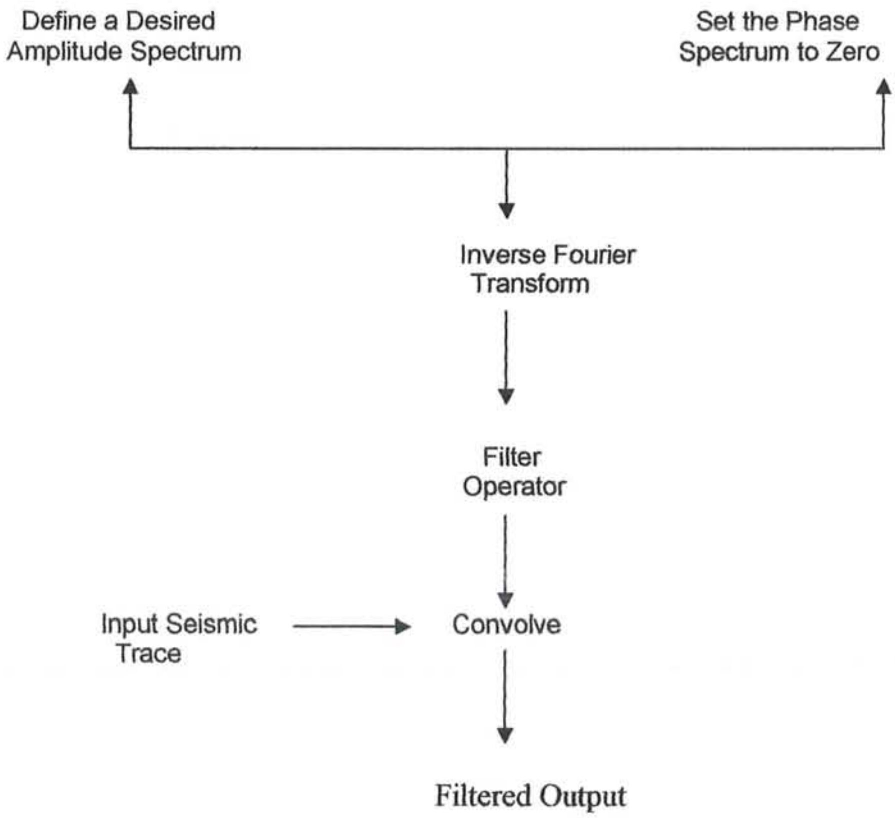
A less obvious but important consideration in deciding which low-cut filter to use in an analysis of the frequency spectrum of the signal one wishes to record versus that of the noise prevalent in the recording area.

AMPLITUDE RESPONSE WITH THE CHANGE IN ORDER OF THE FILTER

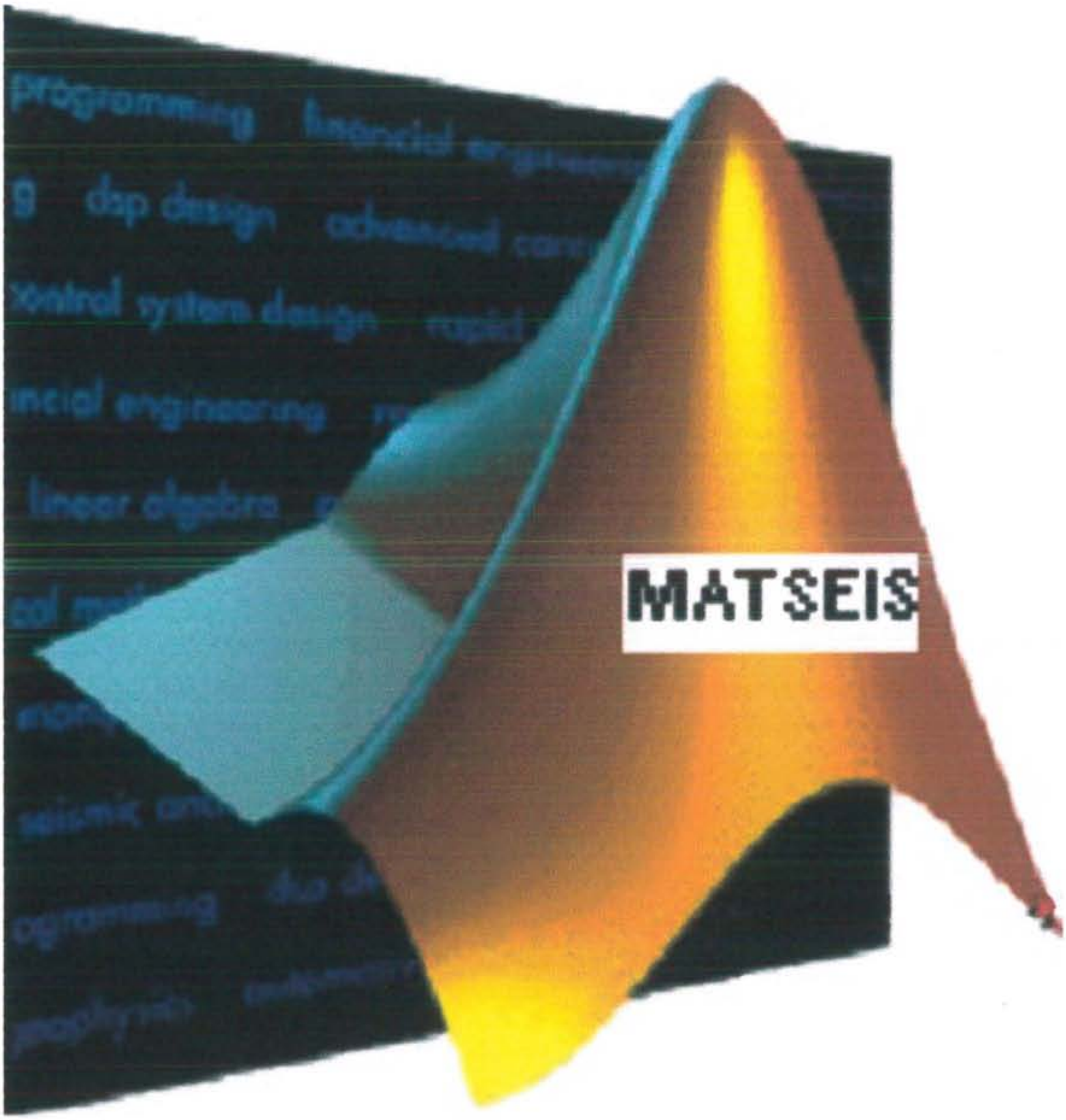


In idealized case the signal is considered as a bell shaped. It is clearly understood that practically along with the signal there is a noise at various extent. In order to remove the noise we apply filters with different orders (depending upon the type of the noise), the function of the orders is to cut the slopes (containing noise) of the sample at various extent. The figure clearly shows the different responses with the orders of the filter. The increase in the order of the filter changes the amplitude response of the sample. Higher the order of the filter, sharper it will cut the slopes, another limitation is that with the increase in the order the filter, heads of the sample (signal) increases so practically we make the balance between the heads and the slope.

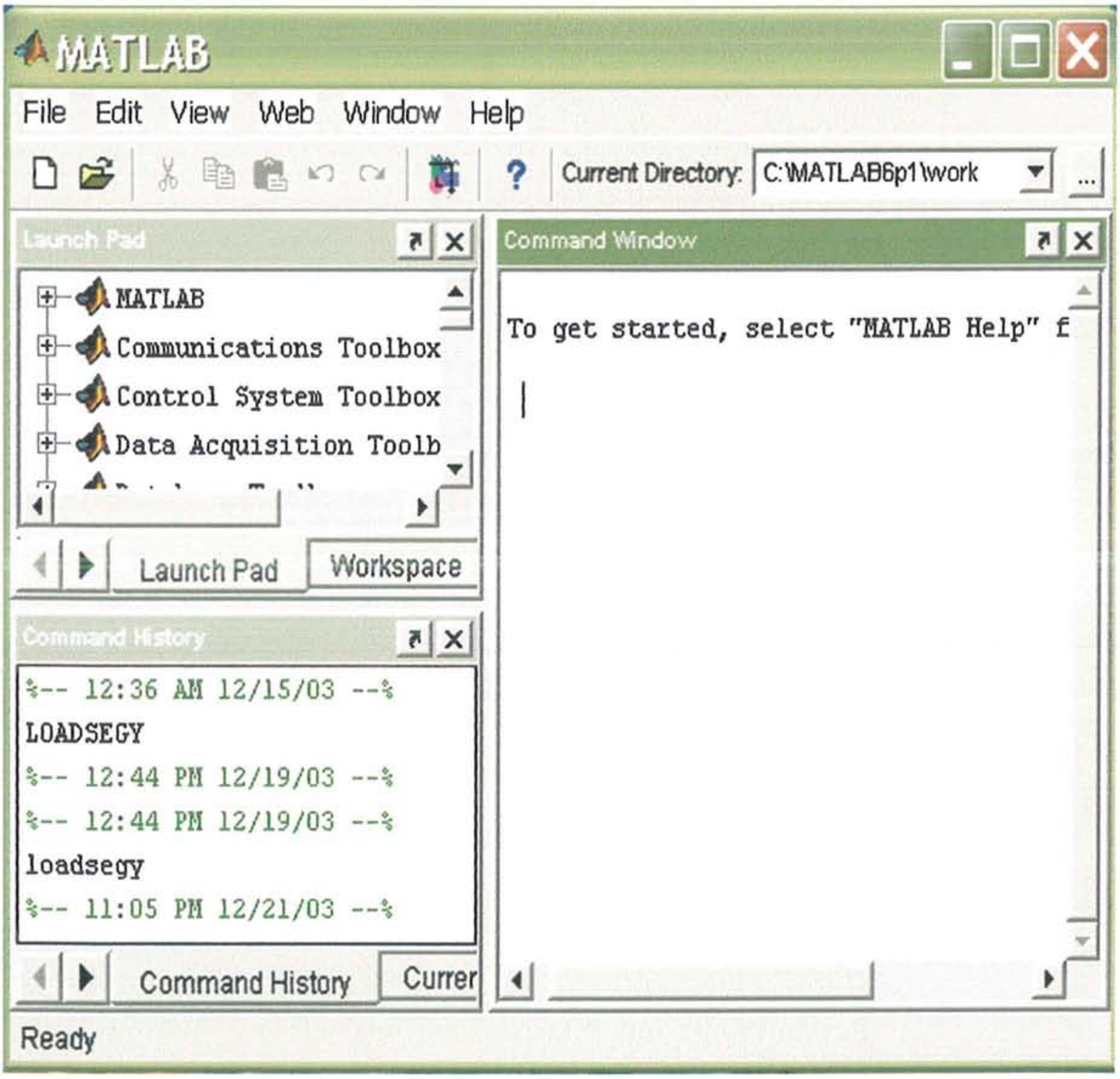
Flow chart


Define a Desired Amplitude Spectrum
Set the Phase Spectrum to Zero
Inverse Fourier Transform
Filter Operator
Input Seismic Trace
Convolve
Filtered Output

Zero Phase Frequency Filter and its applications in time domain.


programming financial engineering
g dsp design advanced
control system design rapid
ncial engineering
linear algebra
seismic
ogramming
geophysics
MATSEIS

The figure shows the opened page of the Command Window of the Matlab.

