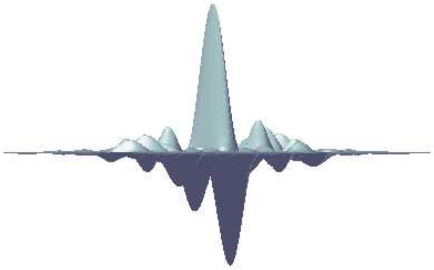


Statistical and Transform Methods in Geophysical Signal Processing

M. D. Sacchi

Department of Physics
University of Alberta

Contact:

Dr M.D.Sacchi
Department of Physics,
University of Alberta,
Edmonton, Canada, AB, T6G 2J1

Sacchi@phys.ualberta.ca
www-geo.phys.ualberta.ca/~sacchi/saig

Contents

Preface

This course will focus on the application on modern processing and inversion techniques to geophysical signal processing. We will also discuss the design and utilization of multi-dimensional linear transforms to suppress deterministic and stochastic noise from seismic records. The course is intended for upper level undergraduate and graduate students in geosciences as well as for geophysicists interested in understanding current technologies utilized in geophysical data processing.

About the author: M.D.Sacchi received a degree in Geophysics fron the National University of La Plata in 1988 and a PhD in Geophysics from the University of British Columbia, Vancouver, Canada, in 1996. Since 1997 he has been with the Department of Physics, University of Alberta in Edmonton, Canada. He is currently an associate professor of geophysics. His areas of interest are seismology, signal processing, imaging and inverse theory.

Chapter 1

Fourier Analysis

1.1 Introduction

In this part of the course we will review some fundamental aspects of Fourier Analysis. In particular, we will first study some aspects of orthogonal expansions. We will also study Fourier series, and the Fourier transform. Along this course we will deal with continuous and discrete signals. In this chapter, we explore the basic treatment of continuous signals. The extension to the discrete case is covered in Chapter 2.

1.1.1 Orthogonal Functions

We present the basic treatment to expand a function (in general a time dependent signal) in terms of a superposition of orthogonal functions.
A set of functions $\Phi_j(t),\ j = 1, 2, 3, \ldots$ is said to be orthogonal in the interval $[t_1, t_2]$ if the following condition is satisfied:

$$\int_{t_1}^{t_2} \phi_i(t)\phi_j(t)dt = k_i\delta_{i,j} \tag{1.1}$$

where $\delta_{i,j}$ is the Kronecker operator

$$\delta_{i,j} = 0 \qquad \text{if} \qquad i \neq j$$

$$\delta_{i,j} = 1 \qquad \text{if} \qquad i = j\,.$$

In signal processing, we usually want to represent a signal as a superposition of simple functions (sines, cosines, boxcar functions). The convenience of this procedure will become clear along the course (I hope!). In general, one can say that the representation should be in terms of functions with some attractive mathematical properties or with some physical meaning.
Let assume that we want to approximate a function $f(t)$ by a superposition of n orthogonal functions:

$$f(t) \approx \sum_{i=1}^{N} c_i \phi_i(t) \tag{1.2}$$

The coefficients $c_i \,, i = 1 \dots N$ can be obtained by minimizing the means square error defined as:

$$MSE = \frac{1}{t_2 - t_1} \int_{t_1}^{t_2} (f(t) - \sum_{i=1}^{N} \phi_i(t))^2 \, dt \tag{1.3}$$

the last equation can be expanded as follows:

$$MSE = \frac{1}{t_2 - t_1} \int_{t_1}^{t_2} (f(t)^2 + \sum_{i=1}^{N} c_i^2 \phi_i(t)^2 - 2 \sum_{i=1}^{N} c_i \phi_i(t) f(t)) \, dt \tag{1.4}$$

I have omitted the cross-products of the form $\phi_i(t)\phi_j(t)$ since according to the definition (1) they cancel up. The last equation can be written as

$$MSE = \frac{1}{t_2 - t_1} \int_{t_1}^{t_2} f(t)^2 \, dt + \sum_{i=1}^{N} c_i^2 k_i - 2 \sum_{i=1}^{N} c_i \gamma_i \tag{1.5}$$

where

$$\gamma_i = \int_{t_1}^{t_2} \phi_i(t) f(t) \, dt \,. \tag{1.6}$$

The term outside the integral in equation (1.5) can be rewritten as follows:

$$\sum_{i=1}^{N}(c_i^2 k_i - 2c_i\gamma_i) = \sum_{i=1}^{N}(c_i\sqrt{k_i} - \frac{\gamma_i}{\sqrt{k_i}})^2 - \sum_{i=1}^{N}\frac{\gamma_i^2}{k_i}\,.$$

We are now in condition of re-writing the MSE as follows:

$$MSE = \frac{1}{t_2 - t_1}\int_{t_1}^{t_2} f(t)^2\,dt + \sum_{i=1}^{N}(c_i\sqrt{k_i} - \frac{\gamma_i}{\sqrt{k_i}})^2 - \sum_{i=1}^{N}\frac{\gamma_i^2}{k_i} \qquad (1.7)$$

It is clear that the MSE is minimum when the second term in the RHS in the last equation is zero:

$$c_i\sqrt{(k_i)} = \frac{\gamma_i}{\sqrt{k_i}} \qquad (1.8)$$

or, in other words, the coefficient c_i is given by

$$c_i = \frac{\gamma_i}{k_i} = \frac{\int_{t_1}^{t_2} f(t)\phi_i(t)\,dt}{\int_{t_1}^{t_2}\phi(t)^2 dt},. \qquad (1.9)$$

We have obtained an expression for the N coefficients of the expansion of $f(t)$. If the $c_i\, i = 1\ldots N$ are chosen according to the last expression, the mean square error becomes:

$$MSE = \frac{1}{t_2 - t_1}\int_{t_1}^{t_2} f(t)^2\,dt - \sum_{i=1}^{N} c_i^2 k_i\,. \qquad (1.10)$$

It can be shown that if $N \to \infty$ the mean square error vanishes ($MSE \to \infty$). In that case, the last expression becomes what is called “Parseval Theorem”:

$$\int_{t_1}^{t_2} f(t)^2 dt = \sum_{i=1}^{\infty} c_i^2 k_i\,. \qquad (1.11)$$

1.1.2 Fourier Series

Consider the orthogonal set given by

$$e^{jn\omega_0 t}\,, \qquad n = 0, \pm 1, \pm 2, \pm 3, \ldots \tag{1.12}$$

this set is orthogonal in $t \in [t_0, t_0 + \frac{2\pi}{\omega_0}]$. To prove the last statement we need to evaluate the following integral [1]

$$\begin{array}{rcl} Int & = & \int_{t_0}^{t_0+2\pi/\omega_0} e^{jn\omega_0 t} e^{-jk\omega_0 t}\, dt \\ & = & \frac{1}{j\omega_0(n-k)} e^{j(n-k)} (e^{j2\pi(n-k)} - 1)\,, \end{array} \tag{1.13}$$

It is easy to see that the integral takes the following values:

$$Int = \left\{ \begin{array}{lll} 0 & \text{if} & n \neq k \\ 1 & \text{if} & n = k \end{array} \right. \tag{1.14}$$

We have proved that $e^{jn\omega_0 t}\,, \quad n = 0, \pm 1, \pm 2 \pm 3, \ldots$ conform an orthogonal set of functions.

When a signal is expanded in terms of exponential we have a Fourier series:

$$f(t) = \sum_{n=-\infty}^{\infty} F_n e^{jn\omega_0 t} \tag{1.15}$$

where the coefficients of the expansion are given by [2]

$$F_n = \frac{2\pi}{\omega_0} \int_{t_0}^{t_0+2\pi/\omega_0} f(t) e^{-jn\omega_0 t}\, dt \tag{1.16}$$

F_n is the complex spectrum of Fourier coefficients. The periodic signal $f(t)$ has been decomposed into a superposition of complex sinusoids of frequency $\omega_0 n$ and amplitude given by F_n. It is important to remember that a continuous and periodic signal has a discrete spectrum of frequencies given by:

[1] The inner product for complex functions is defined as $\int \phi_i(t)\phi_j(t)^*\, dt$, where $*$ stands for conjugate.

[2] we have already obtained this result for an arbitrary set $\phi_i(t)$

$$\omega_n = n\omega_0$$

To analyze non-periodic signals we need to introduce the Fourier Transform. In this case, the signal is represented in terms of a continuous spectrum of frequencies.

1.2 The Fourier Transform

So far we have found an expression that allow us to represent a periodic signal of period T in terms of a superposition of elementary functions (complex exponentials). We have seen that the Fourier series can be used to represent periodic or non-periodic signals. We have to realize, however, that the Fourier series does not properly represent a non-periodic signal outside the interval of $[t_0, t_0 + T]$. In fact, outside $[t_0, t_0 + T]$ the Fourier series provides a periodic extension of $f(t)$.

We have also shown that a periodic signal has a discrete spectrum given by the coefficients of the expansion in terms of the Fourier series, which we have called $F_n, n = 0, \pm 1, \pm 2, \ldots$.

In this section we provide a representation for a non-periodic signal $f(t)$ in $t \in (-\infty, \infty)$ by means of a continuous spectrum of frequencies.

Let us assume a periodic signal in the interval $[-T/2, T/2]$; the signal can be represented in terms of a Fourier series as follows:

$$f(t) = \sum_{n=-\infty}^{\infty} F_n e^{jn\omega_0 t} dt\,, \qquad \omega_0 = \frac{2\pi}{T} \tag{1.17}$$

where the coefficients are given by

$$F_n = \frac{1}{T} \int_{-T/2}^{T/2} f(t) e^{-jn\omega_0 t} dt \tag{1.18}$$

We can substitute equation (1.18) into (1.17) and obtain the following expression:

$$f(t) = \sum_{n=-\infty}^{\infty} \frac{1}{T} \int_{-T/2}^{T/2} f(t) e^{-jn\omega_0 t} dt e^{jn\omega_0 t} \,. \tag{1.19}$$

Now suppose that we make $T \to \infty$[3], we will also assume that the fundamental frequency $\omega_0 \to d\omega$, where $d\omega$ is a differential frequency. In this case, we can transform the discrete variable $n\omega_0$ into a continuous one ω, and finally, since now we have a summation on a continuous variable ω we will convert the summation into an integral

$$f(t) = \int_{-\infty}^{\infty} \frac{d\omega}{2\pi} (\int_{-\infty}^{\infty} f(t) e^{-j\omega t} dt) e^{j\omega t} d\omega \,. \tag{1.20}$$

The integral in brackets is called the Fourier transform of $f(t)$:

$$F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-j\omega t} dt \tag{1.21}$$

It is clear from equation (1.20) that the formula to represent the signal in terms of $F(\omega)$ is given by:

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) e^{j\omega t} d\omega \,. \tag{1.22}$$

The pair (1.21) and (1.22) are used to compute the Fourier transform its inverse, respectively. Equation (1.22) is also refereed as the inverse Fourier transform.[4]

It is important to stress that the signal in $(-\infty, \infty)$ has now a continuous spectrum of frequencies. The Fourier transform is in general a complex function that can be written as follows:

$$F(\omega) = |F(\omega)| e^{j\theta(\omega)} \tag{1.23}$$

[3] We want to extend out periodic signal into a non-periodic one

[4] In fact, one can think that equation (1.21) is a forward transform or a transform *to go* to a new domain (the frequency domain), whereas equation (1.22) is an inverse transform or a transform *to come back* to the original domain (time) from the frequency domain.

where $|F(\omega)|$ is the amplitude spectrum and $\theta(\omega)$ is the phase spectrum. We will come back to the importance of amplitude and phase when dealing with seismic signal.

1.2.1 Properties of the FT

We are not going to prove these properties, most of them can be proved by using the definition of the FT.
We shall use the following notation to indicate that $F(\omega)$ is the FT of $f(t)$:

$$f(t) \leftrightarrow F(\omega)$$

Symmetry.

$$F(t) \leftrightarrow 2\pi f(-\omega)$$

Linearity. If

$$f_1(t) \leftrightarrow F_1(\omega)$$

$$f_2(t) \leftrightarrow F_2(\omega)$$

then

$$f_1(t) + f_2(t) \leftrightarrow F_1(\omega) + F_2(\omega)$$

Scale. If

$$f(at) \leftrightarrow \frac{1}{|a|}F(\frac{\omega}{a})$$

Convolution. If

$$f_1(t) \leftrightarrow F_1(\omega)$$

$$f_2(t) \leftrightarrow F_2(\omega)$$

then

$$\int_{-\infty}^{\infty} f_1(u)f_2(t-u)du \leftrightarrow F_1(\omega)F_2(\omega)$$

or in a few words:*time convolution* ↔ *frequency multiplication*.[5]

[5] This is a very important property and we will make extensive use of it. Most physical systems can be described as linear and time invariant systems, this leads to a convolution integral.

Convolution in frequency. Similar to the previous one, but now:

$$f_1(t).f_2(t) \leftrightarrow \frac{1}{2\pi}\int_{-\infty}^{\infty} F_1(v)F_2(\omega - v)dv$$

or in a few words:*time multiplication* ↔ *frequency convolution.*[6]

Time delay. I like this one, we use it in seismic migration to extrapolate wavefield down into the earth.

$$f(t-\tau) \leftrightarrow F(\omega)e^{-j\omega t_0}$$

Modulation. This property makes you AM radio works.

$$f(t)e^{j\omega_0 t} \leftrightarrow F(\omega - \omega_0)$$

Time derivatives. This is used to compute derivatives (actually, using the discrete Fourier transform which we haven't seen)

$$\frac{df(t)}{dt} \leftrightarrow j\omega F(\omega)$$

It is clear that to take the derivative of $f(t)$ is equivalent to amplify the high frequencies.

The property can be extended for higher derivatives

$$\frac{d^n f(t)}{dt^n} \leftrightarrow (j\omega)^n F(\omega)$$

1.2.2 The FT of some signals

$\boxed{\text{A Boxcar}}$

We will compute the FT of the following function (a boxcar):

$$f(t) = \begin{cases} 1 & |t| < T/2 \\ 0 & otherwise \end{cases} \tag{1.24}$$

We substitute $f(t)$ into the definition of the FT (equation (1.21)) and solve the integral:

[6] We will use this property to estimate the FT of signal that has been recorded in a finite temporal window. See (1.2.3)

$$\begin{array}{rcl} F(\omega) & = & \int_{-T/2}^{T/2} 1.e^{-j\omega t} dt \\ & = & \frac{1}{-j\omega}(e^{-j\omega T/2} - e^{j\omega T/2}) \\ & = & T sinc(\omega T/2) \end{array} \qquad (1.25)$$

where in last equation $sinc(x) = sin(x)/x$. The FT of the boxcar function is a *sinc* function. We will come latter to the importance of the knowing the FT of the box car function when dealing with the spectrum of signal that have been truncated in time.