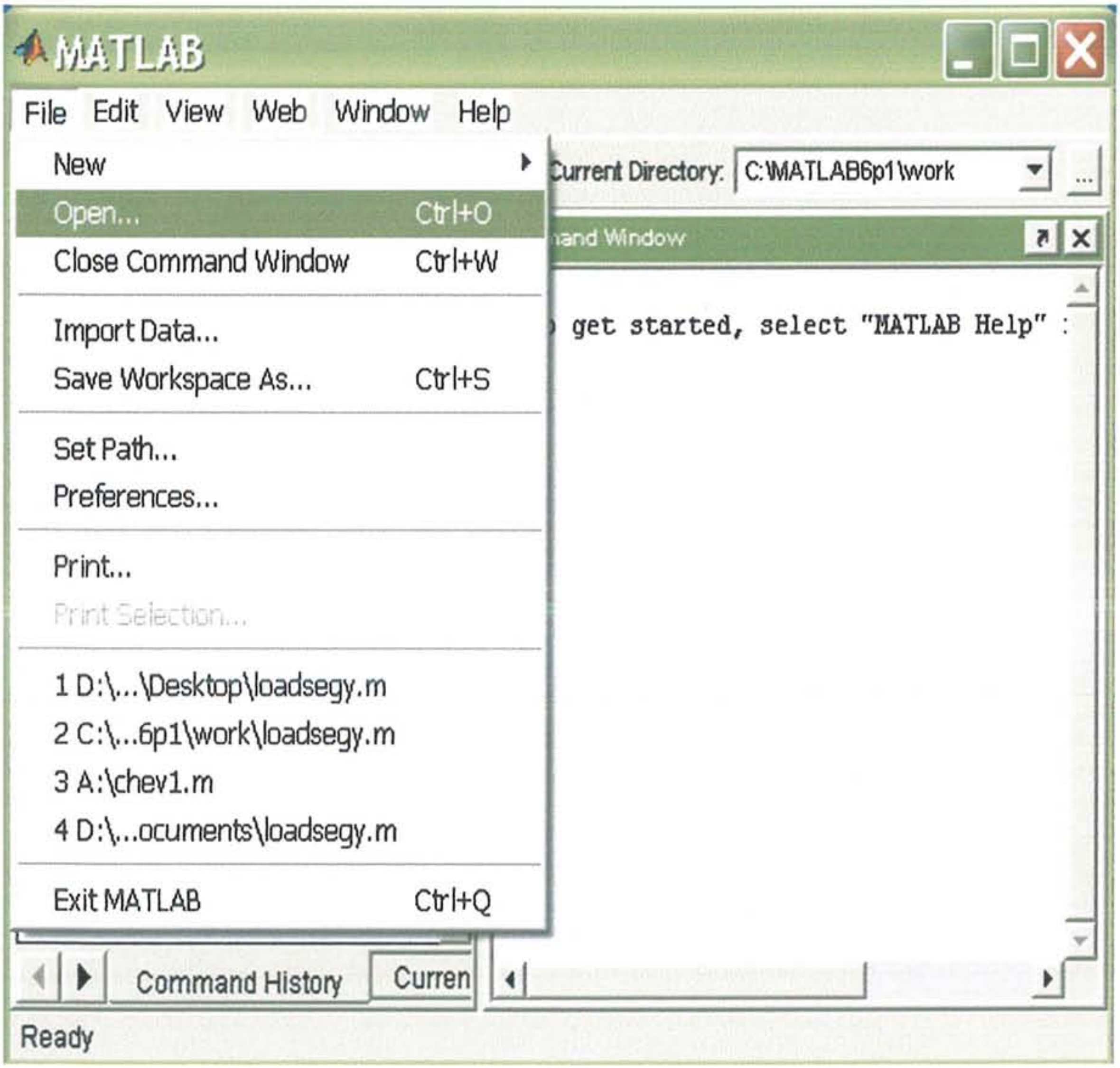


Current Directory: C:\MATLAB6p1\work

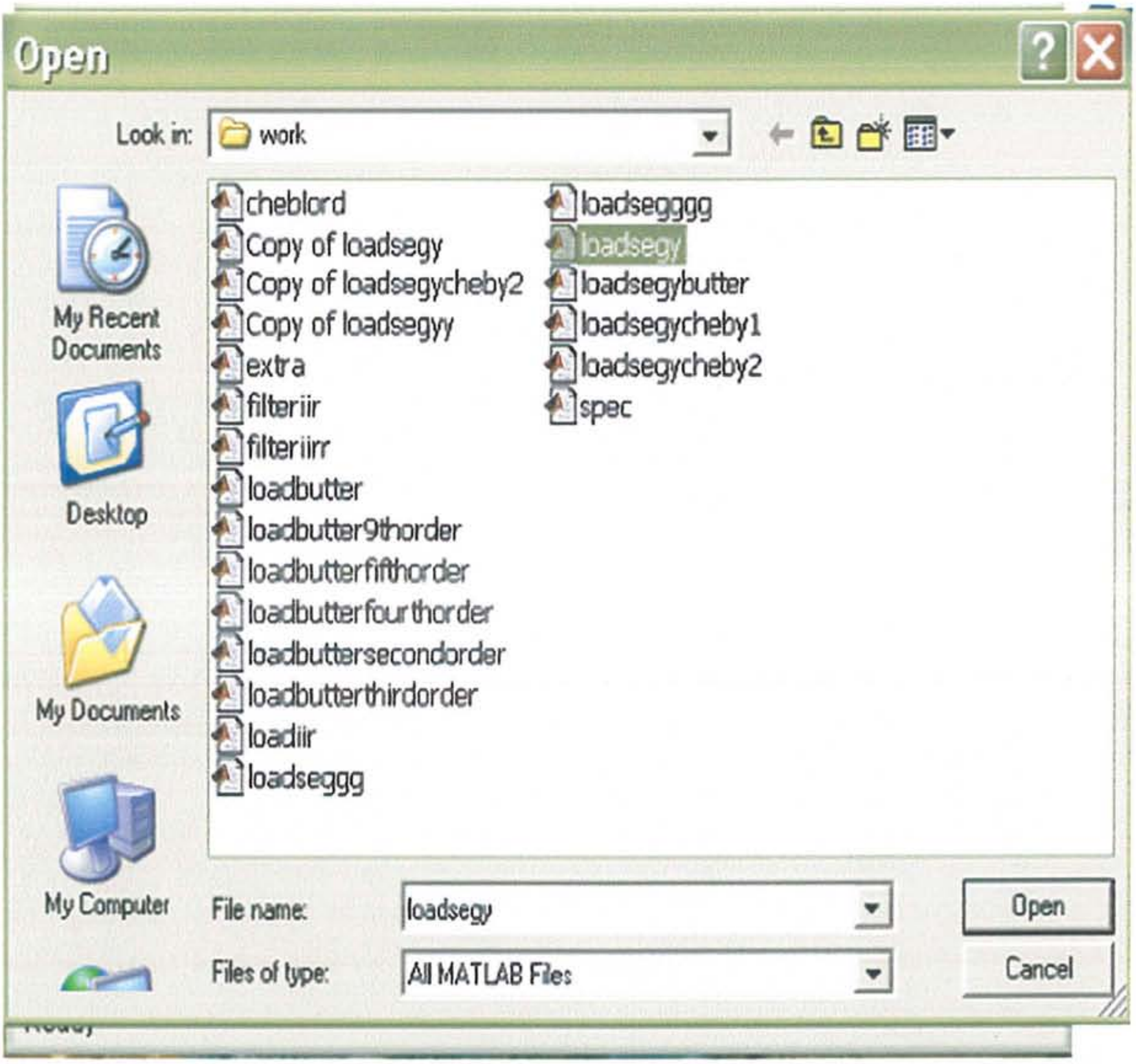
```
To get started, select "MATLAB Help" f
```

```
%-- 12:36 AM 12/15/03 --%
LOADSEGY
%-- 12:44 PM 12/19/03 --%
%-- 12:44 PM 12/19/03 --%
loadsegy
%-- 11:05 PM 12/21/03 --%
```

Use the File menu to navigate to your required file.


MATLAB
File
Edit
View
Web
Window
Help
New
Open... Ctrl+O
Close Command Window Ctrl+W
Import Data...
Save Workspace As... Ctrl+S
Set Path...
Preferences...
Print...
Print Selection...
1 D:\...\Desktop\loadsegy.m
2 C:\...6p1\work\loadsegy.m
3 A:\chev1.m
4 D:\...ocuments\loadsegy.m
Exit MATLAB Ctrl+Q
Current Directory: C:\MATLAB6p1\work
and Window
get started, select "MATLAB Help" :
Command History
Curren
Ready

Brows through the content of the following navigation screen to select your required file.

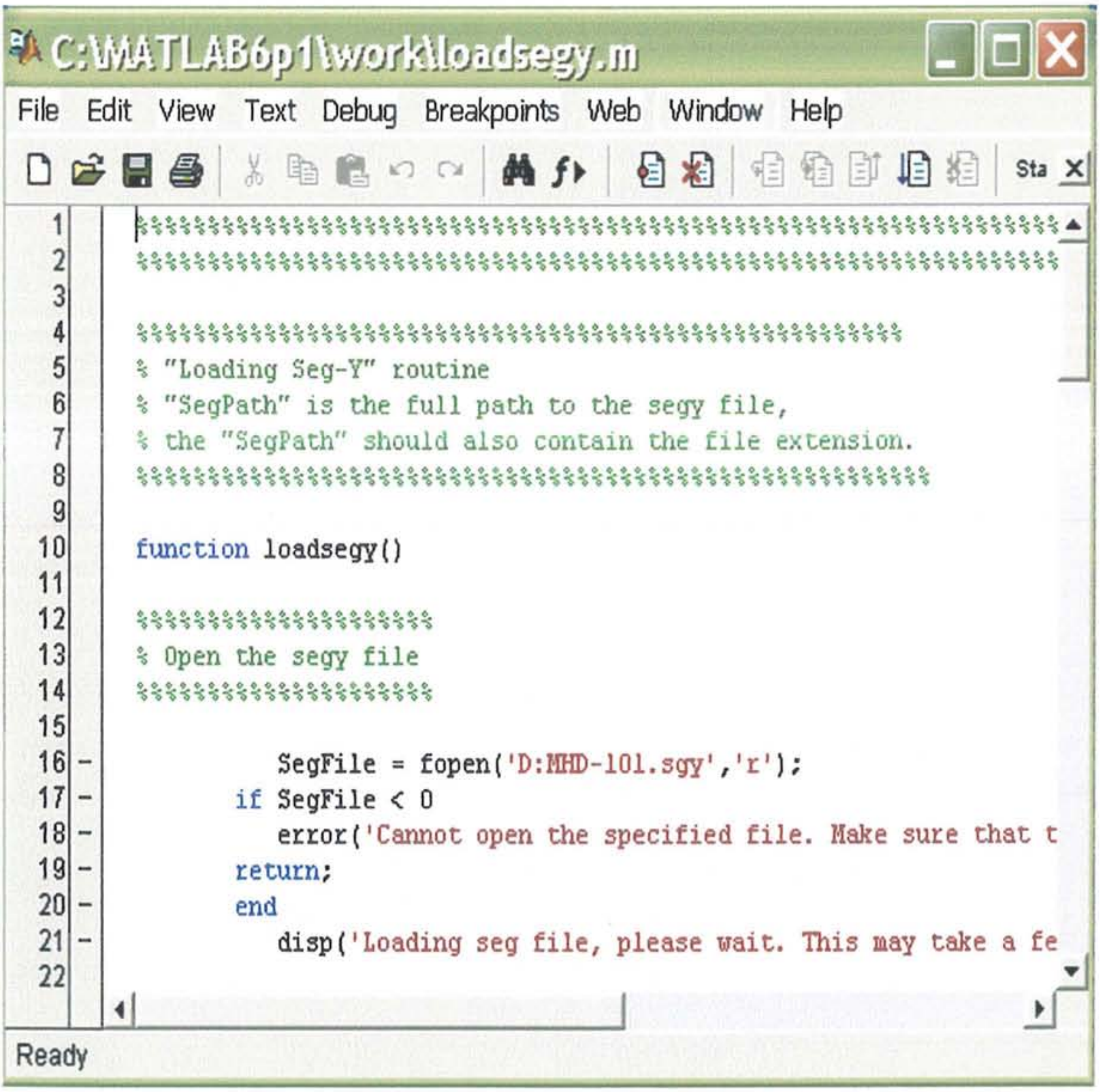


Look in: work

File name: loadsegy

Files of type: All MATLAB Files

This is the resultant file contents. Here you can move up, down, right and left using the scroll bars of the screen to view the contents of the file from all perspectives, because it don't show entire screen in one single view, cause file can extend in code horizontally and definitely vertically (cause approximately all files are often more than one page)



```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% "Loading Seg-Y" routine
% "SegPath" is the full path to the segy file,
% the "SegPath" should also contain the file extension.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function loadsegy()

%%%%%%%%%%%%%%%%%%%%%%
% Open the segy file
%%%%%%%%%%%%%%%%%%%%%%

          SegFile = fopen('D:MHD-101.sgy','r');
       if SegFile < 0
          error('Cannot open the specified file. Make sure that t
       return;
       end
          disp('Loading seg file, please wait. This may take a fe

```

Now you have two ways to run this program

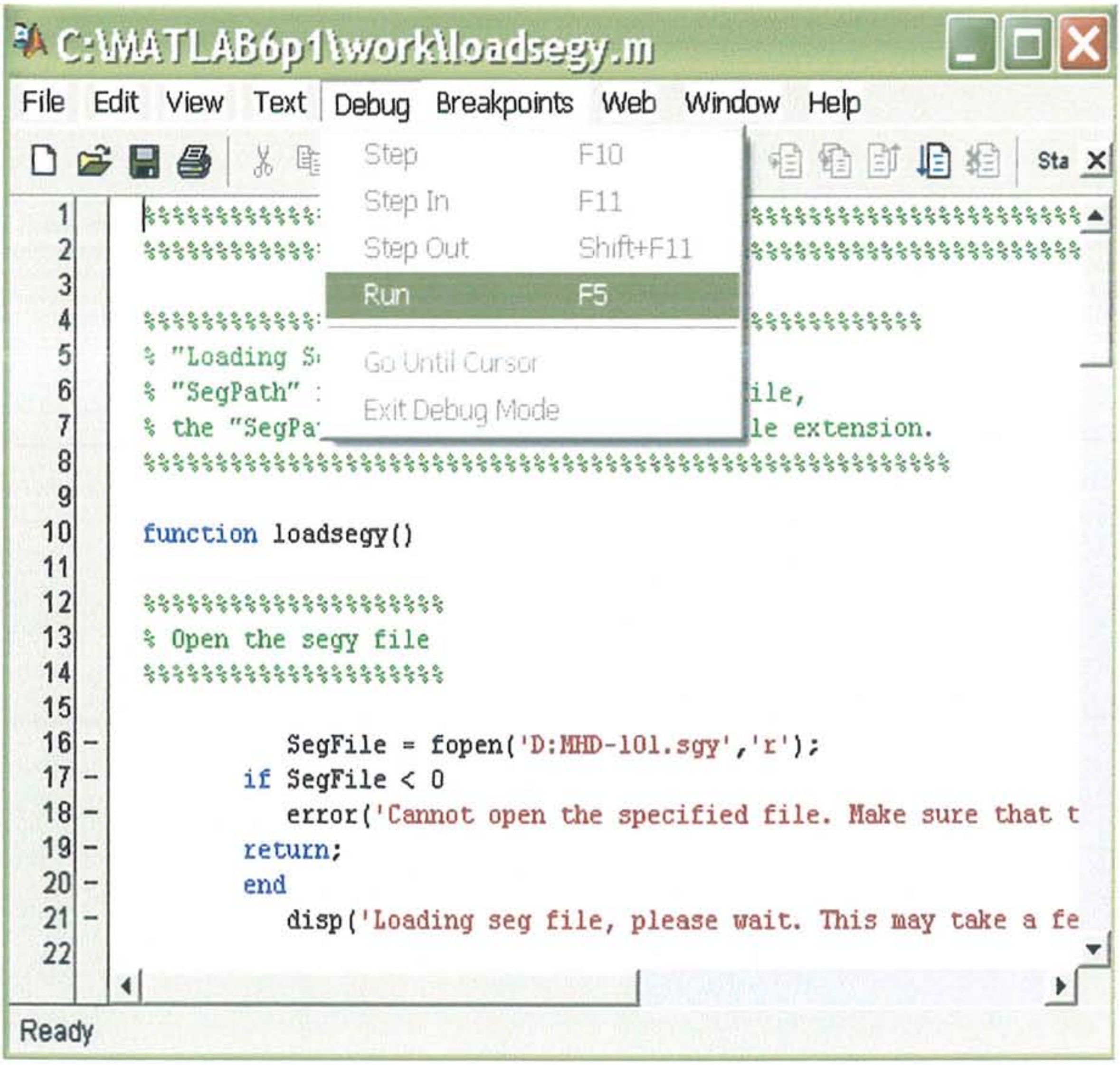
Ways are:

1. Use the short cut keys:

Using the short cut key F5; it let you to run this program directly without going into the menu details and whatever else.