In Figures (4.102) and (1.2.2), I have displayed the Fourier transform of two boxcar functions of width $T = 10$ and 20 secs, respectively.

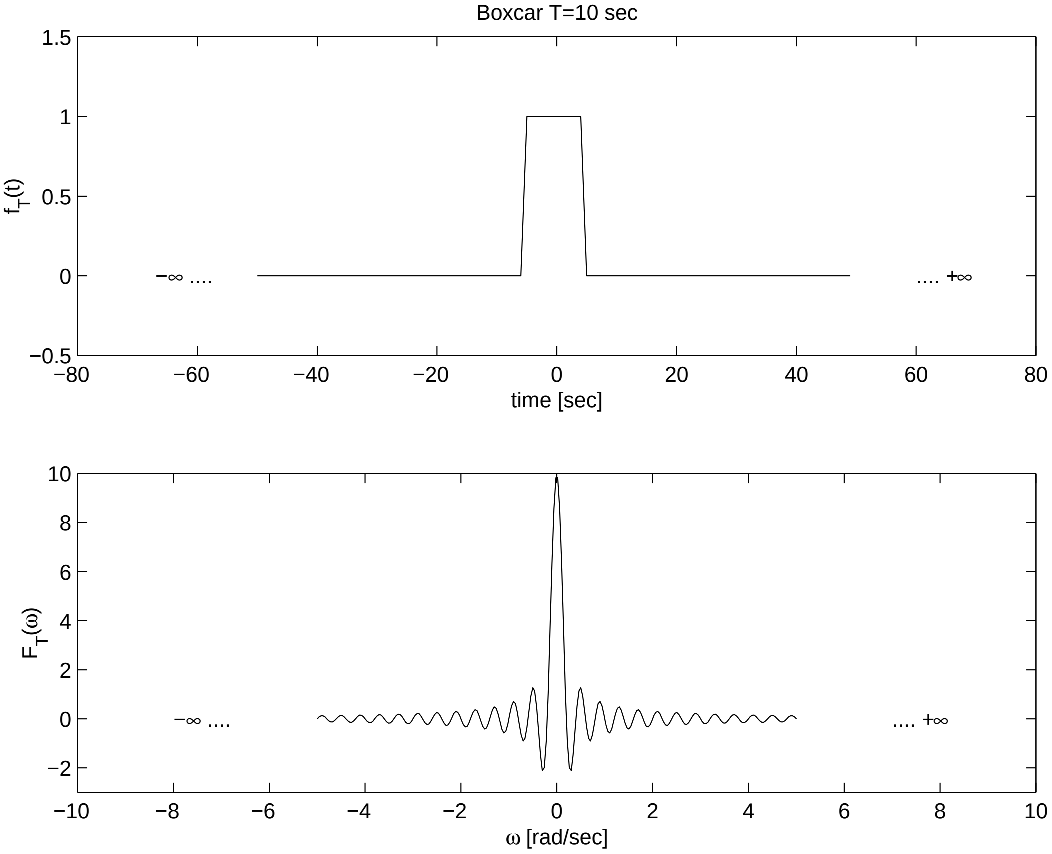


Figure 1.1: The Fourier transform of a boxcar function of $T = 10\,\mathrm{secs}$

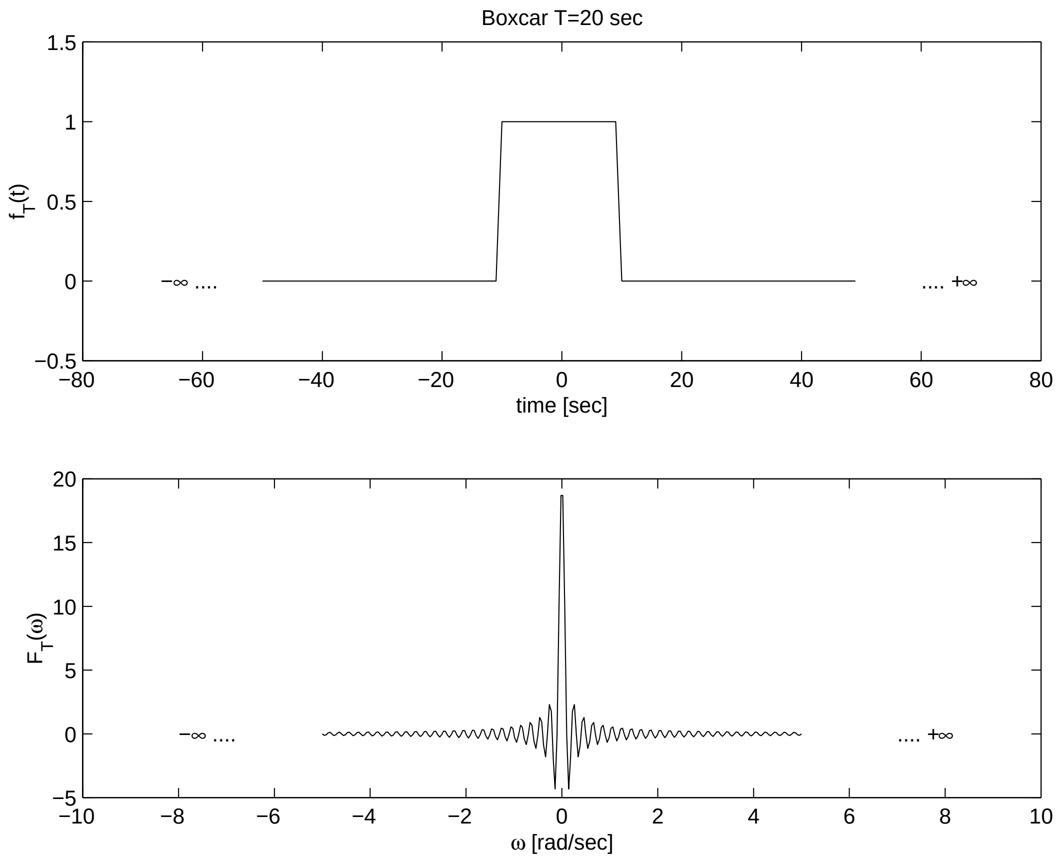


Figure 1.2: The Fourier transform of a boxcar function of $T = 20\,\mathrm{secs}$

Delta function:

$$f(t) = \delta(t)$$

the δ function is defined according to

$$\int g(u)\delta(u)du = g(0)$$

It easy to see from the above definition that the FT of the delta function is

$$F(\omega) = \int_{-\infty}^{\infty} \delta(t)e^{-j\omega t}dt = 1$$

$$\delta(t) \leftrightarrow 1$$

Similarly, if we apply the "time delay" property
we have

$$\delta(t-\tau) \leftrightarrow 1.e^{-j\omega\tau}$$

It is clear that the δ function has a continuous amplitude spectrum with all the frequencies. This is also the ideal seismic wavelet that one would like to have in seismic exploration.

A complex sinusoid:

We can combine the FT of the delta function with the symmetry property to obtain the FT of a complex sinusoid:
We have seen that

$$\delta(t-\tau) \leftrightarrow 1.e^{-j\omega\tau}$$

If we apply the symmetry property

$$F(t) \leftrightarrow 2\pi f(-\omega)$$

we end up with

$$e^{j\omega_0 t} \leftrightarrow 2\pi\delta(\omega - \omega_0)$$

In other words, the FT of complex sinusoid of frequency ω_0 is a delta at the corresponding frequency $\omega = \omega_0$.

1.2.3 Truncation in time

Given $f(t)$ $t \in (-\infty, \infty)$, with $f(t) \leftrightarrow F(\omega)$, how do we obtain the FT of the signal when the signal is recorded in a finite interval $[-T/2, T/2]$?.
We can call $f_T(t)$ the observed signal in $[-T/2, T/2]$ and $f(t)$ the original signal in $(-\infty, \infty)$, in this case is easy to see that

$$f_T(t) = f(t).b_T(t) \tag{1.26}$$

where $b_T(t)$ is a box function like the one analyzed in (1.2.2).
Using the frequency convolution theorem (1.2.1) we can write

$$F_T(\omega) = \frac{1}{2\pi}\int_{-\infty}^{\infty} F(v)B_T(\omega - v)dv = \frac{1}{2\pi}F(\omega) * B_T(\omega) \tag{1.27}$$

where $B_T(\omega) = T sinc(\omega T/2)$. This is remarkably interesting result (it is?). We are saying that our observation window is affecting the FT of the signal. We want to know $F(\omega)$ but since we are recording the signal in a finite interval, we have only access to $F_T(\omega)$. The latter is a distorted version of $F(\omega)$.

$$F_T(\omega) = \frac{1}{2\pi}T\int_{-\infty}^{\infty} F(u) sinc((\omega - u)T/2)du\,. \tag{1.28}$$

It is clear from the above that one does not see $F(\omega)$ but its convolution with a sinc function.
If $f(t) = e^{j\omega_0 t}$ it is easy to see that the truncated version of the complex sinusoid has the following FT:

$$F_T(\omega) = \frac{1}{2\pi}T\int_{-\infty}^{\infty} 2\pi\delta(\omega - \omega_0) sinc((\omega - u)T/2)du\,. \tag{1.29}$$

$$F_T(\omega) = T sinc((\omega - \omega_0)T/2)\,. \tag{1.30}$$

This is a sinc function with a peak at $\omega = \omega_0$.
In Figure (1.2.3) we portray the superposition of 2 complex sinusoids of the form

$$f(t) = e^{j\omega_1 t} + e^{j\omega_2 t}\,, \qquad t \in [-10, 10]\,\mathrm{secs}\,.$$

The FT of such a signal (if measured in an infinity interval) is given by two delta functions at frequencies ω_1 and ω_2. But since we are observing the signal in a finite length interval we have to convolve the *ideal* FT of $f(t)$ with the FT of the boxcar function. In this example I have chosen the following frequencies $\omega_1 = 0.5\,\mathrm{rad/sec}$ and $\omega_2 = 1.\,\mathrm{rad/sec}$.

1.3 Symmetries

Before continuing with the Fourier transform and its applications a few words about the symmetries of the FT are needed. This is very important in the discrete case at the time of writing computer codes to process geophysical data.
Let us start with the definition of the Fourier transform,

$$F(\omega) = \int f(t) e^{-i\omega t} dt \tag{1.31}$$

If the signal $f(t)$ is a real signal, we can write:

$$F(\omega) = R(\omega) + iG(\omega) \tag{1.32}$$

where

$$R(\omega) = \int f(t) cos(\omega t) dt \tag{1.33}$$

and

$$G(\omega) = -\int f(t) sin(\omega t) dt \tag{1.34}$$

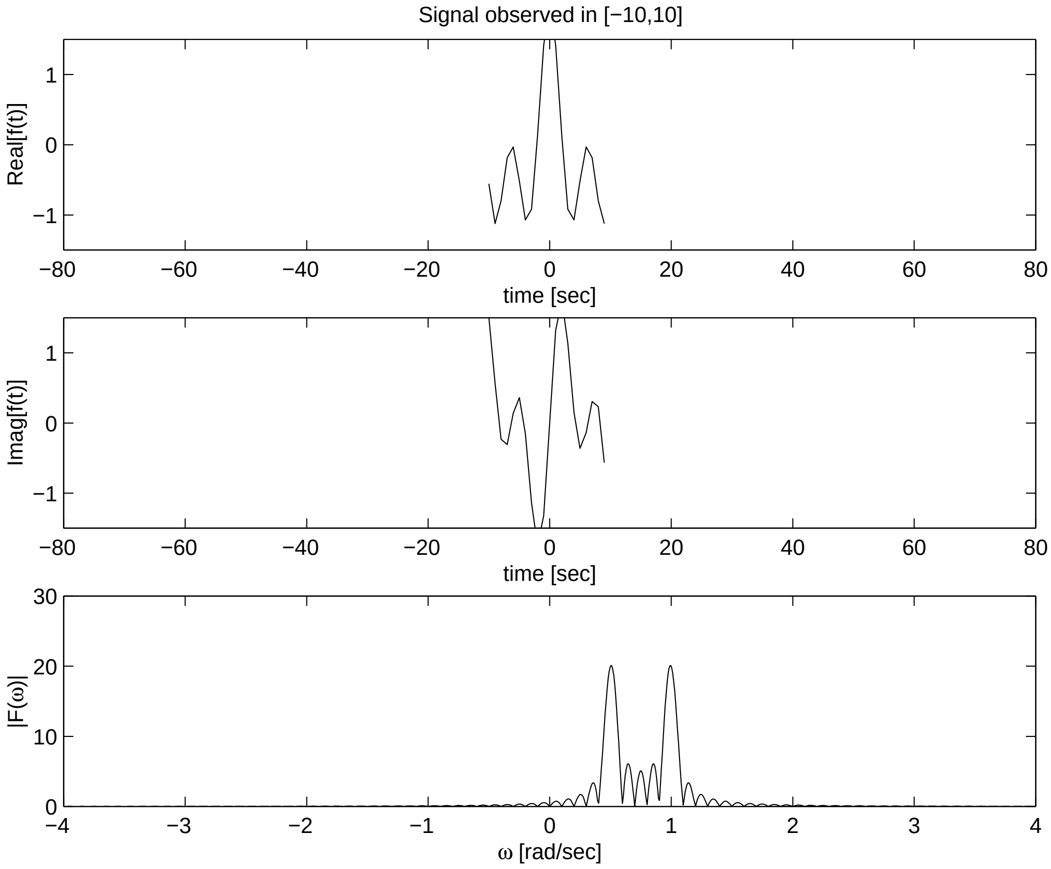


Figure 1.3: The Fourier transform of a the superposition of two complex sinusoids observed in a window of length $T = 20$ secs. Up: real part of the signal. Center: Imaginary part of the signal. Bottom: Amplitude of the Fourier Transform ($|F(\omega)|$).