2. Go through the "Debug Menu":

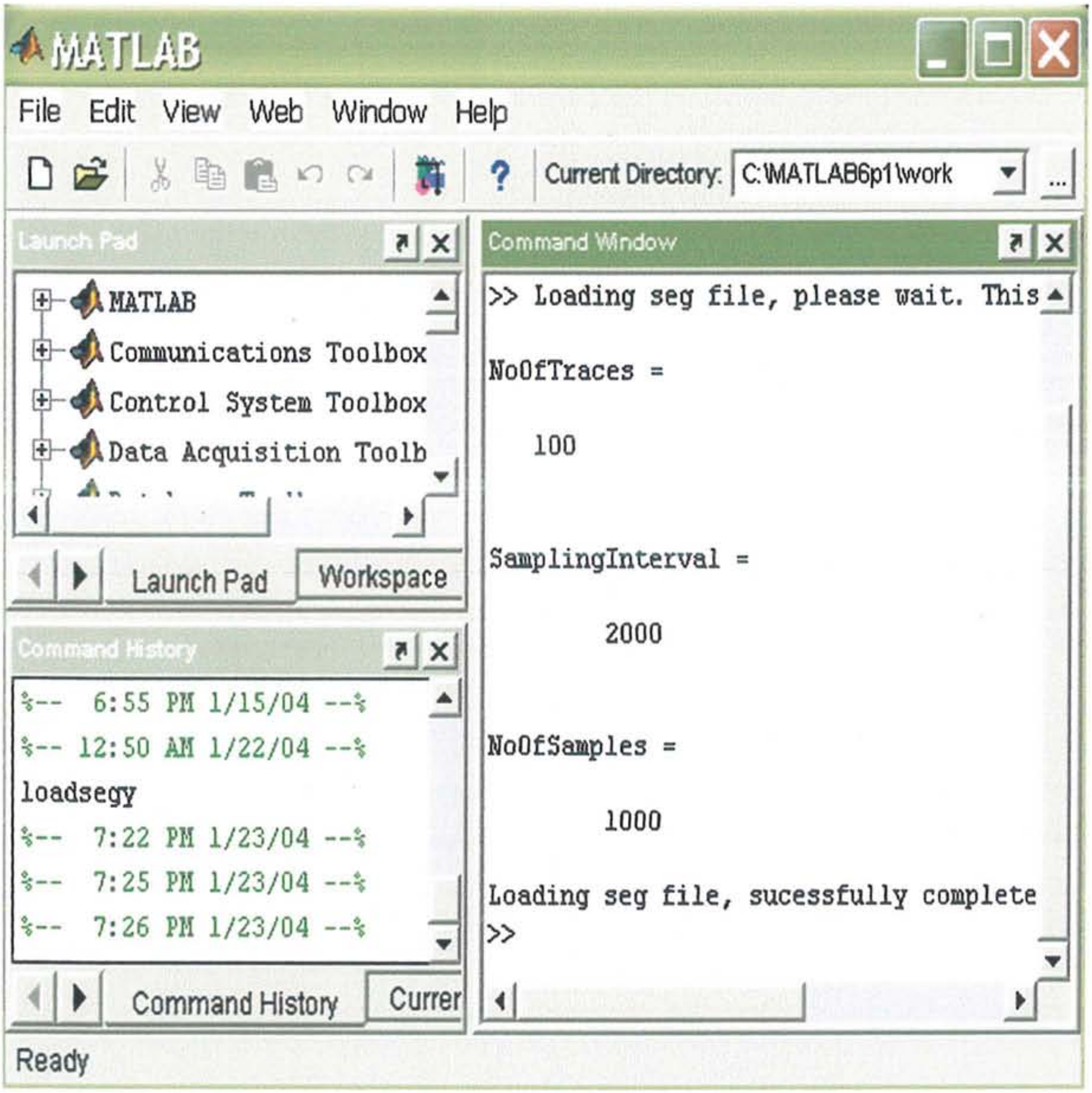
Using the Debug menu's subcommand "Run" You can let MATLAB to run this program for you



```
%%%%%%%%%%%%                                      %%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%                                      %%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%                                      %%%%%%%%%%%%
% "Loading S
% "SegPath" :                                     ile,
% the "SegPa                                      le extension.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function loadsegy()

%%%%%%%%%%%%%%%%%%%%%
% Open the segy file
%%%%%%%%%%%%%%%%%%%%%

          SegFile = fopen('D:MHD-101.sgy','r');
       if SegFile < 0
          error('Cannot open the specified file. Make sure that t
       return;
       end
          disp('Loading seg file, please wait. This may take a fe
```

Following is the command window that will show the following mentioned parameters of the file

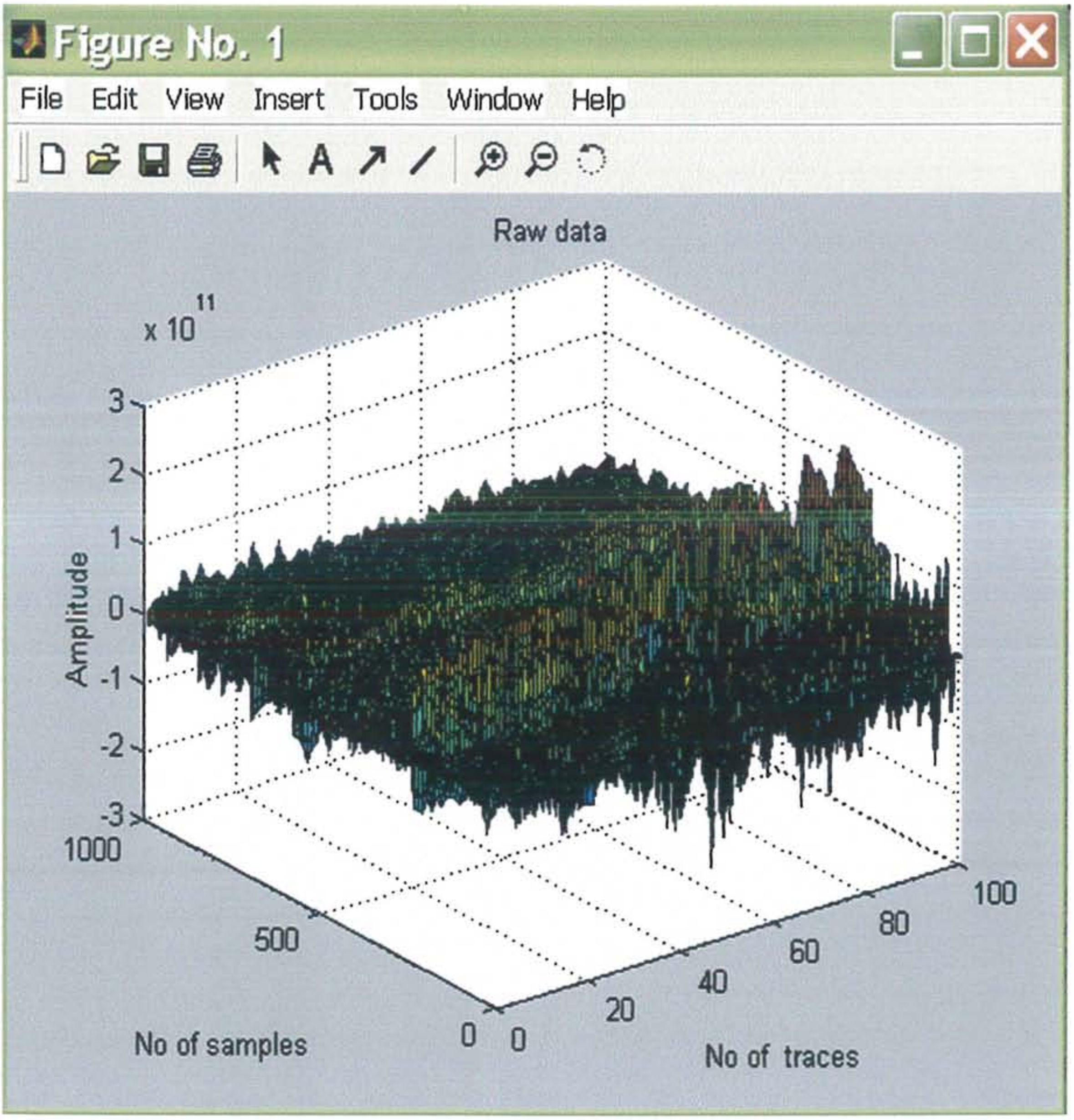
1. Number of traces
2. No of samples
3. Sampling interval



```
%--   6:55 PM 1/15/04 --%
%-- 12:50 AM 1/22/04 --%
loadsegy
%--   7:22 PM 1/23/04 --%
%--   7:25 PM 1/23/04 --%
%--   7:26 PM 1/23/04 --%
```

```
>> Loading seg file, please wait. This

NoOfTraces =

   100


SamplingInterval =

        2000


NoOfSamples =

        1000

Loading seg file, sucessfully complete
>>
```

Following figure shows the raw data in the graphical representation form that is easier to understand and analyze for the viewer. It gives us 3D view for better perception. It shows No of traces, no of samples and Amplitude to its coordinate axis.



CHAPTER 4

SOFTWARE CODING

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% "Loading Seg-Y" routine
% "SegPath" is the full path to the segy file,
% the "SegPath" should also contain the file extension.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function loadsegy()

%%%%%%%%%%%%
% Open the segy file
%%%%%%%%%%%%

SegFile = fopen('D:MHD-101.sgy','r');
if SegFile < 0
error('Cannot open the specified file. Make sure that the path is correct and contains the file extension')
return;
end
disp('Loading seg file, please wait. This may take a few minutes...')

%%%%%%%%%%%%%%%%%%%%%%%%
% Reading the number of traces parameter
%%%%%%%%%%%%%%%%%%%%%%%%%

fseek(SegFile,3212,'bof');
TempMatrix = fread(SegFile,2);
NoOfTraces = bitshift(TempMatrix(1,1),8) + TempMatrix(2,1);
NoOfTraces = 100
if NoOfTraces <= 0
```

```
error ('Enter the Number of traces, at byte location 3213.')
end

%%%%%%%%%%%%%%%%%%%%%%%%
% Reading the sampling interval parameter
%%%%%%%%%%%%%%%%%%%%%%%%

fseek(SegFile,3216,'bof');
TempMatrix = fread(SegFile,2);
SamplingInterval = bitshift(TempMatrix(1,1),8) + TempMatrix(2,1)
if SamplingInterval <= 0
error ('Enter Sampling Interval, at byte location 3217.')
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Reading the  number of number of samples per trace parameter
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

fseek(SegFile,3220,'bof');
TempMatrix = fread(SegFile,2);
NoOfSamples = bitshift(TempMatrix(1,1),8) + TempMatrix(2,1)
if NoOfSamples <= 0
error ('Enter the Number of traces, at byte location 3221.')
end

%%%%%%%%%%%%%%%%%%%%%%
% Reading the Traces
% The Matrix holding the traces is "TraceMatrix"
%%%%%%%%%%%%%%%%%%%%%%%

TraceMatrix = zeros(NoOfSamples,NoOfTraces);
fseek(SegFile,3840,'bof');
```

```
%%%%%%%%%%%%%
%For number of traces
%%%%%%%%%%%%%

for i = 1 : NoOfTraces

%%%%%%%%%%%%%%%
%For Number of Samples
%%%%%%%%%%%%%%%

for j = 1 : NoOfSamples
TempMatrix = fread(SegFile,4);
SignValue = bitand(TempMatrix(1,1),128);
if SignValue == 128;
Sign = -1;
else
Sign = 1;
end
Char = bitand(TempMatrix(1,1),127);
Charac = 16^(Char-64);
Fraction=     bitshift(TempMatrix(2,1),16)     +     bitshift(TempMatrix(3,1),8)     +
TempMatrix(4,1);
SampleValue = Sign*Charac*Fraction;
TraceMatrix(j,i) = SampleValue;
end
fseek(SegFile,240,'cof');
end
TraceMatrix;
disp('Loading seg file, sucessfully completed.')
figure
surf(TraceMatrix)
title('Raw data')
xlabel('No of  traces')
ylabel('No of samples')
```

```
zlabel('Amplitude')

%%%%%%%%%%%%%
% Closing the segy file
%%%%%%%%%%%%%%

fileclose = fclose(SegFile);
if fileclose < 0
error('Error while closing the file')
return;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% "Loading Seg-Y" routine and filtering the data through ButterWorth 1st order
% "SegPath" is the full path to the segy file,
% the "SegPath" should also contain the file extension
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function loadsegy()

%%%%%%%%%%%%
% Open the segy file.
%%%%%%%%%%%%

SegFile = fopen('D:MHD-101.sgy','r');
if SegFile < 0
error('Cannot open the specified file. Make sure that the path is correct and contains the file extension')
return;
end
disp('Loading seg file, please wait. This may take a few minutes...')

%%%%%%%%%%%%%%%%%%%%%%%%
% Reading the number of traces parameter
%%%%%%%%%%%%%%%%%%%%%%%%

fseek(SegFile,3212,'bof');
TempMatrix = fread(SegFile,2);
NoOfTraces = bitshift(TempMatrix(1,1),8) + TempMatrix(2,1);
NoOfTraces = 100
if NoOfTraces <= 0
error ('Enter the Number of traces, at byte location 3213.')
```

```
end

%%%%%%%%%%%%%%%%%%%%%%%%%
% Reading the sampling interval parameter
%%%%%%%%%%%%%%%%%%%%%%%%%

fseek(SegFile,3216,'bof');
TempMatrix = fread(SegFile,2);
SamplingInterval = bitshift(TempMatrix(1,1),8) + TempMatrix(2,1)
if SamplingInterval <= 0
error ('Enter Sampling Interval, at byte location 3217.')
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Reading the  number of number of samples per trace parameter
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

fseek(SegFile,3220,'bof');
TempMatrix = fread(SegFile,2);
NoOfSamples = bitshift(TempMatrix(1,1),8) + TempMatrix(2,1)
if NoOfSamples <= 0
error ('Enter the Number of traces, at byte location 3221.')
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Reading the Traces
% The Matrix holding the traces is "TraceMatrix"
%%%%%%%%%%%%%%%%%%%%%%%%%%%

TraceMatrix = zeros(NoOfSamples,NoOfTraces);
fseek(SegFile,3840,'bof');
%%%%%%%%%%%%%%
%For number of traces
%%%%%%%%%%%%%%
```

```
for i = 1 : NoOfTraces

%%%%%%%%%%%%%%
%For Number of Samples
%%%%%%%%%%%%%%%

for j = 1 : NoOfSamples
TempMatrix = fread(SegFile,4);
SignValue = bitand(TempMatrix(1,1),128);
if SignValue == 128;
Sign = -1;
else
Sign = 1;
end
Char = bitand(TempMatrix(1,1),127);
Charac = 16^(Char-64);
Fraction=      bitshift(TempMatrix(2,1),16)      +      bitshift(TempMatrix(3,1),8)      +
TempMatrix(4,1);
SampleValue = Sign*Charac*Fraction;
TraceMatrix(j,i) = SampleValue;
end
fseek(SegFile,240,'cof');
end
TraceMatrix;
disp('Loading seg file, sucessfully completed.')
figure
surf(TraceMatrix)
title('Raw')
xlabel('No of traces')
ylabel('No of samples')

%%%%%%%%%%%%%
% Closing the segy file
%%%%%%%%%%%%%%

fileclose = fclose(SegFile);
```

```
if fileclose < 0
error('Error while closing the file')
return;
end

disp('Filtering the data. PLease wait...')

%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%ButterWorth%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%

end
[b,a] = Butter(1,15/100);
for i = 1 : NoOfTraces
TempTrace =  conv(a,TraceMatrix(:,i));
FilteredMatrix(:,i)=TempTrace;
end
disp('.....Filtering complete.')
figure
surf(FilteredMatrix)
title('Filtered through BW 1st order')
ylabel('No of samples')
xlabel('No of traces')
zlabel('Amplitude')
return
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%"Loading Seg-Y" routine and filtering the data through ButterWorth 2nd order
filter
% "SegPath" is the full path to the segy file,
% the "SegPath" should also contain the file extension
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function loadsegy()

%%%%%%%%%%%%
% Open the segy file
%%%%%%%%%%%%

SegFile = fopen('D:MHD-101.sgy','r');
if SegFile < 0
error('Cannot open the specified file. Make sure that the path is correct and
contains the file extension')
return;
end

disp('Loading seg file, please wait. This may take a few minutes...')

%%%%%%%%%%%%%%%%%%%%%%%
% Reading the number of traces parameter
%%%%%%%%%%%%%%%%%%%%%%%%

fseek(SegFile,3212,'bof');
TempMatrix = fread(SegFile,2);
NoOfTraces = bitshift(TempMatrix(1,1),8) + TempMatrix(2,1);
NoOfTraces = 100
if NoOfTraces <= 0
error ('Enter the Number of traces, at byte location 3213.')
```

```
end

%%%%%%%%%%%%%%%%%%%%%%%%
% Reading the sampling interval parameter
%%%%%%%%%%%%%%%%%%%%%%%%%


fseek(SegFile,3216,'bof');
TempMatrix = fread(SegFile,2);
SamplingInterval = bitshift(TempMatrix(1,1),8) + TempMatrix(2,1)
if SamplingInterval <= 0
error ('Enter Sampling Interval, at byte location 3217.')
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Reading the  number of number of samples per trace parameter
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

fseek(SegFile,3220,'bof');
TempMatrix = fread(SegFile,2);
NoOfSamples = bitshift(TempMatrix(1,1),8) + TempMatrix(2,1)
if NoOfSamples <= 0
error ('Enter the Number of traces, at byte location 3221.')
end

%%%%%%%%%%%%%%%%%%%%%%%%%%
% Reading the Traces
% The Matrix holding the traces is "TraceMatrix"
%%%%%%%%%%%%%%%%%%%%%%%%%%

TraceMatrix = zeros(NoOfSamples,NoOfTraces);
fseek(SegFile,3840,'bof');
```

```
%%%%%%%%%%%%%%
%For number of traces
%%%%%%%%%%%%%%%

for i = 1 : NoOfTraces

%%%%%%%%%%%%%%%%
%For Number of Samples
%%%%%%%%%%%%%%%%

for j = 1 : NoOfSamples
TempMatrix = fread(SegFile,4);
SignValue = bitand(TempMatrix(1,1),128);
if SignValue == 128;
Sign = -1;
else
Sign = 1;
end
Char = bitand(TempMatrix(1,1),127);
Charac = 16^(Char-64);
Fraction=     bitshift(TempMatrix(2,1),16)     +     bitshift(TempMatrix(3,1),8)     +
TempMatrix(4,1);
SampleValue = Sign*Charac*Fraction;
TraceMatrix(j,i) = SampleValue;
end
fseek(SegFile,240,'cof');
end
TraceMatrix;
disp('Loading seg file, sucessfully completed.')
figure
surf(TraceMatrix)
title('Raw')
%%%%%%%%%%%%%%
% Closing the segy file
%%%%%%%%%%%%%%%
```

```
fileclose = fclose(SegFile);
if fileclose < 0
error('Error while closing the file')
return;
end
disp('Filtering the data. PLease wait...')

%%%%%%%%%%%%%%%%%%%%
%%%%%ButterWorth%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%

[a,b]= butter(2,15/100);
for i = 1 : NoOfTraces
TempTrace =  conv(a,TraceMatrix(:,i));
FilteredMatrix(:,i)=TempTrace;
end
disp('.....Filtering complete.')
figure
surf(FilteredMatrix)
title('Filtered through BW 2nd order')
xlabel('No of traces')
ylabel('No of samples')
zlabel('Amplitude')
return

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%"Loading Seg-Y" routine and filtering the data with buterworth (Third order) filter.
% "SegPath" is the full path to the segy file,
% the "SegPath" should also contain the file extension
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function loadsegy()

%%%%%%%%%%%%
% Open the segy file
%%%%%%%%%%%%

SegFile = fopen('D:MHD-101.sgy','r');
if SegFile < 0
error('Cannot open the specified file. Make sure that the path is correct and
contains the file extension')
return;
end
disp('Loading seg file, please wait. This may take a few minutes...')

%%%%%%%%%%%%%%%%%%%%%%%%
% Reading the number of traces parameter
%%%%%%%%%%%%%%%%%%%%%%%%%

fseek(SegFile,3212,'bof');
TempMatrix = fread(SegFile,2);
NoOfTraces = bitshift(TempMatrix(1,1),8) + TempMatrix(2,1);
NoOfTraces = 100
if NoOfTraces <= 0
error ('Enter the Number of traces, at byte location 3213.')
end
```

```
%%%%%%%%%%%%%%%%%%%%%%%
% Reading the sampling interval parameter
%%%%%%%%%%%%%%%%%%%%%%%%

fseek(SegFile,3216,'bof');
TempMatrix = fread(SegFile,2);
SamplingInterval = bitshift(TempMatrix(1,1),8) + TempMatrix(2,1)
if SamplingInterval <= 0
error ('Enter Sampling Interval, at byte location 3217.')
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Reading the  number of number of samples per trace parameter
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

fseek(SegFile,3220,'bof');
TempMatrix = fread(SegFile,2);
NoOfSamples = bitshift(TempMatrix(1,1),8) + TempMatrix(2,1)
if NoOfSamples <= 0
error ('Enter the Number of traces, at byte location 3221.')
end

%%%%%%%%%%%%%%%%%%%%%%%%%%
% Reading the Traces
% The Matrix holding the traces is "TraceMatrix"
%%%%%%%%%%%%%%%%%%%%%%%%%%%

TraceMatrix = zeros(NoOfSamples,NoOfTraces);
fseek(SegFile,3840,'bof');

%%%%%%%%%%%%
For number of traces
%%%%%%%%%%%%%


for i = 1 : NoOfTraces
```

```
%%%%%%%%%%%%%%
%For Number of Samples
%%%%%%%%%%%%%%%

for j = 1 : NoOfSamples
TempMatrix = fread(SegFile,4);
SignValue = bitand(TempMatrix(1,1),128);
if SignValue == 128;
Sign = -1;
else
Sign = 1;
end
Char = bitand(TempMatrix(1,1),127);
Charac = 16^(Char-64);
Fraction=      bitshift(TempMatrix(2,1),16)      +      bitshift(TempMatrix(3,1),8)      +
TempMatrix(4,1);
SampleValue = Sign*Charac*Fraction;
TraceMatrix(j,i) = SampleValue;
end
fseek(SegFile,240,'cof');
end
TraceMatrix;
disp('Loading seg file, sucessfully completed.')
figure
surf(TraceMatrix)
title('Raw')

%%%%%%%%%%%%%
% Closing the segy file
%%%%%%%%%%%%%%

fileclose = fclose(SegFile);
if fileclose < 0
error('Error while closing the file')
```

```
return;
end
disp('Filtering the data. PLease wait...')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%ButterWorth (Third Order) Filter%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%

[a,b]= butter(3,15/100);
for i = 1 : NoOfTraces
TempTrace =  conv(a,TraceMatrix(:,i));
FilteredMatrix(:,i)=TempTrace;
end
disp('.....Filtering complete.')
figure
surf(FilteredMatrix)
title('Filtered through BW 3rd order')
xlabel('No of traces')
ylabel('No of samples')
zlabel('Amplitude')
return

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% "Loading Seg-Y" routine and filtering the data with ButerWorth fourth (order
filter)filter
% "SegPath" is the full path to the segy file,
% the "SegPath" should also contain the file extension
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function loadsegy()

%%%%%%%%%%%%
% Open the segy file
%%%%%%%%%%%%

SegFile = fopen('D:MHD-101.sgy','r');
if SegFile < 0
error('Cannot open the specified file. Make sure that the path is correct and
contains the file extension')
return;
end
disp('Loading seg file, please wait. This may take a few minutes...')

%%%%%%%%%%%%%%%%%%%%%%%%
% Reading the number of traces parameter
%%%%%%%%%%%%%%%%%%%%%%%%

fseek(SegFile,3212,'bof');
TempMatrix = fread(SegFile,2);
NoOfTraces = bitshift(TempMatrix(1,1),8) + TempMatrix(2,1);
NoOfTraces = 100


if NoOfTraces <= 0
```

```
error ('Enter the Number of traces, at byte location 3213.')
end

%%%%%%%%%%%%%%%%%%%%%%%%
% Reading the sampling interval parameter
%%%%%%%%%%%%%%%%%%%%%%%%%

fseek(SegFile,3216,'bof');
TempMatrix = fread(SegFile,2);
SamplingInterval = bitshift(TempMatrix(1,1),8) + TempMatrix(2,1)
if SamplingInterval <= 0
error ('Enter Sampling Interval, at byte location 3217.')
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Reading the  number of number of samples per trace parameter
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

fseek(SegFile,3220,'bof');
TempMatrix = fread(SegFile,2);
NoOfSamples = bitshift(TempMatrix(1,1),8) + TempMatrix(2,1)
if NoOfSamples <= 0
error ('Enter the Number of traces, at byte location 3221.')
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Reading the Traces
% The Matrix holding the traces is "TraceMatrix"
%%%%%%%%%%%%%%%%%%%%%%%%%%%%

TraceMatrix = zeros(NoOfSamples,NoOfTraces);
fseek(SegFile,3840,'bof');
```

```
%%%%%%%%%%%%%%
%For number of traces
%%%%%%%%%%%%%%

for i = 1 : NoOfTraces

%%%%%%%%%%%%%%%
%For Number of Samples
%%%%%%%%%%%%%%%%

for j = 1 : NoOfSamples
TempMatrix = fread(SegFile,4);
SignValue = bitand(TempMatrix(1,1),128);
if SignValue == 128;
Sign = -1;
else
Sign = 1;
end
Char = bitand(TempMatrix(1,1),127);
Charac = 16^(Char-64);
Fraction=     bitshift(TempMatrix(2,1),16)     +     bitshift(TempMatrix(3,1),8)     +
TempMatrix(4,1);
SampleValue = Sign*Charac*Fraction;
TraceMatrix(j,i) = SampleValue;
end
fseek(SegFile,240,'cof');
end
TraceMatrix;
disp('Loading seg file, sucessfully completed.')
figure
surf(TraceMatrix)
title('Raw')

%%%%%%%%%%%%%%
% Closing the segy file
%%%%%%%%%%%%%%
```

```
fileclose = fclose(SegFile);
if fileclose < 0
error('Error while closing the file')
return;
end
disp('Filtering the data. PLease wait...')

%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%ButterWorth%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%

[a,b]= butter(4,15/100);
for i = 1 : NoOfTraces
TempTrace =  conv(a,TraceMatrix(:,i));
FilteredMatrix(:,i)=TempTrace;
end
disp('.....Filtering complete.')
figure
surf(FilteredMatrix)
title('Filtered through BW 4th order')
xlabel('No of traces')
ylabel('No of samples')
zlabel('Amplitude')
return

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% "Loading Seg-Y" routine and filtering the data with Buterworth (fifth order) Filter
% "SegPath" is the full path to the segy file,
% the "SegPath" should also contain the file extension
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function loadsegy()

%%%%%%%%%%%%
% Open the segy file
%%%%%%%%%%%%

SegFile = fopen('D:MHD-101.sgy','r');
if SegFile < 0
error('Cannot open the specified file. Make sure that the path is correct and
contains the file extension')
return;
end
disp('Loading seg file, please wait. This may take a few minutes...')