Since the *cos* is an even function and the *sin* an odd function:

$$R(\omega) = R(-\omega) \tag{1.35}$$

$$G(\omega) = -G(-\omega) \tag{1.36}$$

If you know $F(\omega)$ for $\omega \geq 0$, you can compute the $F(\omega)$ for $\omega < 0$ by applying the above identities.

In fact, we can always write:

$$F(\omega) = R(\omega) + iG(\omega) \tag{1.37}$$

$$F(-\omega) = R(-\omega) + iG(-\omega) \tag{1.38}$$

by combining last equation with equation (1.36) we obtain

$$F(-\omega) = R(\omega) - iG(\omega)\,. \tag{1.39}$$

The last equation can be used to compute the negative semi-axis of the Fourier transform. This property is often referred as the Hermitian symmetry of the FT. You can also write:

$$F(-\omega) = F(\omega)^*$$

where the $*$ is used to denote complex conjugate. This property is only valid for real time series. This is why, we often plot one semi-axis (in general the positive one) when displaying the Fourier Spectrum of a real signal.

The symmetry properties of the real and imaginary parts of the Fourier transform can also be used to obtain the symmetries properties of the amplitude and phase of the Fourier transform:

$$F(\omega) = |F(\omega)|e^{i\theta(\omega)}\,.$$

It is east to prove that the amplitude is an even function:

$$|F(\omega)| = |F(-\omega)| \tag{1.40}$$

and that the phase is an odd function

$$\theta(\omega) = -\theta(-\omega)\,. \tag{1.41}$$

1.4 Living in a discrete World

So far we have described the FT of continuous (analog) signals. Now, we will start to study discrete signals or time series. This is the connection between the continuous and the discrete world. When working with real data we will use discrete signals. In Chapter 2, we will analyze discrete signals using the discrete Fourier transform and the Z transform.

We will designate $f(t)$ the analog signal and $f_s(t)$ the associated discrete signal. One can think that f_s is obtained by sampling $f(t)$ every Δt seconds

$$f_s(t) = f(t) \sum_{k=-\infty}^{\infty} \delta(t - k\Delta t)\,. \tag{1.42}$$

By the frequency convolution property we can obtain the FT of the sampled signal:

$$F_s(\omega) = \frac{1}{2\pi} F(\omega) * \omega_0 \sum_{k=-\infty}^{\infty} \delta(\omega - k\omega_0)\,, \qquad \omega_0 = \frac{2\pi}{\Delta T} \tag{1.43}$$

where in last equation I have assumed that we know how to compute the FT of the sampling operator $\sum_{k=-\infty}^{\infty} \delta(t - k\Delta t)$.

After a few mathematical manipulations, it is easy to see that

$$F_s(\omega) = \frac{1}{T} \sum_{k=-\infty}^{\infty} F(\omega - n\omega_0) \tag{1.44}$$

One can observe that the FT of the sampled signal is a periodic function with period ω_0.

If one wants to compute $F_s(\omega)$ in such a way that $F(\omega)$ can be completely recovered, the signal $f(t)$ must be a band-limited signal. This is a signal where the spectral components outside the interval $[-\omega_{max}, \omega_{max}]$ are zero. If the following condition is satisfied

$$\omega_0 \geq 2\omega_{max}$$

there is no overlap of spectral contributions, and therefore $F_s(\omega)$, $\omega \in [-w_{wmax}, w_m ax]$ is equivalent, whithin a scale factor $1/T$, to the FT of the analog signal $F(\omega)$. The last contidion can be re-written as follows:

$$\frac{2\pi}{\Delta T} \geq 2 \times 2\pi f_{max}$$

which reduces to

$$\Delta T \leq \frac{1}{2 f_{max}} \, .$$

The last equation is also designated as the sampling or Nyquist theorem. It basically tells us that to recover the FT of the original signal we need to sample the data according to the last inequalitiy.

Real world signals are continuous, and become discrete after going trough acquisition systems (i.e., digital seismograph). To avoid alias, analog filter are usually placed in the acquisition system. The data are first band-limited using analog filters, then sampled and finally, stored digitally.

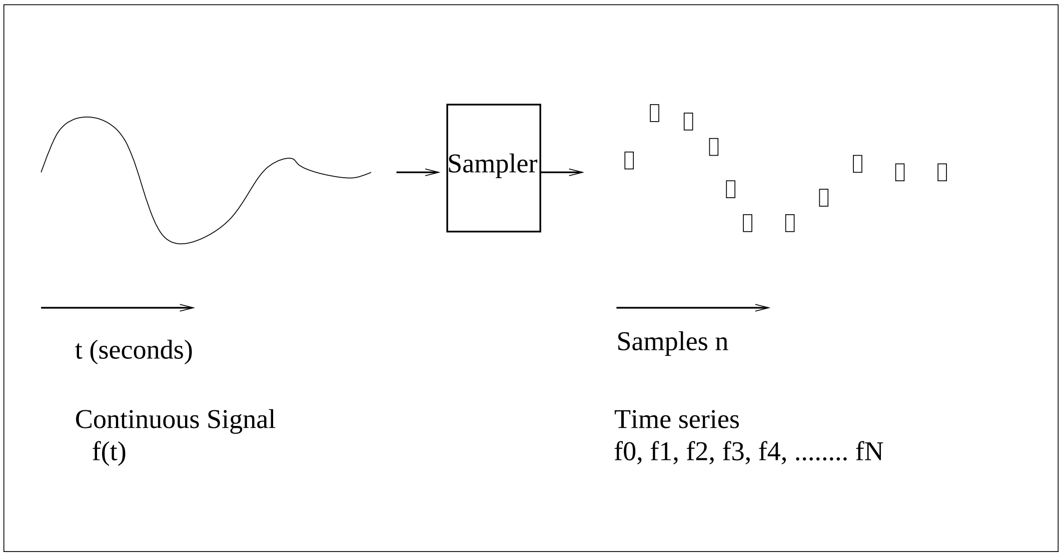


Figure 1.4: Discretization of a continuous signal.

The aliasing effect is described in Figures (1.5)-(1.8). Figure (1.5) corresponds to the Fourier transform of a continuous signal. we can observed that to properly recover the Fourier transform of the continuous signal we need to sample our data according to $w_0 \leq 2w_{max}$. This is true for Figures (1.6) and (1.7). In these two figures it easy to see that the Fourier transform of the original (continuous) signal is well represented by the Fourier transform of the discretized signal in the interval $[-\omega_{max}, \omega_{max}]$. In Figure (1.8) we portray an example where the data has been under-sampled and, therefore, the Fourier transform of the continuous signal cannot be recovered from the Fourier transform of the discretized signal.

1.5 References

Papoulis A., Fourier Integral and Its Applications, McGraw-Hill

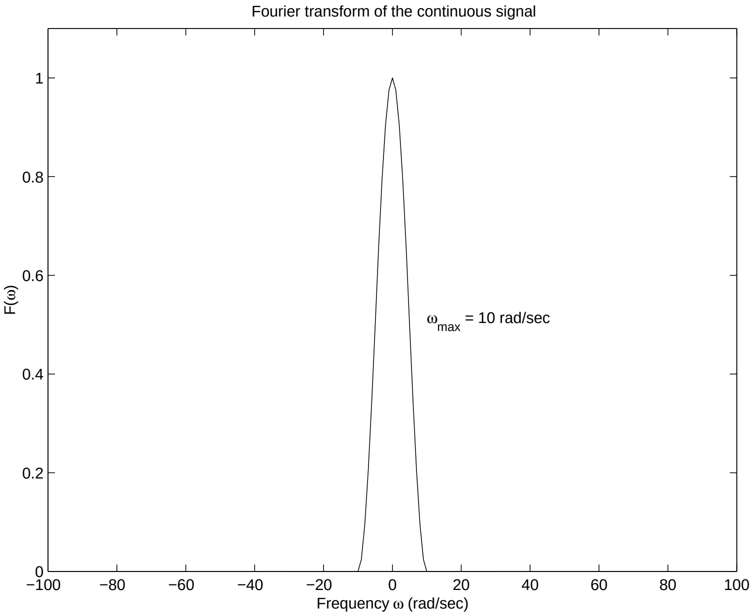


Figure 1.5: The Fourier transform of a continuous signal.

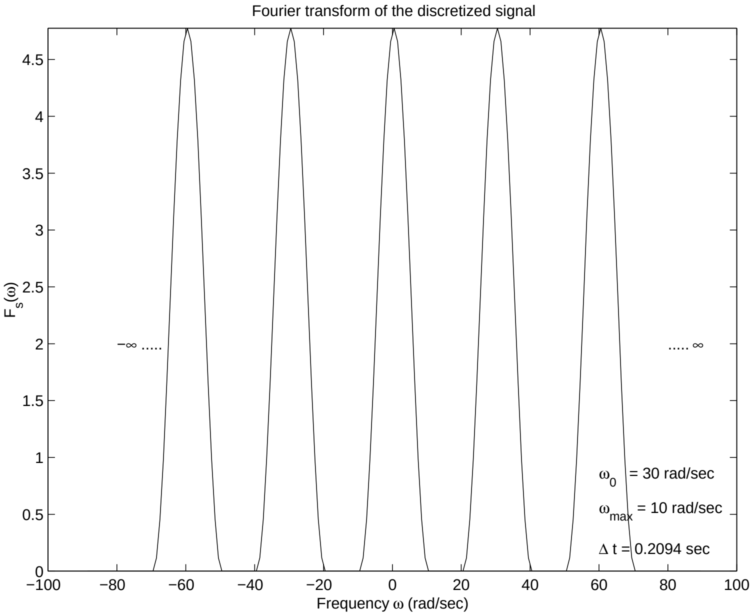


Figure 1.6: The Fourier transform the continuous signal after being discretized, in this case $\omega_{max} = 10$ and $\omega_0 = 30$

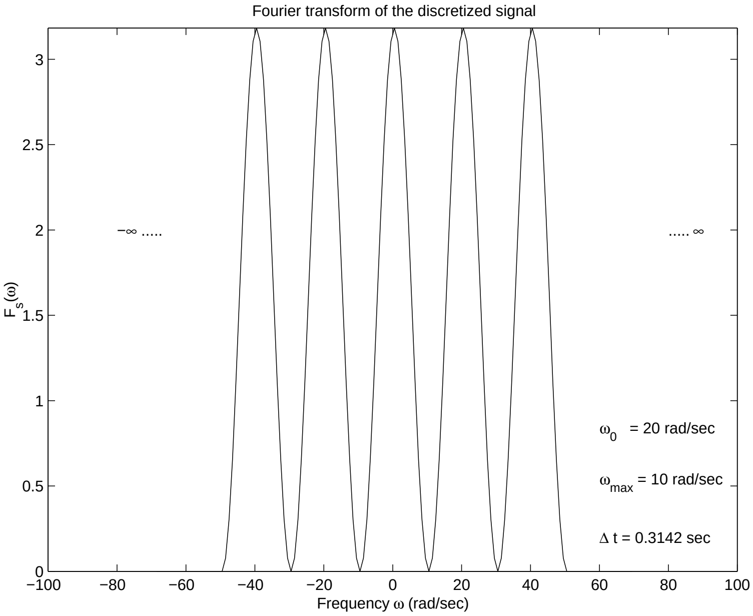


Figure 1.7: The Fourier transform the continuous signal after being discretized, in this case $\omega_{max} = 10$ and $\omega_0 = 20$. The Fourier transform of the continuous signal is perfectly represented in the interval $[-w_{max}, w_{max}]$.

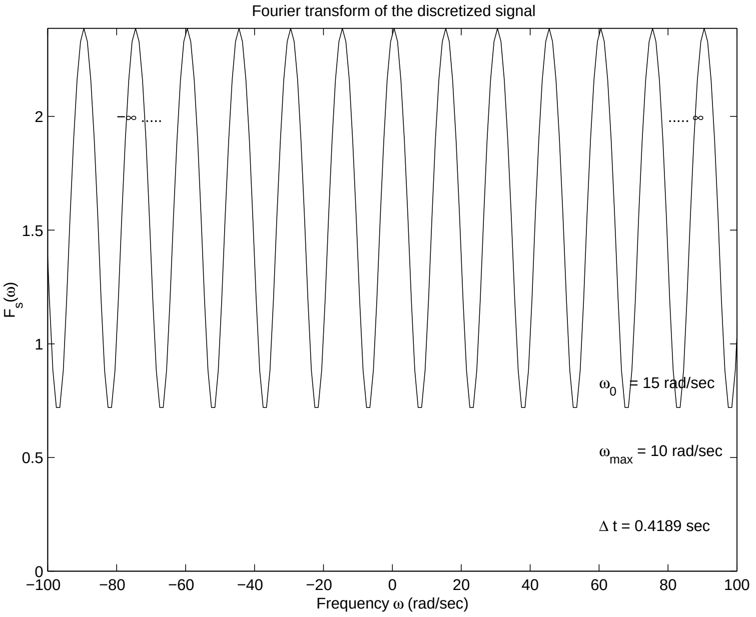


Figure 1.8: The Fourier transform the continuous signal after being discretized, in this case $\omega_{max} = 10$ and $\omega_0 = 15$. The signal is aliased. Note that Nyquist theorem is not satisfied. The Fourier transform of the continuous signal cannot be recovered from the Fourier transform of the sampled signal.

Chapter 2

Z-transform and Convolution

In this chapter we will introduce a new concept that is very useful at the time of dealing with discrete signals and linear systems. The Z transform permits one to do what the Fourier transform to continuous signals. Later we will also find out that the Z transform is related to the discrete Fourier Transform, this is discrete cousin of the Fourier transform studied in Chapter 1.

2.1 Linear Systems

Linear systems are useful to define input/output relationships for continuous and discrete signals. Assume that we have a linear system where the input to the system is a continuous signal $x(t)$ and the output is given by $y(t)$

$$x(t) \to y(t)$$

If the system is linear the following properties must be satisfied:

P1 :

$$\alpha x(t) \to \alpha y(t)\,.$$

P2: If

$$x_1(t) \to y_1(t)$$

and

$$x_2(t) \rightarrow y_2(t)$$

then

$$x_1(t) + x_2(t) \rightarrow y_1(t) + y_2(t)$$

.

P3: Properties **P1** and **P2** can be combined in a single property:

$$\alpha x_1(t) + \beta x_2(t) \rightarrow \alpha y_1(t) + \beta y_2(t)\,.$$

We will say that the linear system is time invariant if and only if:

$$x(t-T) \rightarrow y(t-T)$$

and this is true for any arbitrary T. In other words, if the input signal is delayed by an amount T, the output signal is delayed by the same amount. We will represent our linear system as follows:

$$\mathcal{H}[x(t)] = y(t)\,. \tag{2.1}$$

If the system is linear, the function $\mathcal{H}$ has the following expression:

$$y(t) = \mathcal{H}[x(t)] = \int_{-\infty}^{\infty} h(t,\tau)x(\tau)d\tau\,. \tag{2.2}$$

It is easy to prove that the above expression defines a linear system. When the system is linear and time invariant the following property should also be satisfied:

$$y(t-T) = \mathcal{H}[(x(t-T)]\,. \tag{2.3}$$

In this case we need to rewrite equation (2.2) in order to satisfy the aforementioned requirement. In this case the Green function of the system $h(t,\tau)$ is given by:

$$h(t,\tau) = h(t-\tau)\,. \tag{2.4}$$

If we replace $h(t-\tau)$ in equation (2.2) we end up with the following expression:

$$y(t) = \int_{-\infty}^{\infty} h(t-\tau)x(\tau)d\tau \tag{2.5}$$

It is clear that the above equation defines a linear system, but it is not clear that the system is time invariant. To prove that (2.5) corresponds to the i/o relationship of a time invariant linear system we will apply the following change of variables:

$$u = t - \tau\,.$$

Then,

$$y(t) = -\int_{\infty}^{-\infty} h(u)x(t-u)du = \int_{-\infty}^{\infty} h(u)x(t-u)du = H[x(t)]\,, \tag{2.6}$$

substituting t by $= t - T$

$$y(t-T) = \int_{\infty}^{\infty} h(u)x(t-T-u)du = H[x(t-T)] \tag{2.7}$$

we have proved that the convolution integral given in equation (2.5) defines a time invariant linear system. Using to the convolution theorem, "*convolution in the time domain* $\longrightarrow$ *multiplication in the frequency domain*", we can rewrite the convolution integral as follows:

$$Y(\omega) = H(\omega)\,.X(\omega)\,.$$

The function $h(t)$ is also called the impulse response of the system. The Fourier transform of the impulse response, $H(\omega)$, is the transfer function of the system. If the input to a system is given by $x(t) = \delta(t)$ the output is given by $y(t) = h(t)$. This statement can be easily proved by substituting $x(t) = \delta(t)$ into the convolution integral:

$$y(t) = \int_{-\infty}^{\infty} h(u)\delta(t-u)du = h(t)\,, \tag{2.8}$$

It turns out that if you do not know $h(t)$, it can be obtained by exciting the system with a δ function and measuring the output signal $y(t) = h(t)$ (Figure 2.1).

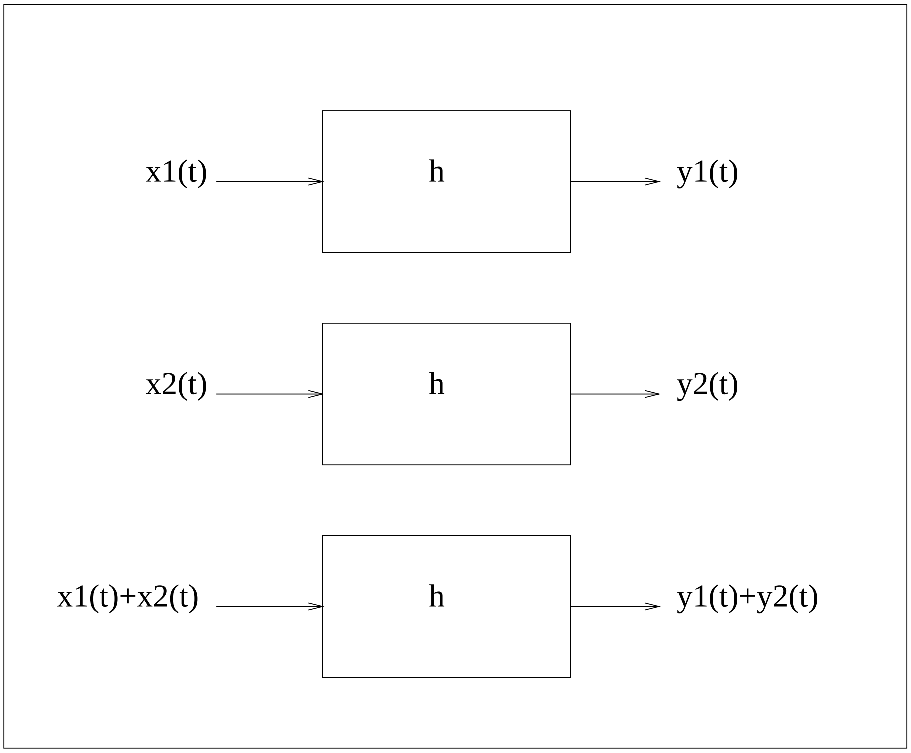


Figure 2.1: A linear System, h is the impulse response of the system.

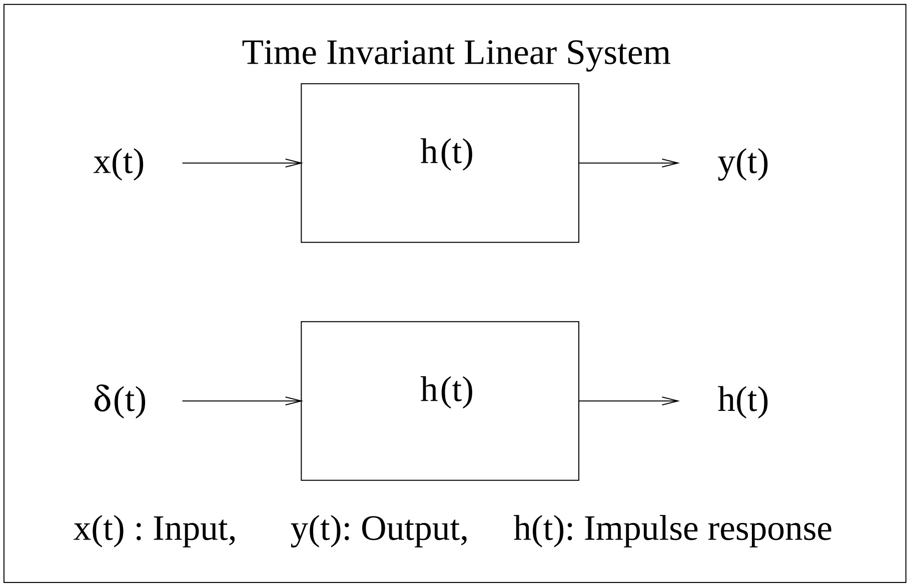


Figure 2.2: A continuous linear time invariant system. The input $x(t)$ produces an output signal denoted by $y(t)$. If the input to the system is $x(t) = \delta(t)$ the output is $y(t) = h(t)$. The signal $h(t)$ is the impulse response of the system.

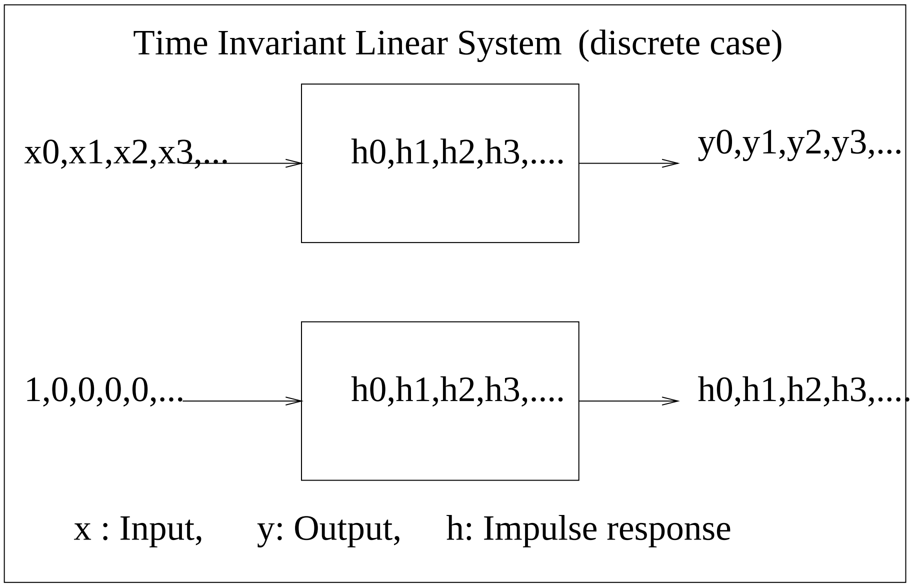


Figure 2.3: A discrete linear system. The input signal is a discrete signal x_n and the output signal is the discrete signal y_n. When the system is excited with a unit impulse signal δ_n the output is the impulse response h_n.

2.1.1 Discrete convolution

If the system is discrete, this is a system where the input and output signals are discrete signals (time series), the convolution integral becomes a summation:

$$y_k = \sum_{n=-\infty}^{\infty} h_n x_{k-n} \tag{2.9}$$

In general, we will be concerned with finite length signals. We will say that

$x_n, \;\; n = 0 : NX - 1$ is a signal of length NX

$y_n, \;\; n = 0 : NY - 1$ is a signal of length NY

$h_n, \;\; n = 0 : NH - 1$ is a signal of length NH

In this case the convolution sum will be composed only of samples defined in the above intervals, i.e., $x_n, n = 0 : NX - 1$,

$$y_k = \sum_{n=p1}^{p2} h_{k-n} x_n \,, \tag{2.10}$$

where $p1$ and $p2$ indicate the finite summation limits.
Assuming that $x = [x_0, x_1, x_2, x_3, x_4]$ and $h = [h_0, h_1, h_2]$, and after carrying out the convolution sum,

$$\begin{array}{lcllll} y_0 & = & x_0h_0 & & \\ y_1 & = & x_1h_0 & +x_0h_1 & \\ y_2 & = & x_2h_0 & +x_1h_1 & +x_0h_2 \\ y_3 & = & x_3h_0 & +x_2h_1 & +x_1h_2 \\ y_4 & = & x_4h_0 & +x_3h_1 & +x_2h_2 \\ y_5 & = & & x_4h_1 & +x_3h_2 \\ y_6 & = & & & x_4h_2 \end{array} \tag{2.11}$$

The output time series is given by $y = [y_0, y_1, y_2, ..., y_7]$. [1]
Note that the above system of equation can be written in matrix form as follows:

[1] Please, take a look at the length of the new time series $NY = NX + NH - 1$.

$$\begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \end{pmatrix} = \begin{pmatrix} x_0 & 0 & 0 \\ x_1 & x_0 & 0 \\ x_2 & x_1 & x_0 \\ x_3 & x_2 & x_1 \\ x_4 & x_3 & x_2 \\ 0 & x_4 & x_3 \\ 0 & 0 & x_4 \end{pmatrix} \begin{pmatrix} h_0 \\ h_1 \\ h_2 \end{pmatrix} \tag{2.12}$$

2.1.2 An algorithm to compute the convolution sum

One can see that the convolution sum can be carried out as a matrix times vector multiplication. But we will see that there is a cheaper way of doing it. I will do the following substitution $k - j = n$, in equation (2.10)

$$y_{j+n} = \sum_{j=0}^{NH-1} h_j x_n \,, \quad j = 0 : NX - 1 \,.\,. \tag{2.13}$$

The latter is the expression that you will need to use at the time of coding up a convolution sum.

Remember that F77 (Fortran) and MATLAB have a vector indexing system that looks like:

```
x(1) x(2) x(3) x(4) ..... x(NX)
```

Where x_0 =`x(1)`.... x_{NX-1}=`x(NX)`. This has to be taken into account at the time of writing a code. As an example I provide a MATLAB code to perform the convolution of two series. You can also use the built-in MATLAB function `conv` to perform the same task [2].

[2] Use “ help conv ”.

Convolution in MATLAB

```
%
% Convolution of x with h in MATLAB
%
x =  [2, 1, 2, 3, -1];
h =  [2,-1, 2];
NX = length(x);
NH = length(h);
NY = NX + NH - 1;
y = zeros(1,NY);
for j = 1:NH
 for n = 1:NX
  y(j+n-1) = y(j+n-1) + h(j) * x(n);
 end
end
```

The same code in Fortran 77 looks like:

```

     subroutine(nx,x,nh,h,ny,y)
c
c convolution of two time series
c
     real x(100),y(100),h(100)
     ny = nx+nh-1
     do k=1,ny
     y(k) = 0.
     enddo
     do j = 1,NH
      do n = 1,NX
       y(j+n-1) = y(j+n-1) + h(j) * x(n)
      enddo
     enddo

     return
```

2.2 The Z transform

A digitized seismogram (a gravity profile, a time series of monthly averages of temperature, etc) is a sequential collection of samples (a time series). For instance a 4 points times series is represented by the following collection of samples:

$$x_0, x_1, x_2, x_3 \tag{2.14}$$

In what follows x_n indicates the sample at time $n\Delta t$. This signal can be obtained by uniformly sampling a continuous signal periodically every Δt seconds.
The z transform of x_k, $k = 1, 2, \dots$ is defined as

$$X(z) = \sum_{k=0}^{\infty} x_n z^n \tag{2.15}$$

For a finite length time series $x_k, k = 0, \dots, N-1$ we write

$$X(z) = \sum_{k=0}^{N-1} x_n z^n \tag{2.16}$$

A simple example is a time series composed of 4 samples

$$x = \underset{\uparrow}{4}, 12, -1, -3\,, \tag{2.17}$$

where the arrow indicated the sample x_0. The z transform of this series is a polynomial in z of degree 3:

$$X(z) = 4 + 12z - 1z^2 + 3z^3\,. \tag{2.18}$$

Suppose that we have a non-casual sequence[3]

[3] I will use the arrow to indicate the sample corresponding to $t = 0$, no arrow indicates that the first sample is the $t = 0$ sample.

$$x = -1, 3, 4, \underset{\uparrow}{3}, 5, 6, -10 \tag{2.19}$$

In this case the z transform is given by

$$X(z) = -z^{-3} + 3z^{-2} + 4z^{-1} + 3 + 5z + 6z^2 - 10z^3\,. \tag{2.20}$$

2.2.1 Convolution and the Z-transform

Let us examine the example given in equation (2.11). We have two times series, $x = [x_0, x_1, x_2, x_3, x_4]$ and $h = [h_0, h_1, h_2]$. The Z-transforms of these series are:

$$X(z) = x_0 + x_1 z + x_2 z^2 + x_3 z^3 + x_4 z^4$$

$$H(z) = h_0 + h_1 z + x_2 z^2$$

Now, let us compute the product of the above polynomials:

$$\begin{aligned}
X(z).H(z) \quad = \quad & x_0 h_0 + \qquad\qquad (2.21) \\
& (x_1 h_0 + x_0 h_1) z + \\
& (x_2 h_0 + x_1 h_1 + x_0 h_2) z^2 + \\
& (x_3 h_0 + x_2 h_1 + x_1 h_2) z^3 + \\
& (x_4 h_0 + x_3 h_1 + x_2 h_2) z^4 + \qquad\qquad (2.22) \\
& (x_4 h_1 + x_3 h_2) z^5 + \\
& (x_5 h_2) z^6
\end{aligned}$$

From equation (2.10) one can see that the coefficient of this new polynomial are the samples of the time series $y = [y_0, y_1, \dots, y_6]$ obtained by convolution of x and h, in other words, $X(z)\,.H(z)$ is the also the Z transform of the time series y:

$$Y(z) = X(z)\,.H(z)\,. \tag{2.23}$$

Therefore, to convolve two time series is equivalent to multiply their Z transforms.