%%%%%%%%%%%%%%%%%%%%%%%%
% Reading the number of traces parameter
%%%%%%%%%%%%%%%%%%%%%%%%

fseek(SegFile,3212,'bof');
TempMatrix = fread(SegFile,2);
NoOfTraces = bitshift(TempMatrix(1,1),8) + TempMatrix(2,1);
NoOfTraces = 100
if NoOfTraces <= 0
error ('Enter the Number of traces, at byte location 3213.')
```

```
end

%%%%%%%%%%%%%%%%%%%%%%%%
% Reading the sampling interval parameter
%%%%%%%%%%%%%%%%%%%%%%%%

fseek(SegFile,3216,'bof');
TempMatrix = fread(SegFile,2);
SamplingInterval = bitshift(TempMatrix(1,1),8) + TempMatrix(2,1)
if SamplingInterval <= 0
error ('Enter Sampling Interval, at byte location 3217.')
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Reading the  number of number of samples per trace parameter
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

fseek(SegFile,3220,'bof');
TempMatrix = fread(SegFile,2);
NoOfSamples = bitshift(TempMatrix(1,1),8) + TempMatrix(2,1)
if NoOfSamples <= 0
error ('Enter the Number of traces, at byte location 3221.')
end

%%%%%%%%%%%%%%%%%%%%%%%%%%
% Reading the Traces
% The Matrix holding the traces is "TraceMatrix"
%%%%%%%%%%%%%%%%%%%%%%%%%%%

TraceMatrix = zeros(NoOfSamples,NoOfTraces);
fseek(SegFile,3840,'bof');

%%%%%%%%%%%%%%
%For number of traces
%%%%%%%%%%%%%%
```

```
for i = 1 : NoOfTraces

%%%%%%%%%%%%%%%%
%For Number of Samples
%%%%%%%%%%%%%%%%

for j = 1 : NoOfSamples
TempMatrix = fread(SegFile,4);
SignValue = bitand(TempMatrix(1,1),128);
if SignValue == 128;
Sign = -1;
else
Sign = 1;
end
Char = bitand(TempMatrix(1,1),127);
Charac = 16^(Char-64);
Fraction=     bitshift(TempMatrix(2,1),16)     +     bitshift(TempMatrix(3,1),8)     +
TempMatrix(4,1);
SampleValue = Sign*Charac*Fraction;
TraceMatrix(j,i) = SampleValue;
end
fseek(SegFile,240,'cof');
end
TraceMatrix;
disp('Loading seg file, sucessfully completed.')
figure
surf(TraceMatrix)
title('Raw')

%%%%%%%%%%%%%%%
% Closing the segy file
%%%%%%%%%%%%%%%

fileclose = fclose(SegFile);
if fileclose < 0
error('Error while closing the file')
```

```
return;
end
disp('Filtering the data. PLease wait...')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%ButterWorth (Fifth Order)Filter%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

[a,b]= butter(5,15/100);
for i = 1 : NoOfTraces
TempTrace =  conv(a,TraceMatrix(:,i));
FilteredMatrix(:,i)=TempTrace;
end
disp('.....Filtering complete.')
figure
surf(FilteredMatrix)
title('Filtered through BW 5th order')
xlabel('No of traces')
ylabel('No of samples')
zlabel('Amplitude')
return

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% "Loading Seg-Y" routine and filtering the data with Butterworth (Ninth order)
filter
% "SegPath" is the full path to the segy file,
% the "SegPath" should also contain the file extension
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function loadsegy()

%%%%%%%%%%%%%
% Open the segy file
%%%%%%%%%%%%%
SegFile = fopen('D:MHD-101.sgy','r');
if SegFile < 0
error('Cannot open the specified file. Make sure that the path is correct and
contains the file extension')
return;
end
disp('Loading seg file, please wait. This may take a few minutes...')

%%%%%%%%%%%%%%%%%%%%%%%%%%
% Reading the number of traces parameter
%%%%%%%%%%%%%%%%%%%%%%%%%%

fseek(SegFile,3212,'bof');
TempMatrix = fread(SegFile,2);
NoOfTraces = bitshift(TempMatrix(1,1),8) + TempMatrix(2,1);
NoOfTraces = 100
if NoOfTraces <= 0
error ('Enter the Number of traces, at byte location 3213.')
end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%
% Reading the sampling interval parameter
%%%%%%%%%%%%%%%%%%%%%%%%%


fseek(SegFile,3216,'bof');
TempMatrix = fread(SegFile,2);
SamplingInterval = bitshift(TempMatrix(1,1),8) + TempMatrix(2,1)
if SamplingInterval <= 0
error ('Enter Sampling Interval, at byte location 3217.')
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Reading the  number of number of samples per trace parameter
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fseek(SegFile,3220,'bof');
TempMatrix = fread(SegFile,2);
NoOfSamples = bitshift(TempMatrix(1,1),8) + TempMatrix(2,1)
if NoOfSamples <= 0
error ('Enter the Number of traces, at byte location 3221.')
end

%%%%%%%%%%%%%%%%%%%%%%%%%%
% Reading the Traces
% The Matrix holding the traces is "TraceMatrix"
%%%%%%%%%%%%%%%%%%%%%%%%%%%

TraceMatrix = zeros(NoOfSamples,NoOfTraces);
fseek(SegFile,3840,'bof');

%%%%%%%%%%%%%
%For number of traces
%%%%%%%%%%%%%

for i = 1 : NoOfTraces
```

```
%%%%%%%%%%%%%%%
%For Number of Samples
%%%%%%%%%%%%%%%%

for j = 1 : NoOfSamples
TempMatrix = fread(SegFile,4);
SignValue = bitand(TempMatrix(1,1),128);
if SignValue == 128;
Sign = -1;
else
Sign = 1;
end
Char = bitand(TempMatrix(1,1),127);
Charac = 16^(Char-64);
Fraction=      bitshift(TempMatrix(2,1),16)      +      bitshift(TempMatrix(3,1),8)      +
TempMatrix(4,1);
SampleValue = Sign*Charac*Fraction;
TraceMatrix(j,i) = SampleValue;
end
fseek(SegFile,240,'cof');
end
TraceMatrix;
disp('Loading seg file, sucessfully completed.')
figure
surf(TraceMatrix)
title('Raw data')
xlabel('No of traces')
ylabel('No of samples')
zlabel('Amplitude')

%%%%%%%%%%%%%
% Closing the segy file
%%%%%%%%%%%%%%
fileclose = fclose(SegFile);

if fileclose < 0
```

```
error('Error while closing the file')
return;
end
disp('Filtering the data. PLease wait...')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%BUTTERWORTH(NINTHORDER)FILTER%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%

[b,a] = butter(9,400/1000,'low')
for i=1 :NoOfTraces
TempTrace =conv(a,TraceMatrix(:,i));
FilteredMatrix(:,i)=TempTrace;
end

disp('.....Filtering complete.')
figure
surf(FilteredMatrix)
title('Filtered through BW 9th order ')
xlabel('No of traces')
ylabel('No of samples')
zlabel('Amplitude')
return

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% "Loading Seg-Y" routine and filtering the data with Butterworth (Tenth order)
filter
% "SegPath" is the full path to the segy file,
% the "SegPath" should also contain the file extension
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function loadsegy()

%%%%%%%%%%%%
% Open the segy file
%%%%%%%%%%%%%
SegFile = fopen('D:MHD-101.sgy','r');
if SegFile < 0
error('Cannot open the specified file. Make sure that the path is correct and
contains the file extension')
return;
end
disp('Loading seg file, please wait. This may take a few minutes...')

%%%%%%%%%%%%%%%%%%%%%%%
% Reading the number of traces parameter
%%%%%%%%%%%%%%%%%%%%%%%%

fseek(SegFile,3212,'bof');
TempMatrix = fread(SegFile,2);
NoOfTraces = bitshift(TempMatrix(1,1),8) + TempMatrix(2,1);
NoOfTraces = 100
if NoOfTraces <= 0
error ('Enter the Number of traces, at byte location 3213.')
end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%
% Reading the sampling interval parameter
%%%%%%%%%%%%%%%%%%%%%%%%%


fseek(SegFile,3216,'bof');
TempMatrix = fread(SegFile,2);
SamplingInterval = bitshift(TempMatrix(1,1),8) + TempMatrix(2,1)
if SamplingInterval <= 0
error ('Enter Sampling Interval, at byte location 3217.')
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Reading the  number of number of samples per trace parameter
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

fseek(SegFile,3220,'bof');
TempMatrix = fread(SegFile,2);
NoOfSamples = bitshift(TempMatrix(1,1),8) + TempMatrix(2,1)
if NoOfSamples <= 0
error ('Enter the Number of traces, at byte location 3221.')
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Reading the Traces
% The Matrix holding the traces is "TraceMatrix"
%%%%%%%%%%%%%%%%%%%%%%%%%%%

TraceMatrix = zeros(NoOfSamples,NoOfTraces);
fseek(SegFile,3840,'bof');

%%%%%%%%%%%%%
%For number of traces
%%%%%%%%%%%%%%
```

```
for i = 1 : NoOfTraces

%%%%%%%%%%%%%%%
%For Number of Samples
%%%%%%%%%%%%%%%%

for j = 1 : NoOfSamples
TempMatrix = fread(SegFile,4);
SignValue = bitand(TempMatrix(1,1),128);
if SignValue == 128;
Sign = -1;
else
Sign = 1;
end
Char = bitand(TempMatrix(1,1),127);
Charac = 16^(Char-64);
Fraction= bitshift(TempMatrix(2,1),16) + bitshift(TempMatrix(3,1),8) +
TempMatrix(4,1);
SampleValue = Sign*Charac*Fraction;
TraceMatrix(j,i) = SampleValue;
end
fseek(SegFile,240,'cof');
end
TraceMatrix;
disp('Loading seg file, sucessfully completed.')
figure
surf(TraceMatrix)
title('Raw data')
xlabel('No of traces')
ylabel('No of samples')
zlabel('Amplitude')
```

```
%%%%%%%%%%%%%
% Closing the segy file
%%%%%%%%%%%%%%

fileclose = fclose(SegFile);
if fileclose < 0
error('Error while closing the file')
return;
end
disp('Filtering the data. PLease wait...')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%BUTTERWORTH(TENTH ORDER)FILTER%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%

[b,a] = butter(10,400/1000,'low')
for i=1 :NoOfTraces
TempTrace =conv(a,TraceMatrix(:,i));
FilteredMatrix(:,i)=TempTrace;
end

disp('.....Filtering complete.')
figure
surf(FilteredMatrix)
title('Filtered through BW 10th order ')
xlabel('No of traces')
ylabel('No of samples')
zlabel('Amplitude')
return

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% "Loading Seg-Y" routine and filtering the data with Chebyshev1 Filter
% "SegPath" is the full path to the segy file,
% the "SegPath" should also contain the file extension
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function loadsegy()

%%%%%%%%%%%%
% Open the segy file
%%%%%%%%%%%%

SegFile = fopen('D:MHD-101.sgy','r');
if SegFile < 0
error('Cannot open the specified file. Make sure that the path is correct and
contains the file extension')
return;
end
disp('Loading seg file, please wait. This may take a few minutes...')

%%%%%%%%%%%%%%%%%%%%%%%%
% Reading the number of traces parameter
%%%%%%%%%%%%%%%%%%%%%%%%

fseek(SegFile,3212,'bof');
TempMatrix = fread(SegFile,2);
NoOfTraces = bitshift(TempMatrix(1,1),8) + TempMatrix(2,1);
NoOfTraces = 100
if NoOfTraces <= 0
error ('Enter the Number of traces, at byte location 3213.')


end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%
% Reading the sampling interval parameter
%%%%%%%%%%%%%%%%%%%%%%%%%

fseek(SegFile,3216,'bof');
TempMatrix = fread(SegFile,2);
SamplingInterval = bitshift(TempMatrix(1,1),8) + TempMatrix(2,1)
if SamplingInterval <= 0
error ('Enter Sampling Interval, at byte location 3217.')
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Reading the  number of number of samples per trace parameter
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

fseek(SegFile,3220,'bof');
TempMatrix = fread(SegFile,2);
NoOfSamples = bitshift(TempMatrix(1,1),8) + TempMatrix(2,1)
if NoOfSamples <= 0
error ('Enter the Number of traces, at byte location 3221.')
end

%%%%%%%%%%%%%%%%%%%%%%%%%%
% Reading the Traces
% The Matrix holding the traces is "TraceMatrix"
%%%%%%%%%%%%%%%%%%%%%%%%%%%

TraceMatrix = zeros(NoOfSamples,NoOfTraces);
fseek(SegFile,3840,'bof');

%%%%%%%%%%%%%%
%For number of traces
%%%%%%%%%%%%%%

for i = 1 : NoOfTraces
```

```
%%%%%%%%%%%%%%
%For Number of Samples
%%%%%%%%%%%%%%

for j = 1 : NoOfSamples
TempMatrix = fread(SegFile,4);
SignValue = bitand(TempMatrix(1,1),128);
if SignValue == 128;
Sign = -1;
else
Sign = 1;
end
Char = bitand(TempMatrix(1,1),127);
Charac = 16^(Char-64);
Fraction= bitshift(TempMatrix(2,1),16) + bitshift(TempMatrix(3,1),8) + TempMatrix(4,1);
SampleValue = Sign*Charac*Fraction;
TraceMatrix(j,i) = SampleValue;
end
fseek(SegFile,240,'cof');
end
TraceMatrix;
disp('Loading seg file, sucessfully completed.')
figure
surf(TraceMatrix)
title('Raw')

%%%%%%%%%%%%%
% Closing the segy file
%%%%%%%%%%%%%%

fileclose = fclose(SegFile);
if fileclose < 0
error('Error while closing the file')
```

```
return;
end
disp('Filtering the data. PLease wait...')

%%%%%%%%%%%%%%%%%%%%%%
%%%%%%CHEBYSHEV1 FILTER%%%%%
%%%%%%%%%%%%%%%%%%%%%%]

[b,a] = cheby1(9,0.5,300/500);
for i=1 :NoOfTraces
TempTrace =conv(a,TraceMatrix(:,i));
FilteredMatrix(:,i)=TempTrace;
end

disp('.....Filtering complete.')
figure
surf(FilteredMatrix)
title('Filtered through Chebyshev 1')
xlabel('No of traces')
ylabel('No of samples')
zlabel('Amplitude')
return

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% "Loading Seg-Y" routine and filtering the data with CHEBYSHEV2 FILTER.
% "SegPath" is the full path to the segy file,
% the "SegPath" should also contain the file extension
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function loadsegy()

%%%%%%%%%%%%
% Open the segy file
%%%%%%%%%%%%

SegFile = fopen('D:MHD-101.sgy','r');
if SegFile < 0
error('Cannot open the specified file. Make sure that the path is correct and
contains the file extension')
return;
end
disp('Loading seg file, please wait. This may take a few minutes...')