2.2.2 Deconvolution

We will come back to this point when dealing with seismic signals. It is clear that the convolution process in the Z-domain entails the multiplication of two polynomials. This is only feasible for short time series.
In the convolution process two time series are convolved to produce a new time series:

$$y_k = h_k \ast x_k \quad \rightarrow \quad Y(z) = H(z)\,.X(z)$$

In the deconvolution process we will attempt to estimate x_k from y_k and x_k. In the Z-domain this is equivalent to polynomial division:

$$X(z) = \frac{Y(z)}{H(z)}\,. \tag{2.24}$$

The inverse operator is defined as:

$$F(z) \;=\; \frac{1}{H(z)}\,, \tag{2.25}$$

therefore, the signal $X(Z)$ can be recovered

$$X(z) = F(z)\,.\,Y(z) \tag{2.26}$$

It is clear that if one is capable of finding $F(z) = \sum_k f_k z^k$, then the coefficients f_k define the discrete inverse filter in time domain that recovers x_k via convolution:

$$x_k \;=\; f_k \ast y_k\,. \tag{2.27}$$

This is quite important in seismological data processing. We will assume that the observed seismogram is composed of two time series: the Earth's impulse response, and the seismic wavelet (also called the source function).

s_k: Seismogram (this is what you measure)

q_k: Earth's impulse response (this is your unknown)

w_k: Wavelet (well... assume that you know it!)

where

$$s_k \ = w_k \ * q_k \, . \tag{2.28}$$

In the deconvolution process we attempt to design an inverse to remove the wavelet.

$$s_k \ = w_k \ * q_k \, . \rightarrow S(z) = W(z).Q(z)$$

if we apply the inverse filter of the wavelet to both sides of last equation we have

$$f_k * s_k \ = f_k * w_k \ * q_k \, . \rightarrow F(z).S(z) = F(z).W(z).Q(z)$$

it is clear that if $F(z) = \frac{1}{W(z)}$ the output sequence is the impulse response (our unknown)

$$q_k = f_k * s_k \, .$$

In the following sections we will analyze the problem of inverting the undesired signal (w_k).

2.3 Elementary Signals: Dipoles

In this section we will analyze the deconvolution of very simple signals. We will see that by understanding how to work with simple signals we will be capable of dealing with more complicated signals.

2.3.1 Minimum phase dipoles

A simple manner of visualizing the properties of a time series in the z domain is by decomposing the polynomial into dipoles or elementary functions of the type

$$1 + az \tag{2.29}$$

As an example, we compute the Z-transform of the series $x = [4, 12, -1, 3]$:

$$X(z) = 4 + 12z - 1z^2 + 3z^3 = 4(1 + \frac{1}{2}z)(1 - \frac{1}{2}z)(1 + 3z)\,. \tag{2.30}$$

We have already seen that two multiply the Z-transform of two time series is equivalent to convolve the time series in the time domain. Therefore, the above expression can also be expressed as convolution of several time series:

$$4, 12, -1, 3 = 4[\,(1, \frac{1}{2}) * (1, -\frac{1}{2}) * (1, 3z)\ . \tag{2.31}$$

In order to simplify the problem, we will analyze the properties of a single dipole. The extension to time series that require the multiplication of several dipoles is straightforward.

Let us assume that the dipole, which I will call $D(z)$, is given by

$$D(z) = 1 + az\,. \tag{2.32}$$

This dipole corresponds to a time series composed of two elements: $1, a$. Now, let assume that we want to compute the inverse of the dipole, in other words we would like to compute a function $F(z)$ such that

$$F(z)D(z) = 1\,. \tag{2.33}$$

This problem can be solved by expanding the inverse of the dipole in a series:

$$F(z) = \frac{1}{D(z)} = \frac{1}{1 + az}\,, \tag{2.34}$$

if $|a| < 1$ the denominator can be expanded according to the following expression [4]:

$$F(z) = 1 - az + (az)^2 - (az)^3 + (az)^4 \dots \tag{2.35}$$

[4] A geometric series.

Since $|a| < 1$ the above series is a convergent series. $F(z)$ is the z transform of the time series $f_k, k = 0, \dots, \infty$:

$$\underset{\uparrow}{1}, -a, a^2, -a^3, a^4, \dots \tag{2.36}$$

which represent the inverse filter of the dipole. The convolution of the dipole with the filter yields

$$(\underset{\uparrow}{1}, a) * (\underset{\uparrow}{1}, -a, a^2, -a^3, a^4, \dots) = \underset{\uparrow}{1}, 0, 0, 0, 0, 0, \dots \tag{2.37}$$

which represent a single spike at $n = 0$.

The dipole $(1, a)$ is a minimum phase sequence provided that $|a| < 1$. We have shown that a minimum phase dipole has a casual inverse given by $1, -a, a^2, -a^3, a^4, \dots$ If $|a| \approx 1 < 1$ the coefficients of the inverse filter will slowly tend to zero. On the other hand if $|a| \approx 0$ only a few coefficient will be required to properly model the inverse of the dipole.

We can visualize this fact with a very simple example. Let us compute the inverse of the following dipoles: (1,0.9) and (1, 0.01). In the first case we have $a = 0.9$:

$$F(z) = \frac{1}{1 + 0.9z} = 1 - 0.9z + 0.81z^2 - 0.729z^3 + 0.6561z^4 \dots . \tag{2.38}$$

In the second case, we have:

$$F(z) = \frac{1}{1 + 0.1z} = 1 - 0.1z + 0.01z^2 - 0.001z^3 + 0.0001z^4 \dots . \tag{2.39}$$

It is clear that when $a = 0.1$ we can truncate our expansion without affecting the performance of the filter. To show the last statement we convolve the dipoles with their truncated inverses. In both examples, we truncate the inverse to 5 coefficients:

$$(1, 0.9) * (1, -0.9, 0.81, -0.729, 0.6561) = (1, 0.0, 0.0, 0.0, 0.59) \tag{2.40}$$

$$(1, 0.1) * (1, -0.1, 0.01, -0.001, 0.0001) = (1, 0.0, 0.0, 0.0, 0.0)\,. \tag{2.41}$$

It is clear that the truncation is negligible when $a = 0.1$. This is not true when $a \approx 1$. In this case a long filer is needed to properly invert the dipole. The aforementioned shortcoming can be overcome by adopting a least squares strategy to compute the inverse filter (this is the basis of *spiking deconvolution*.)

So far we have define a minimum phase dipole as a signal of the type $(1, a)$ where $|a| < 1$. It is important to stress that the Z-transform of this signal has a root, ξ, which lies outside the unit circle,

$$X(z) = 1 + az \;\Rightarrow X(\xi) = 1 + a\xi = 0 \;\Rightarrow \xi = -\frac{1}{a} \tag{2.42}$$

since $|a| < 1$, the root satisfies the following $|\xi| > 1$.

A seismic signal is more complicated than a simple dipole. But we can always factorize the Z-transform of the signal in terms of elementary dipoles. If the signal is minimum phase, the decomposition is in terms of minimum phase dipoles

$$X(z) = x_0 + x_1 z + x_2 z^2 + x_3 z^3 \ldots = A(1 + a_1 z)(1 + a_2 z)(1 + a_3 z) \ldots. \tag{2.43}$$

If $|a_i| < 1$, $\forall i$, the signal is a minimum phase signal. In this case all the zeros lie outside the unit circle

$$X(\xi) = 0 \;\Rightarrow \xi_i = -\frac{1}{a_i} \;\Rightarrow |a_i| < 1 \;\Rightarrow |\xi_i| > 1\,. \tag{2.44}$$

Now, let us assume that $X(z)$ is a minimum phase signal of length N, that can be factorized in terms of minimum phase dipoles. The inverse filter $F(z)$ of $X(z)$ must satisfied the following expression:

$$
\begin{aligned}
&X(z)F(z) = 1 \qquad (2.45)\\
&(1 + a_1 z)(1 + a_2 z)(1 + a_3 z) \ldots . F(z) = 1 \,.
\end{aligned}
$$

From the above equation where we can write

$$
\begin{aligned}
F(z) &= (1 + a_1 z)^{-1}(1 + a_2 z)^{-1}(1 + a_3 z)^{-1} \ldots \qquad (2.46)\\
&= [(1 - a_1 z + (a_1 z)^2 - (a_1 z)^3 \ldots][(1 - a_2 z + (a_2 z)^2 - (a_2 z)^3 \ldots]\\
&\quad [(1 - a_3 z + (a_3 z)^2 - (a_3 z)^3 \ldots] \ldots .
\end{aligned}
$$

It is clear that the inverse operator can be written as

$$
f_0, f_1, f_2, f_3, \ldots = (1, -a_1, a_1^2, -a_1^3) * (1, -a_2, a_2^2, -a_2^3) * (1, -a_3, a_3^2, -a_3^3) * \ldots \qquad (2.47)
$$

In Figures 2.4, 2.5 and 2.6, we examine the inverse of various minimum phase dipoles. In the first case the root is close to the unit circle, and therefore the inverse filter requires a large number of coefficient to avoid truncation artifacts. It is clear in the output sequence (the convolution of the dipole with the filter) that the truncation has introduced a spike at the end of the sequence. In Figures 2.5 and 2.6, we have used dipoles with roots $\xi = 2$ and $\xi = 10$, respectively. In these examples the truncation artifacts are minimal.

2.3.2 Maximum phase dipoles

Elementary signal of the form $(1, b)$, $|b| > 1$ are called maximum phase dipoles. A maximum phase dipole has a zero inside the unit circle:

$$
D(z) = 1 + bz \Rightarrow D(\xi) = 1 + b\xi = 0 \Rightarrow \xi = -1/b \,. \qquad (2.48)
$$

Since $|b| < 1$, it is easy to see that $|\xi| < 1$.

In this section we will prove that the inverse of a maximum phase dipole is a non-casual sequence. The inverse of the maximum phase dipole can be computed by expanding the denominator in series

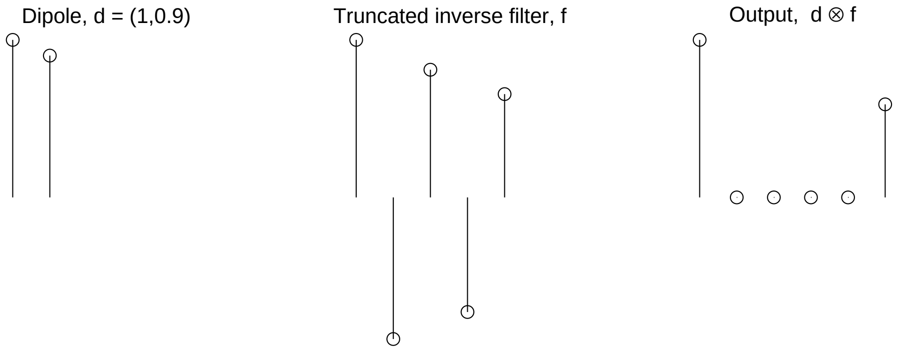


Figure 2.4: Inversion of minimum phase dipoles. The slow convergence of the inverse filter is a consequence of having a zero close to the unit circle.

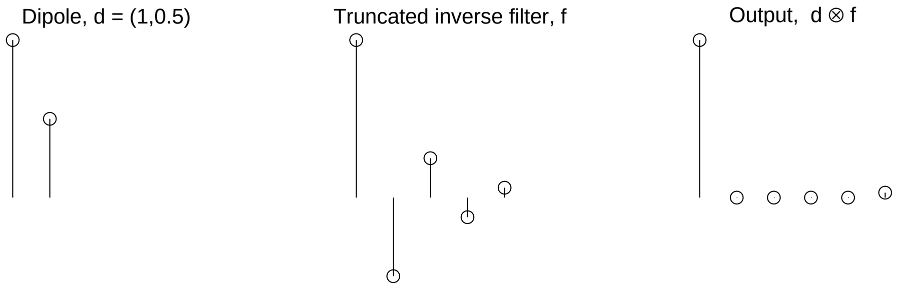


Figure 2.5: Inversion of minimum phase dipoles

$$F(z)D(z) = 1 \Rightarrow F(z) = \frac{1}{D(z)} = \frac{1}{1+bz} \tag{2.49}$$

If last equation is expanded in a series of positive powers of z we have

$$\frac{1}{1+bz} = 1 - bz + (bz)^2 - (bz)^3 \dots \tag{2.50}$$

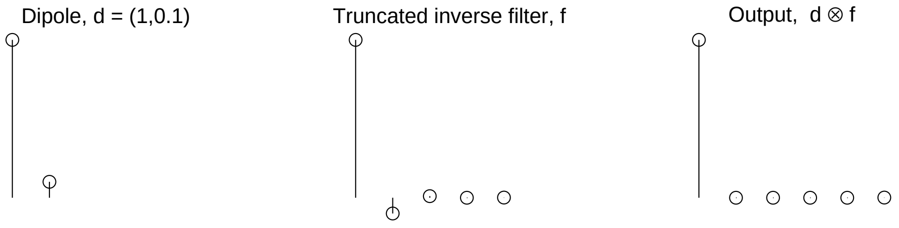


Figure 2.6: Inversion of minimum phase dipoles, in this case the zero of the dipole is far from the unit circle, this explains the fast convergence of the inverse filter.

The later is a series that does not converge; the magnitude of the coefficients of the operator $(1, -b, b^2, -b^3 \ldots)$ increases with time. The trick to overcome this problem is to compute a *stable non-casual* operator. First, we rearrange expression (2.49)

$$F(z) = \frac{1}{1+bz} = \frac{1}{bz(1+(bz)^{-1})} \tag{2.51}$$

this expression admits an stable expansion of the form

$$F(z) = (bz)^{-1}(1 - (bz)^{-1} + (bz)^{-1} - (bz)^{-3} \ldots)\,. \tag{2.52}$$

Now the inverse is stable and non-casual, the associated operator is given by

$$f = \ldots, -b^{-3}, b^{-2}, -b^{-1}, \underset{\uparrow}{0} \tag{2.53}$$

The following example will clarify the problem. First, given the maximum phase dipole $(1, 2)$ we compute the non-casual inverse sequence (truncated to 6 coefficients):

$$f = (-0.0156, 0.0312, -0.0625, 0.125, -0.25, 0.5, \underset{\uparrow}{0}) \tag{2.54}$$

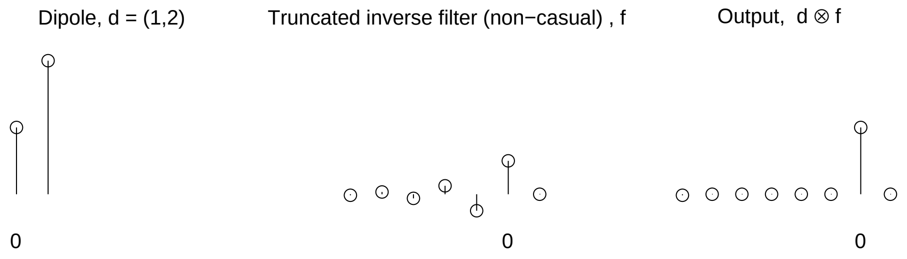


Figure 2.7: Maximum phase dipole, its non-casual truncated inverse, f, and the output $d * f$.

the convolution of f with the maximum phase dipole produces the following output sequence

$$\begin{aligned} d * f &= (-0.0156, 0.0312, -0.0625, 0.125, -0.25, 0.5, \underset{\uparrow}{0}) * (\underset{\uparrow}{1}, 2) \\ &= (-0.0156, 0, 0, 0, 0, 0, \underset{\uparrow}{1}, 0) \end{aligned} \tag{2.55}$$

2.3.3 Autocorrelation function of dipoles

The autoccorelation function of a sequence with z-transform $X(z)$ is defined as

$$R(z) = X(z)X^*(z^{-1}) \tag{2.56}$$

In this section we will analyze some properties of minimum and maximum phase dipoles that are very useful at the time of designing deconvolution operators.
We will consider two dipoles a minimum phase dipole of the form $(1, a)$, $|a| < 1$ and a maximum phase dipole of the form $(a^*, 1)$[5]. In the z domain we have

$$D_{min}(z) = 1 + az \tag{2.57}$$

$$D_{max}(z) = a^* + z \tag{2.58}$$

The autocorrelation function for the minimum phase sequence is given by:

$$R_{min}(z) = a^*z^{-1} + (1 + |a|^2) + a\,z\,, \tag{2.59}$$

Similarly, the autocorrelation function for the maximum phase dipole is given by

$$R_{max}(z) = a^*z^{-1} + (1 + |a|^2) + a\,z\,. \tag{2.60}$$

We have arrived to a very important conclusion

$$R_{max}(z) = R_{min}(z) = R(z) \tag{2.61}$$

[5] Note that for real dipoles, $a^* = a$

or in other words, two different sequences can have the same autocorrelation function. The autocorrelation sequence in both cases is the following time series

$$a^*, (1 + \underset{\uparrow}{a^2}), a \tag{2.62}$$

or

$$r_k = \begin{cases} a & \text{if } k = 1 \\ 1 + a^2 & \text{if } k = 0 \\ a^* & \text{if } k = -1 \\ 0 & \text{otherwise} \end{cases} \tag{2.63}$$

If the dipoles are real ($a = a^*$), the autocorrelation function is a a symmetric sequence about zero. Note that the autocorrelation function $R(z)$ is the Z-transform of the autocorrelation sequence.

$$R(z) = r_1 z^{-1} + r_0 + r_1 z^{-1} = a^* z^{-1} + (1 + a^2) + a z^{-1} \tag{2.64}$$

In general for more complicated signals (so far we only considered dipoles), the autocorrelation function of the the signal is the Z-transform of the autocorrelation sequence of the signal given by

$$r_k = \sum_n x_n^* x_{n+k} \,, \tag{2.65}$$

$$R(z) = X(z)\,.X^*(z^{-1})\,, \tag{2.66}$$

where k is the time-lag of the autocorrelation function.

Let's assume that we are only able to measure the autocorrelation of a dipole. Given the autorelation of the dipole you are asked to find the associated dipole. It is clear that you have two possible solutions. One is the minimum phase dipole; the other is the maximum phase dipole. It is also true that this two sequences have the same amplitude spectrum. We define the amplitude spectrum.

$$R(\omega) = R(z)|_{z=e^{-i\omega}} \tag{2.67}$$

or

$$R(\omega) = [X(z)\,.X^*(z^{-1})]_{z=e^{-i\omega}} \tag{2.68}$$

To evaluate the amplitude spectrum of the signal we replace z by $e^{-i\omega}$. This is equivalent to use the discrete Fourier transform instead of the z transform. We will come back to this point in Chapter 3. If the signal is a minimum phase dipole:

$$D_{min}(z) = 1 + az \;\Rightarrow z = e^{-i\omega} \;\Rightarrow D_{min}(\omega) = 1 + ae^{-iw}\,. \tag{2.69}$$

Whereas for the maximum phase dipole we have

$$D_{max}(z) = a + z \;\Rightarrow z = e^{-i\omega} \;\Rightarrow D_{max}(\omega) = a + 1e^{-iw} \tag{2.70}$$

Now we are in condition of evaluating the amplitude and phase spectrum of the minimum and maximum phase dipoles:

$$R_{D_{min}}(\omega) = \sqrt{1 + 2a\,\cos(\omega) + a^2} \tag{2.71}$$

$$\theta_{min}(\omega) = \arctan(\frac{a\,\sin(\omega)}{1 + a\,\cos(\omega)}) \tag{2.72}$$

For the maximum phase signal we have

$$R_{D_{max}}(\omega) = \sqrt{1 + 2a\,\cos(\omega) + a^2} \tag{2.73}$$

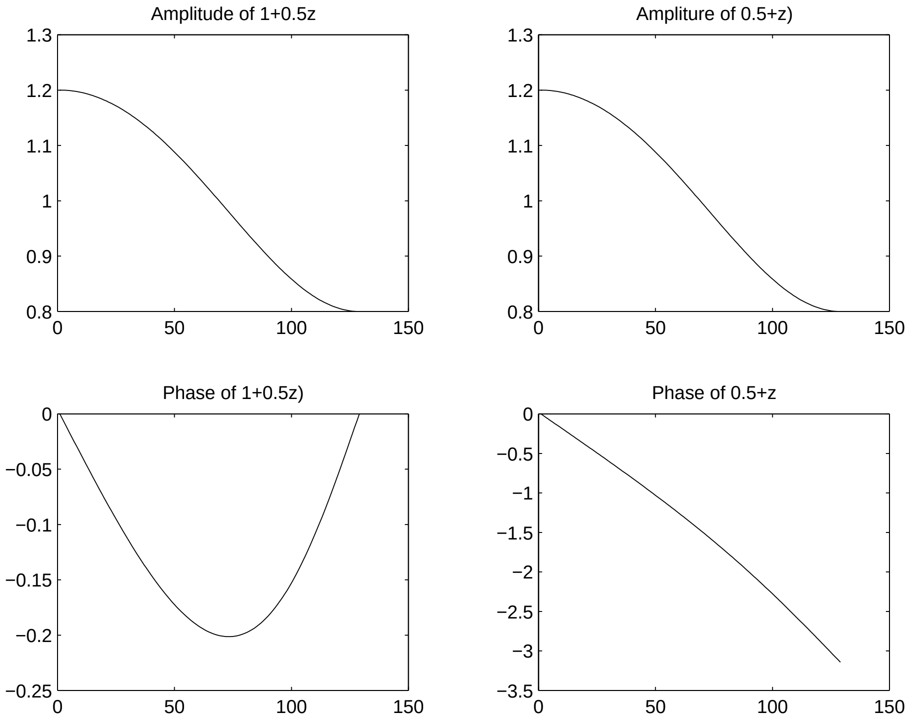


Figure 2.8: Amplitude and phase spectrum of a minimum phase dipole $1+az$ and a maximum phase dipole $a*+z$, $|a|<1$.

$$\theta_{max}(\omega) = \arctan(\frac{\sin(\omega)}{a+\cos(\omega)}) \tag{2.74}$$

In Figure (2.8) we portray the amplitude and phase spectrum for a minimum phase dipole of the form $(1, 0.5)$ and a maximum phase dipole $(0.5, 1)$. Note that the amplitude spectra of these signals are equal.

2.3.4 Least squares inversion of a minimum phase dipole

We have already seen that one of the problems of inverting a dipole via an expansion of the denominator in terms of a series is that the inverse filter can result in a long operator. This is particularly true when we have a zero close to the unit circle.

Our problem is to find a filter where, when applied to a dipole, it's output resembles the ideal output one would have obtained by using an infinite number of terms in the series expansion of the filter.

In our case, we want to invert the minimum phase dipole $(1, a)$, $|a| < 1^6$. In a preceding section we found an expression for the ideal inverse filter, the z transform of the ideal inverse filter satisfies the following equation:

$$D(z)F(z) = 1\,. \tag{2.75}$$

Now, our task is to construct a finite length filter that with the following property

$$D(z)F_N(z) \approx 1\,, \tag{2.76}$$

where $F_N(z)$ denotes the z transform of the finite length operator. The above equation can be written in the time domain as follows (assuming $N = 3$),

$$(1, a) * (f_0, f_1, f_2) \approx (\underset{\uparrow}{1}, 0, 0, 0)\,. \tag{2.77}$$

The latter can be written in matrix form as follows:

$$\begin{pmatrix} 1 & 0 & 0 \\ a & 1 & 0 \\ 0 & a & 1 \\ 0 & 0 & a \end{pmatrix} \begin{pmatrix} f_0 \\ f_1 \\ f_2 \end{pmatrix} \approx \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}\,. \tag{2.78}$$

[6] Let's assume that $a = a^*$, (a is real)

The last system of equations corresponds to an over-determined system of equation that can be solved using least squares. In order to avoid notational clutter we will represent the last system as follows

$$\mathbf{C}\mathbf{f} \approx \mathbf{b}\,, \tag{2.79}$$

where $\mathbf{C}$ is the matrix that contains the dipole properly padded with zeros in order to properly represent the convolution $(*)$. The unknown inverse filter is denoted by the vector $\mathbf{f}$ and the desire output by $\mathbf{d}$. It is clear that the solution vector is the one that minimized the mean squared error

$$\epsilon = ||\mathbf{C}\,\mathbf{f} - \mathbf{b}||^2\,. \tag{2.80}$$

The least squares solution of this system is found by solving the following system of normal equations

$$\mathbf{C}^T\,\mathbf{C}\,\mathbf{f} = \mathbf{C}^T\,\mathbf{b}\,. \tag{2.81}$$

Now the we have a system of normal equations (a square system) that can be inverted by any method. The resulting filter is

$$\mathbf{f} = \mathbf{R}^{-1}\mathbf{C}^T\,\mathbf{b}\,, \tag{2.82}$$

where $\mathbf{R} = \mathbf{C}^T\,\mathbf{C}$. The story does not end here, it turns out that the matrix $\mathbf{R}$ has a special structure,

$$\mathbf{R} = \left(\begin{array}{ccc} 1+a^2 & a & 0 \\ a & 1+a^2 & a \\ 0 & a & 1+a^2 \end{array} \right)\,. \tag{2.83}$$

One can see that each row of the matrix $\mathbf{R}$ is composed by elements of the autocorrelation sequence given by equation (2.63)

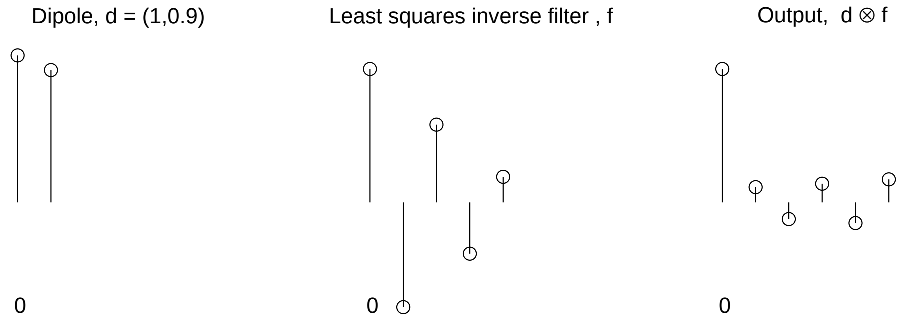


Figure 2.9: Inversion of a minimum phase dipole using least squares.

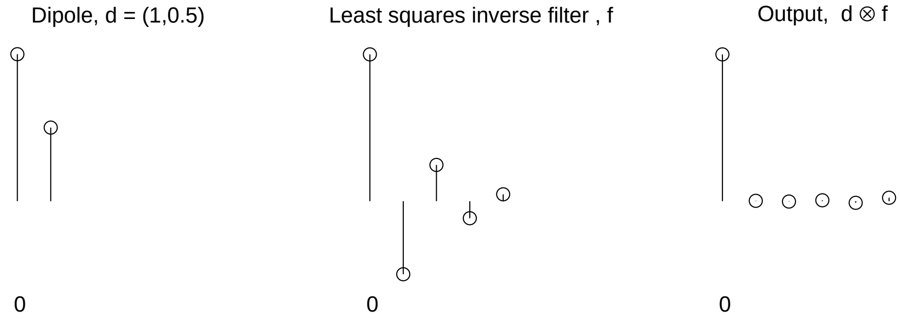


Figure 2.10: Inversion of a minimum phase dipole using least squares.

$$\mathbf{R} = \begin{pmatrix} r_0 & r_1 & 0 \\ r_1 & r_0 & r_1 \\ 0 & r_1 & r_0 \end{pmatrix} . \qquad (2.84)$$

The above matrix is a Toeplitz form. One interesting feature of a Toeplitz matrix (in this case a Hermitian Toeplitz matrix) is that only one row of the matrix is needed to define all the elements of the matrix. This special symmetry is used by a fast algorithm, the Levinson algorithm, to invert the matrix **R**.