%%%%%%%%%%%%%%%%%%%%%%%%
% Reading the number of traces parameter
%%%%%%%%%%%%%%%%%%%%%%%%

fseek(SegFile,3212,'bof');
TempMatrix = fread(SegFile,2);
NoOfTraces = bitshift(TempMatrix(1,1),8) + TempMatrix(2,1);
NoOfTraces = 100
if NoOfTraces <= 0
error ('Enter the Number of traces, at byte location 3213.')
```

```
end

%%%%%%%%%%%%%%%%%%%%%%%
% Reading the sampling interval parameter
%%%%%%%%%%%%%%%%%%%%%%%%

fseek(SegFile,3216,'bof');
TempMatrix = fread(SegFile,2);
SamplingInterval = bitshift(TempMatrix(1,1),8) + TempMatrix(2,1)
if SamplingInterval <= 0
error ('Enter Sampling Interval, at byte location 3217.')
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Reading the  number of number of samples per trace parameter
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

fseek(SegFile,3220,'bof');
TempMatrix = fread(SegFile,2);
NoOfSamples = bitshift(TempMatrix(1,1),8) + TempMatrix(2,1)
if NoOfSamples <= 0
error ('Enter the Number of traces, at byte location 3221.')
end

%%%%%%%%%%%%%%%%%%%%%%%%%%
% Reading the Traces
% The Matrix holding the traces is "TraceMatrix"
%%%%%%%%%%%%%%%%%%%%%%%%%%

TraceMatrix = zeros(NoOfSamples,NoOfTraces);
fseek(SegFile,3840,'bof');

%%%%%%%%%%%%
%For number of traces
%%%%%%%%%%%%
```

```
for i = 1 : NoOfTraces

%%%%%%%%%%%%%%%
%For Number of Samples
%%%%%%%%%%%%%%%%

for j = 1 : NoOfSamples
TempMatrix = fread(SegFile,4);
SignValue = bitand(TempMatrix(1,1),128);
if SignValue == 128;
Sign = -1;
else
Sign = 1;
end
Char = bitand(TempMatrix(1,1),127);
Charac = 16^(Char-64);
Fraction= bitshift(TempMatrix(2,1),16) + bitshift(TempMatrix(3,1),8) +
TempMatrix(4,1);
SampleValue = Sign*Charac*Fraction;
TraceMatrix(j,i) = SampleValue;
end
fseek(SegFile,240,'cof');
end
TraceMatrix;
disp('Loading seg file, sucessfully completed.')
figure
surf(TraceMatrix)
title('Raw')

%%%%%%%%%%%%%
% Closing the segy file
%%%%%%%%%%%%%%

fileclose = fclose(SegFile);
if fileclose < 0
error('Error while closing the file')
```

```
return;
end
disp('Filtering the data. PLease wait...')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%CHEBYSHEV2 FILTER%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

[b,a] = cheby2(9,20,300/500);
for i=1 :NoOfTraces
TempTrace =conv(a,TraceMatrix(:,i));
FilteredMatrix(:,i)=TempTrace;
end
disp('.....Filtering complete.')
figure
surf(FilteredMatrix)
title('Filtered through Chebyshev 2')
xlabel('No of traces')
ylabel('No of samples')
zlabel('Amplitude')
return

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

This is the rotated resultant view of the last view of raw data picture

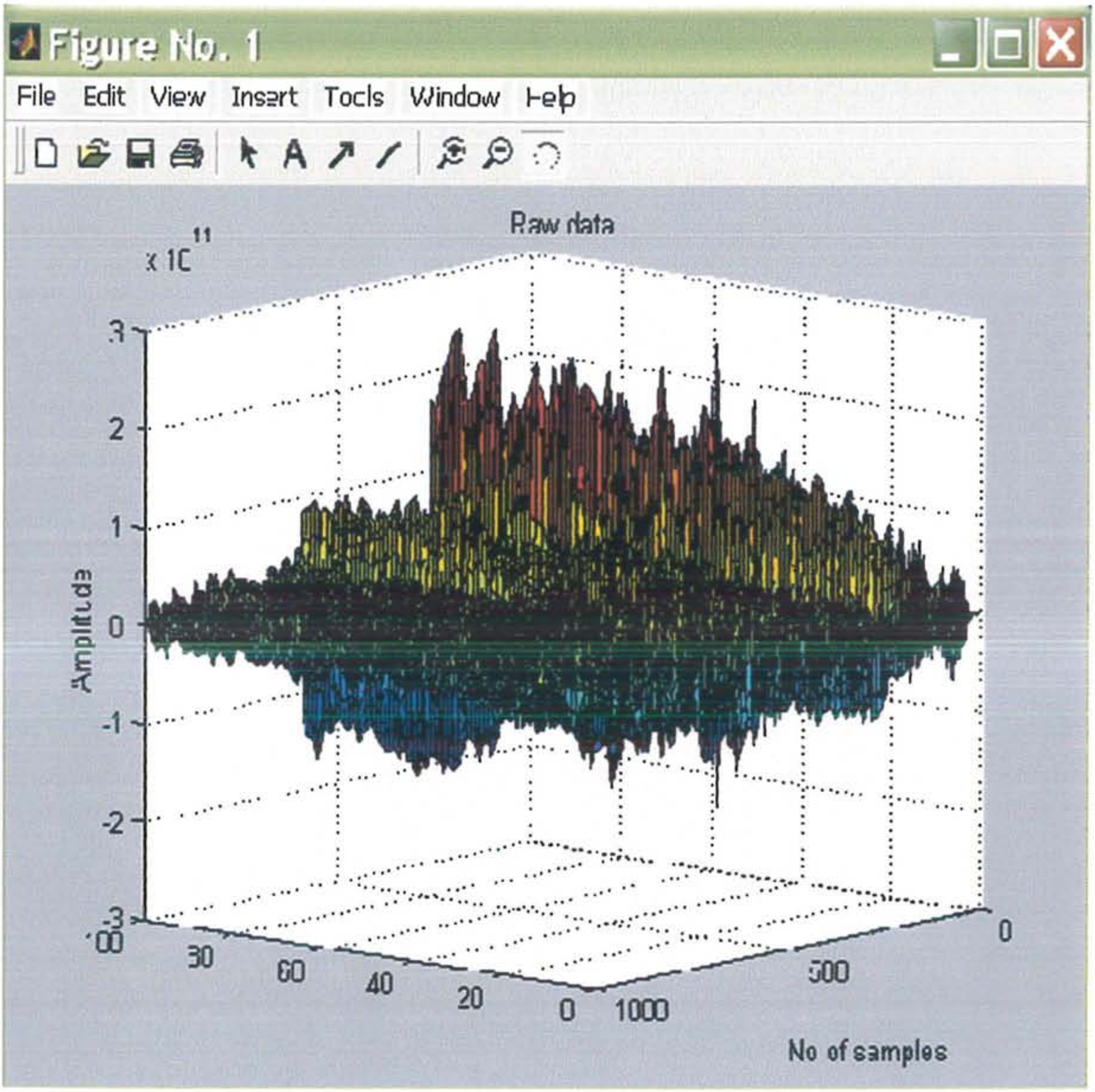


Figure shows the graphical plot of the data, filtered through the Butter worth first order filter with the cutoff frequency of 15/100.

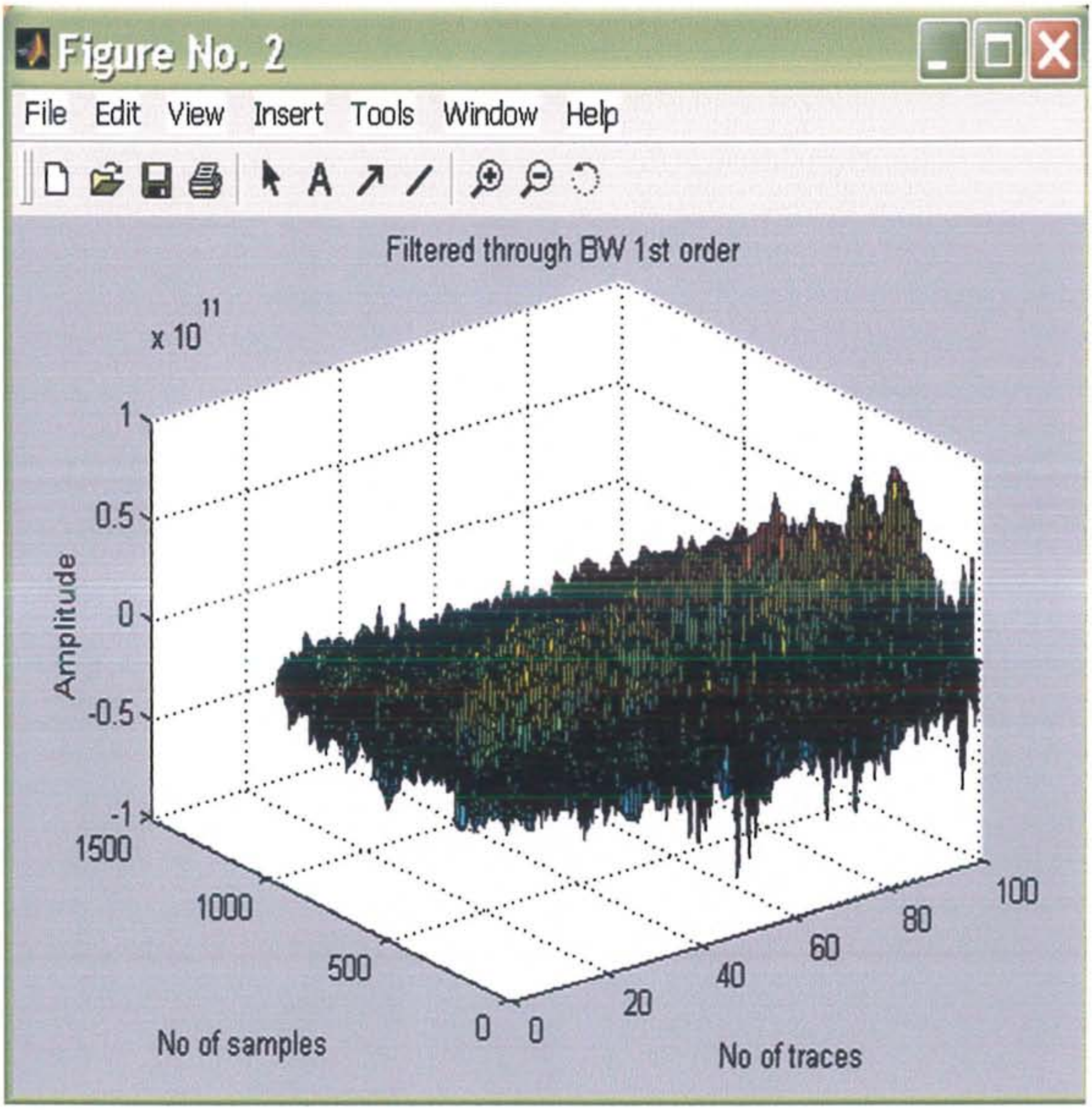


Figure shows the graphical plot of the data, filtered through the Butter worth second order filter with the cutoff frequency of 15/100.

Shows the different response with the change of the filter's order.

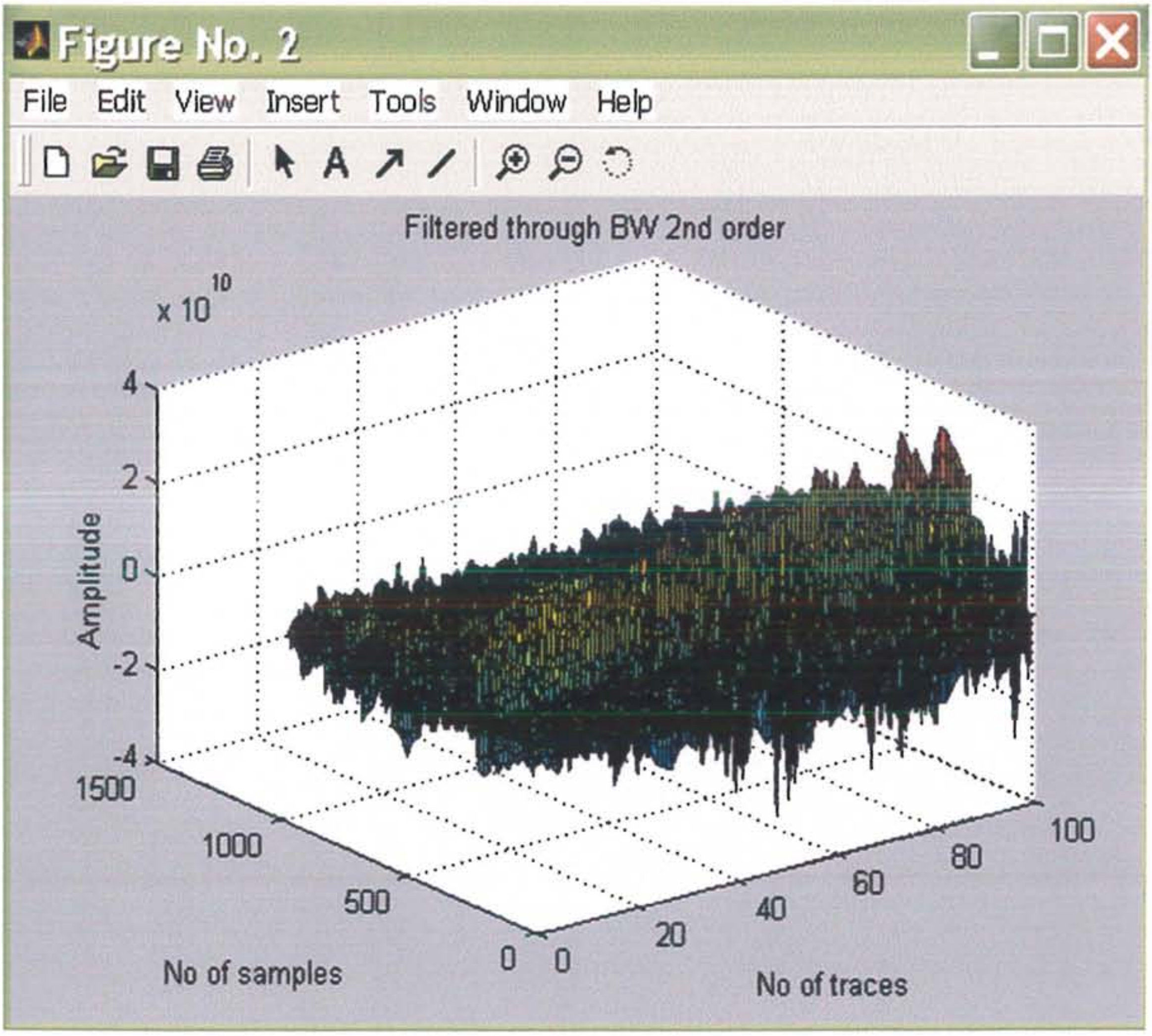
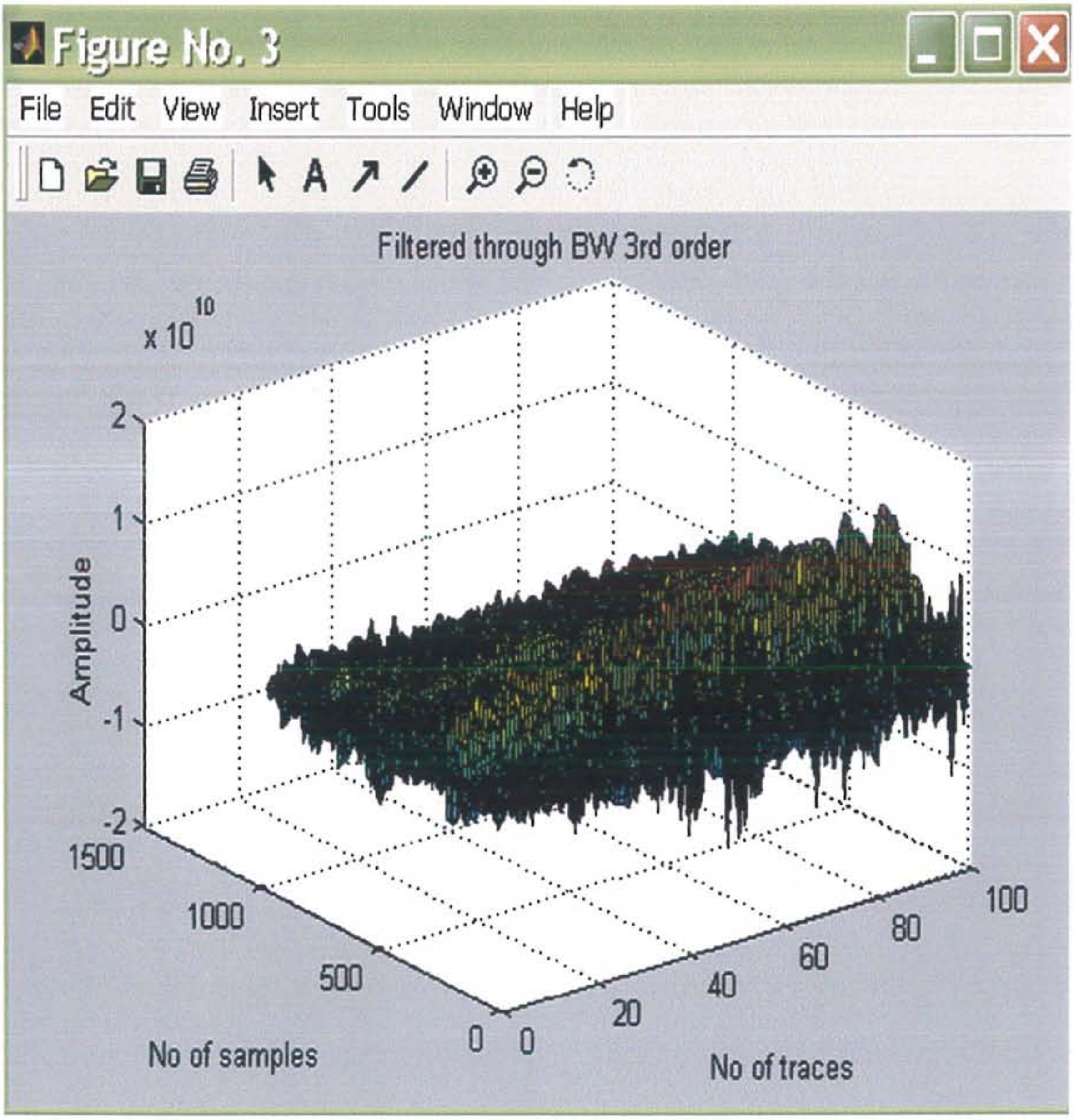
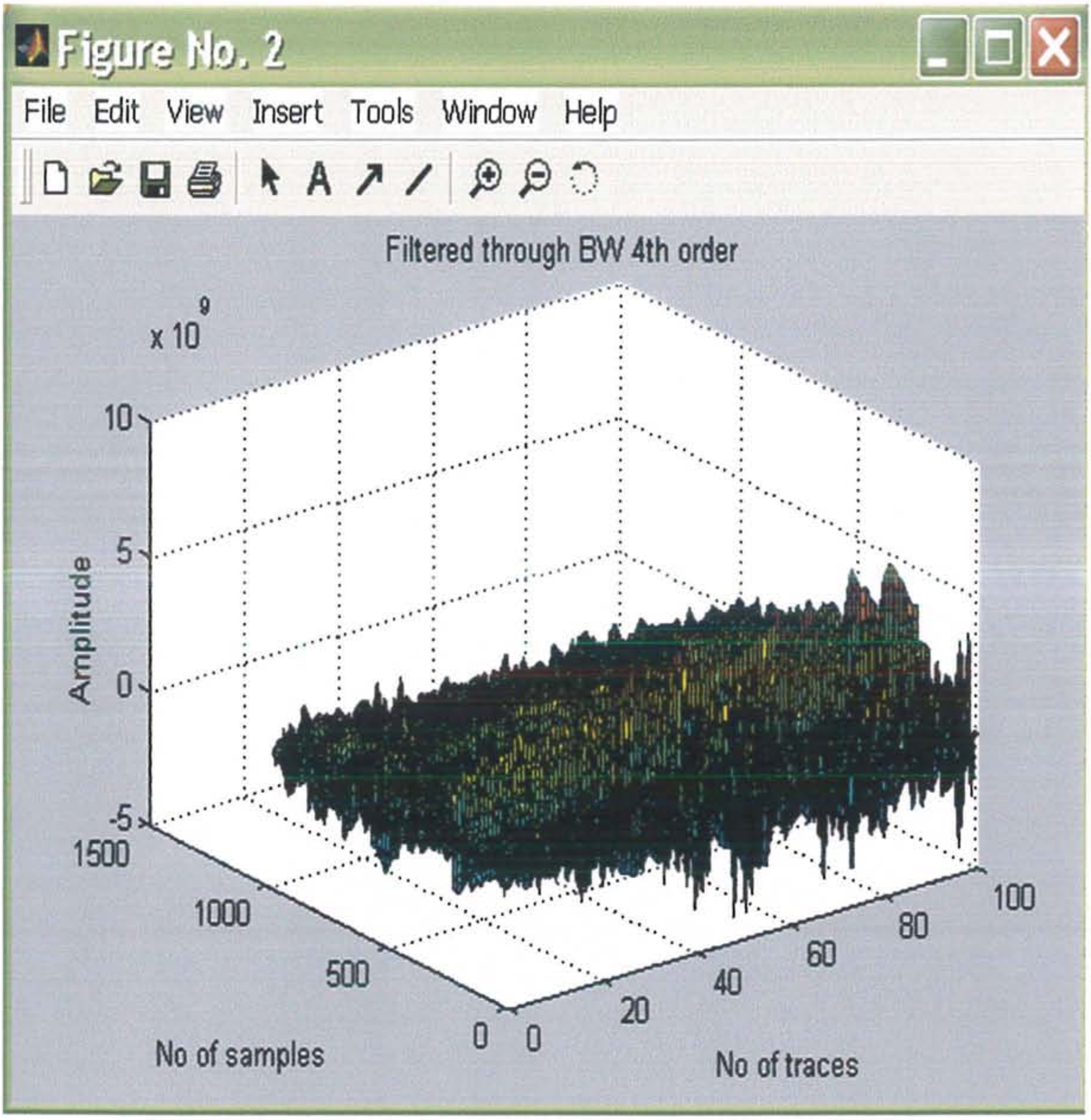


Figure shows the graphical plot of the data, filtered through the Butter worth third order filter with the cutoff frequency of 15/100.

Figure clearly Shows the decries of the amplitude with the change of the filter's order having the same cutoff frequency.



Graphical plot of the data filtered through the Butter worth fourth order filter.



Graphical plot of the data filtered through Butter worth fifth order filter, showing the different response with the change of order of the filter.

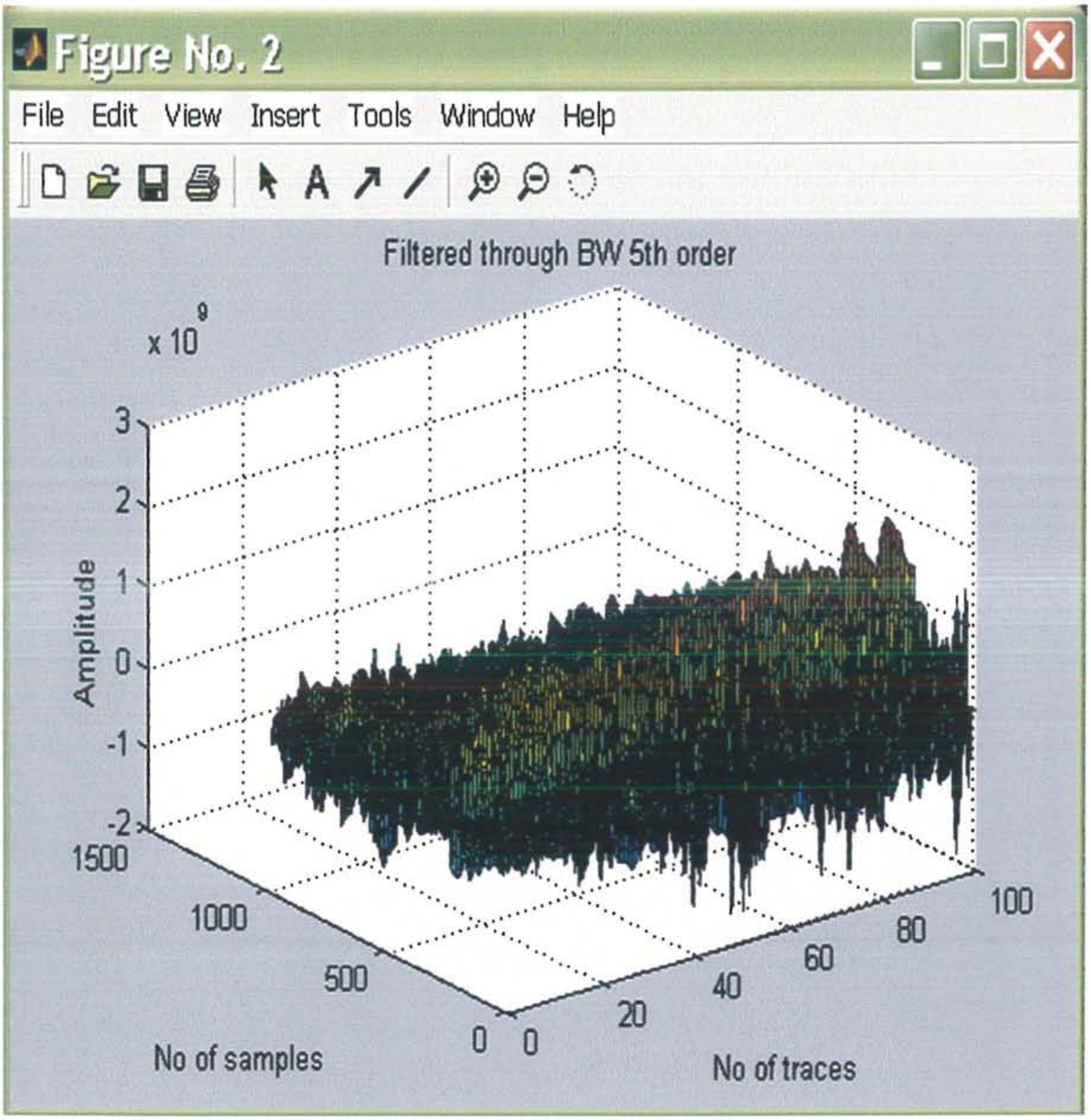


Figure shows the graphical plot of the data, filtered through the Butter worth ninth order filter with the cutoff frequency of 400/1000; it is the low pass filter.

Shows the different response with the change of the filter's order and cutoff frequency.

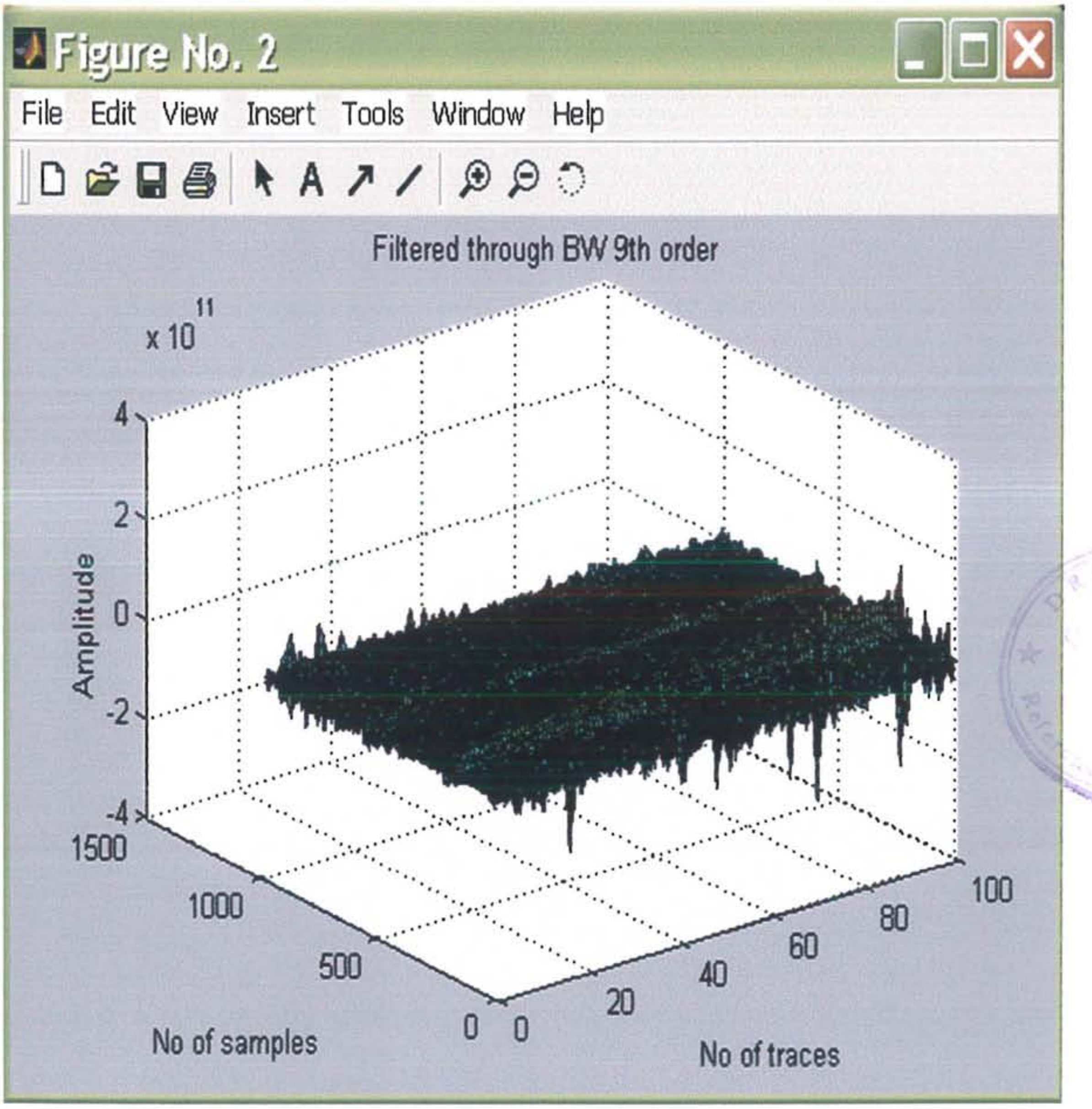


Figure shows the graphical plot of the data, filtered through the Butter worth tenth order filter with the cutoff frequency of 400/1000; it is the low pass filter.