It is interesting to note that the condition number of the Toeplitz matrix[7]

[7] The condition number of the matrix **R** is the ratio $\lambda_{max}/\lambda_{min}$ where λ_{max} and λ_{min} are the largest and smallest eigenvalues of **R**, respectively. A large condition number

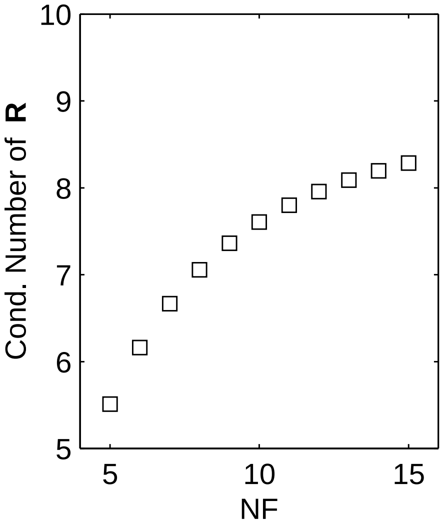


Figure 2.11: Condition number of the Toeplitz matrix versus NF (filter length). The dipole is the minimum phase sequence $(1, 0.5)$.

increases with NF (the filter length). This is shown in Figure (2.11). Similarly in Figure (2.12) we portray the condition number of the Toeplitz matrix for a dipole of the form $(1, a)$ for different values of the parameter a. It is clear that when $a \to 1$ the zero of the dipole moves towards the unit circle and the system of equation becomes ill-conditioned.

indicates that the problem is ill-conditioned (numerical problems will arise at the time of inverting the matrix.)

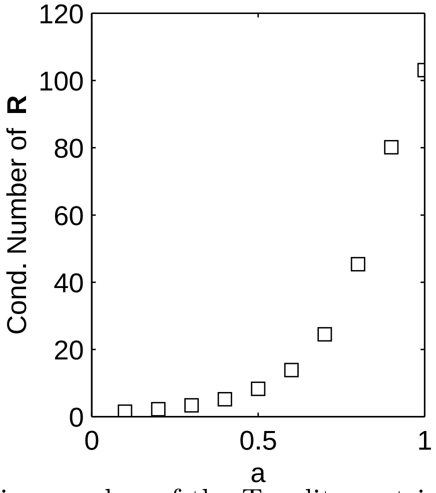


Figure 2.12: Condition number of the Toeplitz matrix versus a for a minimum phase dipole $(1, a)$. The length of operator is fixed to $NF = 15$.

2.3.5 Inversion of Minimum Phase sequences

So far we have discuss the problem of inverting elementary dipoles, and we have observed that minimum phase dipoles accept a casual and stable inverse.

This is also valid for more complicated signals (i.e., a seismic wavelet). The columns of the convolution matrix are wavelets of length NW instead of dipoles of length 2.

Given a minimum phase wavelet, this a signal that can be decomposed trough factorization in minimum phase dipoles[8], we would like to find the inverse operator. This is, again, the filter that converts the wavelet into a spike. Given the wavelet $w_k, k = 1, \dots, NW$, the filter $f_k, k = 1 \dots, NF$ needs to be designed to satisfy the following equation:

$$(w_0, w_1, \dots w_{NW-1}) * (f_0, f_1, \dots, f_{NF-1}) \approx (1, 0, \dots, 0) \tag{2.85}$$

In matrix form we can write the following expression (assuming $NW = 7$ and $NF = 4$)

$$\begin{pmatrix} w_0 & 0 & 0 & 0 \\ w_1 & w_0 & 0 & 0 \\ w_2 & w_1 & w_0 & 0 \\ w_3 & w_2 & w_1 & w_0 \\ w_4 & w_3 & w_2 & w_1 \\ w_5 & w_4 & w_3 & w_2 \\ w_6 & w_5 & w_4 & w_3 \\ 0 & w_6 & w_5 & w_4 \\ 0 & 0 & w_6 & w_5 \\ 0 & 0 & 0 & w_6 \end{pmatrix} \begin{pmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \end{pmatrix} \approx \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} . \tag{2.86}$$

Again, this system is written in matrix form as $\mathbf{Cf} \approx \mathbf{d}$. We will compute the inverse filter by minimizing the error function (mean squared error) ϵ:

$$\epsilon = ||\mathbf{e}||^2 = ||\mathbf{C}\,\mathbf{f} - \mathbf{b}||^2; \tag{2.87}$$

[8] In other words, all the zeros of the z transform of the wavelet lie outside the unit circle.

The Euclidean norm of the error vector $\mathbf{e} = \mathbf{C}\,\mathbf{f} - \mathbf{b}$ can be written down as

$$\epsilon = \mathbf{e}^T\,\mathbf{e} = (\mathbf{C}\,\mathbf{f} - \mathbf{b})^T\,(\mathbf{C}\,\mathbf{f} - \mathbf{b})\,. \tag{2.88}$$

The mean squared error is minimized when the following condition if satisfied:

$$\frac{d\epsilon}{d\mathbf{f}} = 0\,, \tag{2.89}$$

Taking derivatives with respect to the filter coefficients and equating them to zero leads to the following system of normal equations

$$\mathbf{C}^T\,\mathbf{C}\,\mathbf{f} = \mathbf{C}^T\,\mathbf{b}\,. \tag{2.90}$$

It is clear that the inverse filter is solved by inverting the Toeplitz form $\mathbf{R} = \mathbf{C}^T\,\mathbf{C}$, but this matrix (which depends on the wavelet) might have a set of eigenvalues which are close to zero.

If the matrix is ill-conditioned there exists a set of eigenvalues that are zero or close to zero. This will lead to numerical unstabilties at the time of inversion. This shortcoming can be avoided by using a regularization strategy. Instead of minimizing the misfit function ϵ we will minimize the following penalized objective function

$$J = \epsilon + \mu||\mathbf{f}||^2\,, \tag{2.91}$$

The solution is now given by a penalized least squares estimator where the parameter μ is also called the regularization parameter (also: ridge regression parameter or pre-whitening parameter). The condition

$$\frac{dJ}{d\mathbf{f}} = 0\,, \tag{2.92}$$

leads to the following solution

$$\mathbf{f} = (\mathbf{R} + \mu \mathbf{I})^{-1} \mathbf{C}^T \mathbf{d}\,. \tag{2.93}$$

It is clear that the parameter μ is a protection against small eigenvalues which may lead to an unstable filter. It is important to note that in the objective function J we are trying to accomplished two different wishes. On one hand we want to minimize the error function ϵ , on the other hand we try to keep the energy of the filter bounded. When $\mu \to 0$ the error function will be minimum but the filter may have an undesired oscillatory behavior. When μ is large the energy of the filter will be small and the misfit function ϵ will be large. In this case we have a matching filter of the form

$$\mathbf{f} = \mu^{-1} \mathbf{C}^T \mathbf{d}\,. \tag{2.94}$$

Last equation was obtained by doing the following replacement $(\mathbf{R} + \mu \mathbf{I}) \approx \mu \mathbf{I}$, which is valid only when μ is large.

In Figure (2.13) we illustrate the effect of the tradeoff parameter in the filter design and in the actual output of the deconvolution. It is clear that when μ is small the output sequence is a spike, when we increase μ the output sequence is a band-limited signal. This concept is very important when dealing with noisy signals. We will come back latter to this problem when we analyze the deconvolution of reflectivity sequences.

In Figure (2.14) we portray the so called tradeoff curve. This is a curve where we display the misfit, ϵ, versus the norm of the filter $||\mathbf{f}||^2$ for various values of the tradeoff parameter μ. This curve is also called the Tikkonov curve or the L-curve, this is a popular method to analyze the tradeoff that exists between resolution and variance in linear inverse problems.

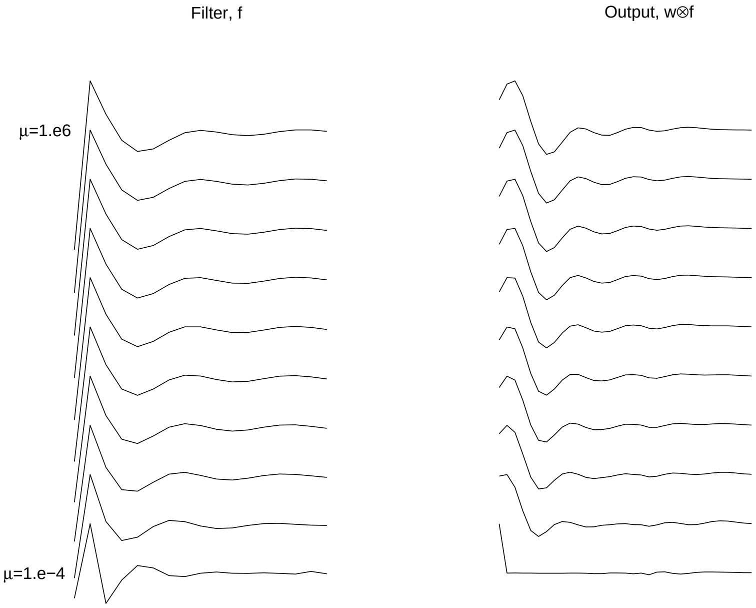


Figure 2.13: A minimum phase wavelet inverted using different tradeoff parameters (μ).

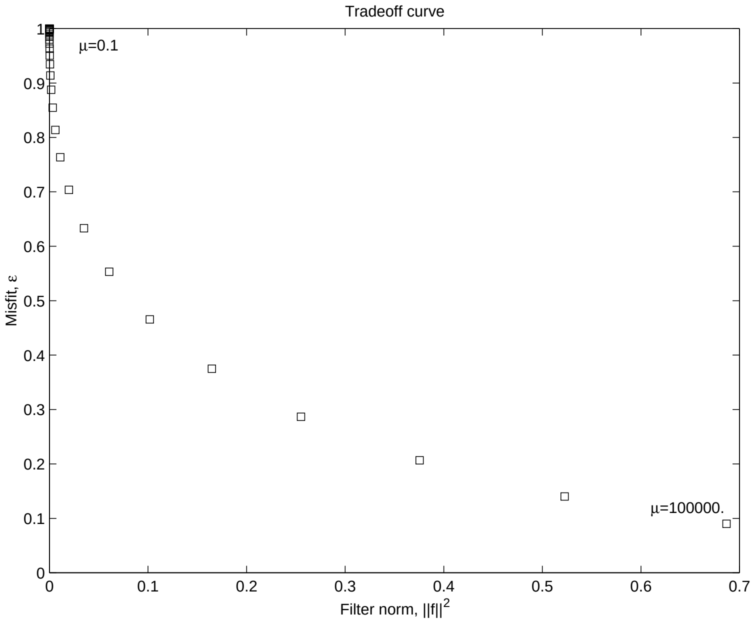


Figure 2.14: Tradeoff curve for the previous example. The vertical axis indicated the misfit and the horizontal the norm of the filter.

2.4 MATLAB codes used in Chapter 2

2.4.1 Inversion of dipoles

This code was used to generate Figures (2.4), (2.5), and (2.6).

```
% Dipole.m
% This code is used to invert a min. phase dipole
% The inverse is truncated to N samples

N = 5;                % Length of the inverse filter
a = 0.1;              % coeff. of the dipole
t = 1:1:N;            % Time  samples
d = [1 a];            % Dipole
f = (-a).^(t-1);      % Inverse filter
o = conv(d,f)         % Compute the output d*f

% Plot The dipole, the filter and the output

figure(1); stem(d); figure(2); stem(f); figure(3); stem(o);
```

2.4.2 Amplitude and phase

This code was used to generate Figure (2.8).

```
% Minmax.m
% A MATLAB code to compute amplitude and phase
% of min and max phase dipoles

a = 0.2;
d_min = [1,a];        % Min phase dipole
d_max = [a,1];        % Max phase dipole

% Compute amplitude and phase using an FFT
```

```
D_min = fft(d_min,256); A_min = abs(D_min); theta_min = angle(D_min);
D_max = fft(d_max,256); A_max = abs(D_max); theta_max = angle(D_max);

% Plot the results

n = 256/2+1;
subplot(221); plot(A_min(1:n));title('Amplitude of 1+0.2z')
subplot(222); plot(A_max(1:n));title('Amplitude of 0.2+z)')
subplot(223); plot(unwrap(theta_min(1:n))); title('Phase of 1+0.2z)')
subplot(224); plot(unwrap(theta_max(1:n))); title('Phase of 0.2+z')
```

2.4.3 Least squares inversion of a dipole

This code was used to generate Figures (2.9) and (2.10).

```
% LS_dipole.m
% Least squares inversion of a
% min. phase dipole

NF = 5;                   % length of the filter
a = 0.5;
d = [1,a]';               % Data (the dipole)
ND = max(size(d)) ;       % Length of the data
NO = ND+NF-1              % length of the output
b = [1,zeros(1,NO-1)]';   % Desire output
C = convmtx(d,NO-1);      % Convolution matrix

R = C'*C;                 %  Toeplitz Matrix
rhs = C'*b;               %  Right hand side vector
f = inv(R)*rhs;           %  Filter
o = conv(f,d);            %  Actual output
figure(1); stem(d); figure(2); stem(f); figure(3); stem(o);
```

2.4.4 Eigenvalues of the Toeplitz matrix

This code was used to generate Figures (2.11) and (2.12).

```
% Eigen_dipole.m
% Condition number versus filter length.
% for a min. phase dipole

for NF = 5:15
a = 0.5;
d = [1,a]';              % Data (the dipole)
ND = max(size(d)) ;
NO = ND+NF-1
C = convmtx(d,NO-1);    % Convolution matrix
R = C'*C;               % Toeplitz Matrix
Eigen = eig(R);         % Eigenvalues of the Toeplitz Matrix
Cond = max(Eigen)/min(Eigen);
subplot(231);plot(NF,Cond,'s'); hold on;
end
```