Shows the different response from the previous plot with the change of the filter's order.

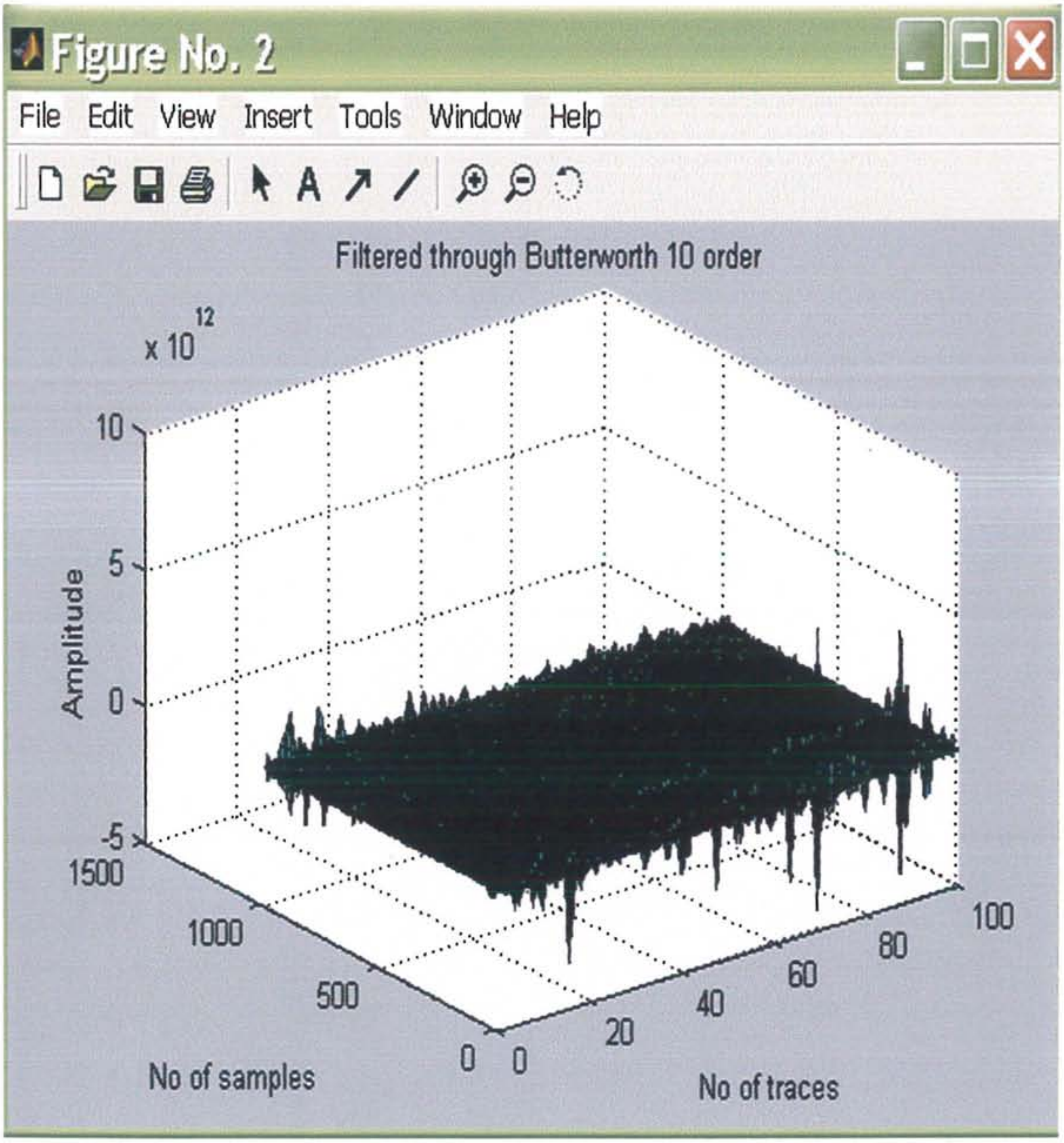
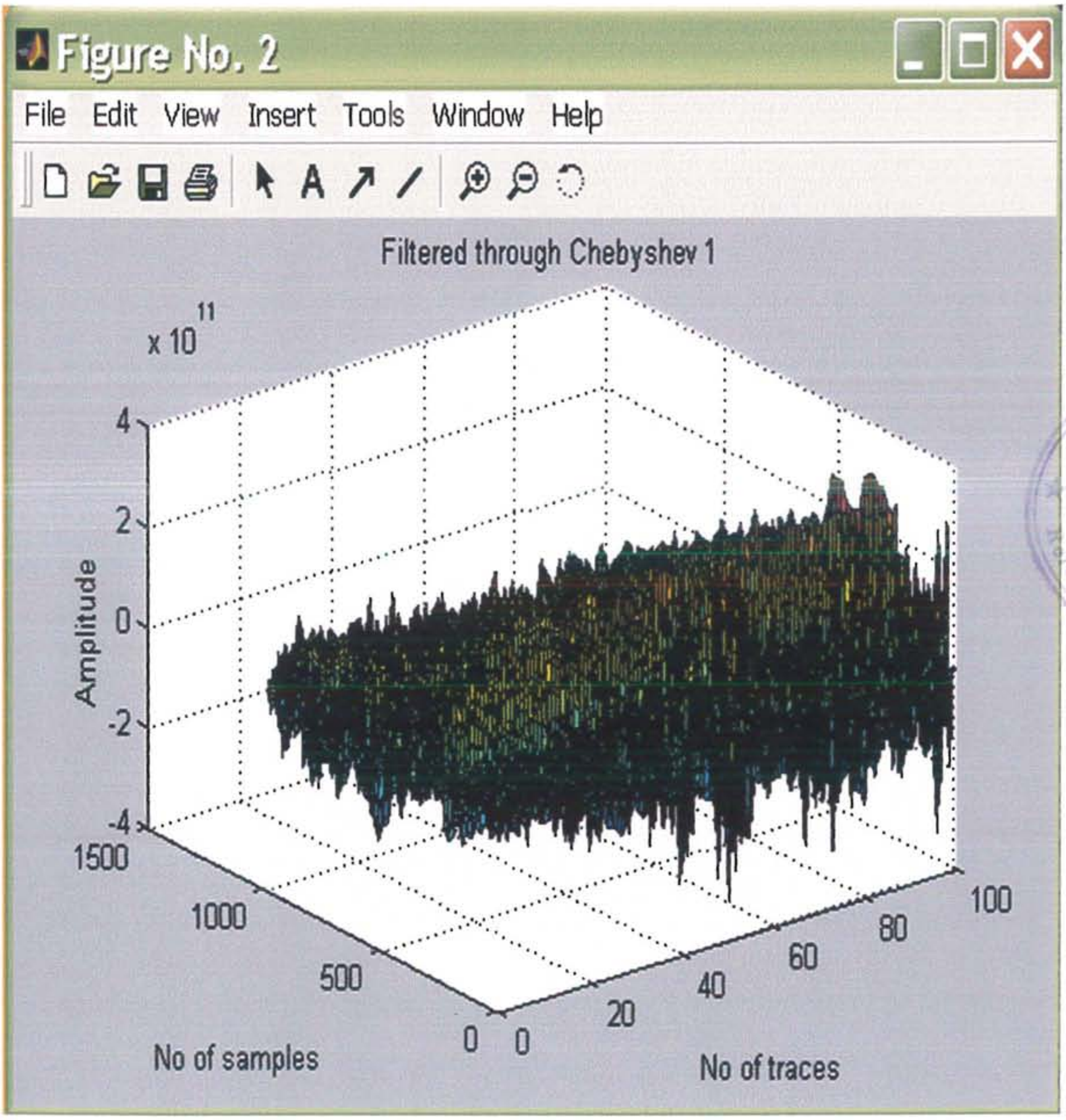
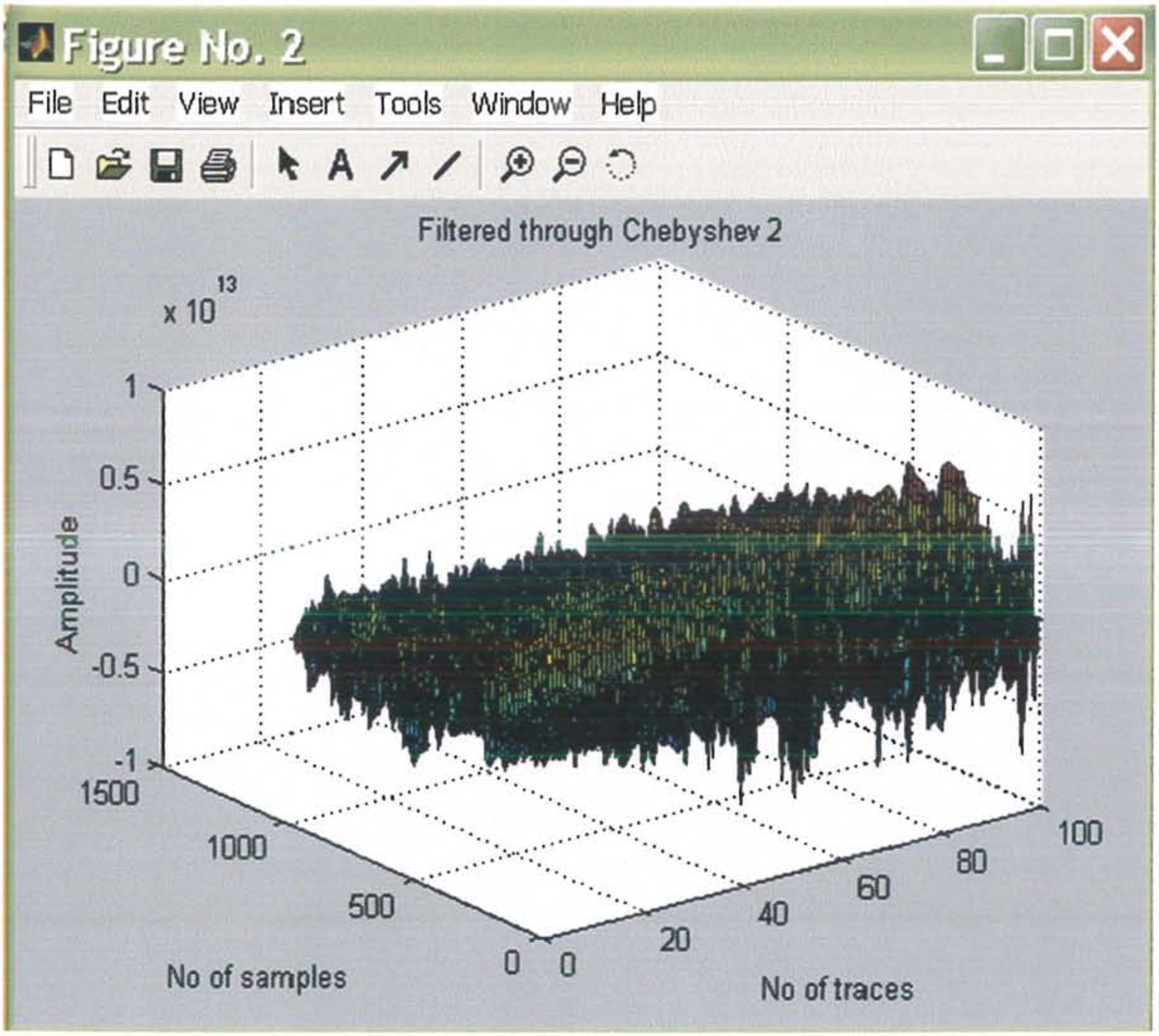


Figure shows the graphical plot of the data, filtered through the Chebyshev1 ninth order filter with the cutoff frequency of 300/500 and rp (ripple in bandpass) of 0.5; it is the low pass digital filter.



Graphical of the data filtered through Chebyshev 2 ninth order filter with the cutoff frequency of 300/500. it is also the low pass filter.



REFERENCES

- A Review To Digital Processing Of Geophysical Data By Roy O lindseth p.

- Basic Exploration Geophysics By Edwin S. Robinson, Virginia Polytechnic Institute And State University .

- Digital Filters Analysis, Design And Application (2ND EDITION) BY Andreas Antoniou, University Of Victoria.

- Geophysical Signal Analysis By Enders A. Robinson Sven Treitel

- Practical Seismic Data Processing BY Zia-ur-Rehaman

- Seismic Data processing BY Ozdogan Yalmaz

- Seismic Exploration By Hamid N. Al-Sadi (1980).

- Whole Earth Geophysics BY Robert J. Lillie, Oregon State University