2.4.5 Least square inverse filters

Program used to obtain Figure (2.14).

```
function [f,o] = LS_min(w,NF,mu);
% LS_min.m
% Given an input wavelet w this programs
% computes the wavelet inverse filter
% and the actual output o.
% NF is the filter length.
% Note that w is a column wavelet
% mu is the pre-whitening
NW = max(size(w));        % length of the wavelet
NO = NW+NF-1
```

```
b = [1,zeros(1,NO-1)]';  % Desire output
C = convmtx(w,NF);       % Convolution matrix
R = C'*C;                % Toeplitz Matrix
rhs = C'*b;              % Right hand side vector
I = eye(R)*mu;
f = inv(R+I)*rhs;        % Filter
o = conv(f,w);           % Actual output
return
```

2.5 The autocorrelation function

Consider a time series of the form

$$X(z) = x_0 + x_1 z + x_2 z^2$$

and compute the following function (autocorrelation function)

$$R(z) = X(z)\, X^*(z^{-1}) \tag{2.95}$$

$$R(z) = x_0 x_2^* z^{-2} + (x_0 x_1^* + x_1 x_2^*) z^{-1} + (x_0 x_0^* + x_1 x_1^* + x_2 x_2^2) + (x_1 x_0^* + x_2 x_1^*) z + x_2 x_0^* z^2 \,. \tag{2.96}$$

The function $R(z)$ is the Z-transform of a sequence r_k that we call the autocorrelation sequence:

$$R(z) = \sum_{k=-\infty}^{\infty} r_k z^k \tag{2.97}$$

where

$$\begin{array}{rcl} r_{-2} & = & x_0 x_2^* \\ r_{-1} & = & x_0 x_1^* + x_1 x_2^* \\ r_0 & = & x_0 x_0^* + x_1 x_1^* + x_2 x_2^* \\ r_1 & = & x_1 x_0^* + x_2 x_1^* \\ r_2 & = & x_2 x_0^* \\ r_k & = & 0 \quad \text{otherwise}\,. \end{array} \tag{2.98}$$

It is easy to show that for a time series of length NX

$$x_0, x_1, x_2, x_3, \dots, x_{NX-1}$$

the autocorrelation coefficient can be computed using the following formulas:

$$\begin{aligned} r_{-k} &= \textstyle\sum_{i=0}^{NX-1-k} x_i x_{i+k}^* \quad k = 1,2,3,\dots,NX-1 \\ r_0 &= \textstyle\sum_{i=0}^{NX-1} x_i x_i^* \qquad (2.99) \\ r_k &= \textstyle\sum_{i=0}^{NX-1-k} x_{i+k} x_i^* \quad k = 1,2,3,\dots,NX-1 \quad \text{[Note]}^9 \end{aligned}$$

Properties of the autocorrelation sequence:

1. Hermitian Symmetry: $r_k = r_{-k}^* \quad k = \pm 1, \pm 2, \dots$
2. $r_0 > |r_k| \quad k = \pm 1, \pm 2 \dots$
3. r_0 represents the energy of the signal; for a zero mean stationary stochastic process r_0/NX is an estimator of the variance of the process:

$$\hat{\sigma}^2 = \frac{r_0}{NX} = \frac{\sum_{k=0}^{NX-1} |x_k|^2}{NX}\,.$$

4. If $x_0, x_1, \dots, x_{NX-1}$ is a real time series then, $r_k = r_{-k}$.

2.5.1 The Toeplitz matrix and the autocorrelation coefficients

In section 2.5.5 we have use the method of least squares to find an inverse operator that enables us to collapse a wavelet into a spike. We have seen that the least squares filter is computed by solving a system of equations of the form

$$\mathbf{C}^T\mathbf{C}\mathbf{f} = \mathbf{C}\mathbf{b} \qquad (2.100)$$

Where **C** is a matrix with entries given by the wavelet properly pad with zeros and shifted in order to represent a convolution operator, in our example

$$\mathbf{C} = \begin{pmatrix} w_0 & 0 & 0 & 0 \\ w_1 & w_0 & 0 & 0 \\ w_2 & w_1 & w_0 & 0 \\ w_3 & w_2 & w_1 & w_0 \\ w_4 & w_3 & w_2 & w_1 \\ w_5 & w_4 & w_3 & w_2 \\ w_6 & w_5 & w_4 & w_3 \\ 0 & w_6 & w_5 & w_4 \\ 0 & 0 & w_6 & w_5 \\ 0 & 0 & 0 & w_6 \end{pmatrix} \tag{2.101}$$

This is the convolution matrix for a wavelet or length $NW = 7$ and a filter of length $NF = 4$. It is easy to see that the Toeplitz matrix $\mathbf{R} = \mathbf{C}^T\mathbf{C}$ is given by

$$\mathbf{R} = \begin{pmatrix} r_0 & r_1 & r_2 & r_3 \\ r_1 & r_0 & r_1 & r_2 \\ r_2 & r_1 & r_0 & r_1 \\ r_3 & r_2 & r_1 & r_0 \end{pmatrix} \tag{2.102}$$

where the elements of **R** are given by:

$$r_k = \sum_{i=0}^{NW-1-k} w_{i+k}\, w_i \qquad k = 0, 1, 2, 3, \ldots, NF-1 \tag{2.103}$$

The coefficients r_k are the **correlation coefficients** the wavelet (compare this result with equation (2.99)). It is interesting to note that the zero lag correlation coefficient (k=0) represents the energy of the wavelet:

$$r_0 = \sum_{i=0}^{NW-1} w_i^2 \tag{2.104}$$

It is important to stress that at the time of computing the Toeplitz matrix we do not need to compute the product $\mathbf{C}^T\mathbf{C}$; it is more efficient to compute the elements of the Toeplitz matrix using formula (2.103).

The following code can be used to compute the autocorrelation sequence of a real time series.

```
function [r0,r] = correlation(x);
%
% Function to compute the autocorrelation sequence
% of a real series
% IN  x: time series
% OUT r0: zero lag autocorrelation
%     r : vector containing autocorrelation samples
%         for lags k=1,2,3...nx-1
%
 r0 = sum(x.*x);
 nx = length(x);
 for k=1:nx-1;
  r(k) = 0;
   for j = 1:nx-k
    r(k) = r(k) + x(j) * x(j+k);
   end
  end
```

2.6 Inversion of non-minimum phase wavelets: optimun lag Spiking filters

Minimum phase wavelets are inverted using least-squares. The resulting filter is often called the Wiener filter or the spiking deconvolution operator. In general seismic wavelets are not minimum phase (some roots might lie inside the unit circle, they are mixed phase). An interesting feature of the the Least-squares approach is that the filter is also minimum phase.
If the wavele is not minimum phase, the actual output (the convolution of the filter with the wavelet) does not resembles the desire output.i The problem can be alleviated by defining an optimum lag Wiener filter. This is an inverse filter where the desired output is the following sequence:

$$(0,0,0,0\ldots,1,0,0,0,\ldots) \tag{2.105}$$

The filter design problem is equivalent to what has been already studied in section (2.3.5). However, now the right side term in equation (2.86) is a spike that has been **delayed** by an amount we called L (lag).
The optimum lag L_{opt} is given by the value L where the actual output resembles the desired output. It is clear that we need to define some measure that is capable of measuring how close the actual output is to the desired output. This is done by defining a filter performance norm

$$P = 1 - E \tag{2.106}$$

$$E = \frac{1}{r_0}||\mathbf{C}\,\hat{\mathbf{f}} - \mathbf{b}||^2 \tag{2.107}$$

where E is the normalized mean square error, r_0 is the zero lag autocorrelation coefficient. It can be shown that

$$0 \geq E \leq 1$$

when $E = 0$ we have a perfect filter (the desired and the actual output are equal). When $E = 1$ there is no agreement between the desired and the actual output. The filter performance on the other hand is maximized, $P = 1$, for an optimum filter. In practical applications we do a search for the value of L that maximizes the filter performance P, the value L where P is maximized is usually called the *optimum lag*.

Chapter 3

Discrete Fourier Transform

In this Chapter will present the transition from the Z transform to the DFT (Discrete Fourier Transform). The DFT is used to compute the Fourier transform of discrete data.

3.1 The Z transform and the DFT

We have already defined the Z-transform of a time series as follows:

$$X(z) = \sum_{n=0}^{N-1} x_n z^n \,. \tag{3.1}$$

The Z-transform provides a representation of our time series in terms of a polynomial. Let us introduce the following definition:

$$z = e^{-i\omega} \tag{3.2}$$

in this case the z-transform becomes, the DFT:

$$X(\omega) = \sum_{n=0}^{N-1} x_n e^{-i\omega n} \tag{3.3}$$

We have mapped the Z-transform into the unit circle ($z = e^{-i\omega}$ is a complex variable of unit magnitude and phase given by ω). The phase ω is also the frequency variable in units of radians. It is easy to make an analogy with the Fourier transform

$$F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-i\omega t} dt \tag{3.4}$$

In the last equation the frequency is given in $radian/sec$ when the time is measured in seconds. In the Fourier transform an analog signal is multiplied by the Fourier kernel $e^{-i\omega t}$ and integrated with respect to time. In the DFT the integration is replaced by summation and the Fourier Kernel by $e^{-i\omega n}$. Since n does not have units, ω has units of radians.
The DFT maps a discrete signal into the frequency domain. So far, the frequency ω is a continuous variable, but let us assume that one wants to discretized ω in the same way we have discretized the temporal variable t. The limits of ω are given by $[0, 2\pi)$, remember that ω is an angular frequency. If the time series is a N points time series, we can discretized the frequency axis as follows:

$$\omega_k = k\, 2\pi/N\,, \quad k = 0, 1, \ldots, N-1 \tag{3.5}$$

Now we can define the DFT as follows:

$$X(\omega_k) = \sum_{n=0}^{N-1} x(n) e^{-i\omega_k\, n}\,, k = 0, 1, \ldots, N-1\,. \tag{3.6}$$

Now the DFT is a transformation of a N points signal into N Fourier coefficients $X_k = X(\omega_k)$. We can also write down our transform in matrix form

$$\begin{pmatrix} X_0 \\ X_1 \\ X_2 \\ \vdots \\ X_{N-1} \end{pmatrix} = \begin{pmatrix} 1 & & 1 & \ldots & 1 \\ 1 & e^{-i2\pi/N} & e^{-i2\pi 2/N} & \ldots & e^{-i2\pi(N-1)/N} \\ 1 & e^{-i2\pi 2/N} & e^{-i2\pi 4/N} & \ldots & e^{-i2\pi 2(N-1)/N} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & e^{-i2\pi(N-1)/N} & e^{-i2\pi 2(N-1)/N} & \ldots & e^{-i2\pi(N-1)(N-1)/N} \end{pmatrix} \cdot \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_{N-1} \end{pmatrix} \tag{3.7}$$

The last equation can be written in compact form as follows:

$$\mathbf{X} = \mathbf{F} \,.\, \mathbf{x} \tag{3.8}$$

It is clear that the DFT can be interpreted as a matrix that maps an M-dimensional vector into another M-dimensional vector. The remaining problem entails the invertivility of the DFT. We need a transform to come back from the frequency domain to the time domain, in other words we need $\mathbf{F}^{-1}$.

3.1.1 Inverse DFT

We propose the following inverse

$$x_n = \sum_{l=0}^{N-1} \alpha_l e^{i2\pi ln/N} \,. \tag{3.9}$$

where the coefficients α must be determined. This formula is analogous to the one use to invert the Fourier transform, however it is important to note that because the discrete nature of the problem we have interchanged the integration symbol by a summation. The parameters α_k are our unknowns. In order to find our unknowns we proceed as follows, first we replace the last equation into equation (3.6),

$$X_k = \sum_{n=0}^{N-1} \sum_{l=0}^{N-1} \alpha_l e^{i2\pi n(l-k)/N} \,. \tag{3.10}$$

The last equation can be rewritten as

$$X_k = \sum_{l=0}^{N-1} \alpha_l \sum_{n=0}^{N-1} e^{i2\pi n(l-k)/N} = \sum_{l=0}^{N-1} \alpha_l s_{l-k} \,, \tag{3.11}$$

where the sequence s_{l-k} is given by

$$s_{l-k} = \sum_{n=0}^{N-1} e^{i2\pi n(l-k)/N} \,. \tag{3.12}$$

At this point we realize the the last equation is a geometric series[1] with a sum given by

$$\sum_{n=0}^{N-1} u^n = \begin{cases} N & \text{if} \quad u = 1 \\ \frac{u^N}{1-u} & \text{if} \quad u \neq 1 \end{cases} \tag{3.13}$$

In equation (3.12) we can identify $u = e^{i2\pi n(l-k)/N}$, therefore

$$s_{l-k} = \begin{cases} N & \text{if} \quad l = k \\ 0 & \text{if} \quad l \neq k \end{cases} , \tag{3.14}$$

after introducing the final result into equation (3.11) we obtain the following expression for our unknown coefficients α_k

$$X_k = N\alpha_k, \quad k = 0, \dots, N-1\,. \tag{3.15}$$

our inversion formula becomes:

$$x_n = \frac{1}{N} \sum_{l=0}^{N-1} X_l e^{i2\pi ln/N}\,. \tag{3.16}$$

This equation can also be written as follows:

$$\mathbf{x} = \frac{1}{N} \mathbf{F}^H \mathbf{X}\,. \tag{3.17}$$

The matrix $\mathbf{F}^H$ is the Hermitian transpose of the matrix $\mathbf{F}$. It is clear that the $N \times N$ matrix $\mathbf{F}$ is an orthogonal matrix,

$$\mathbf{F}^H\,\mathbf{F} = N\,\mathbf{I}_N\,, \tag{3.18}$$

where $\mathbf{I}_N$ is an $N \times N$ identity matrix. Finally we have a pair of transforms, the DFT and the IDFT (inverse DFT) , given by

[1] We have used a geometric series to find the inverse of a dipole in Chapter 2

$$X_k = \sum_{n=0}^{N-1} x_n e^{-i2\pi kn/N}\,, \quad k = 0, \dots, N-1\,, \tag{3.19}$$

$$x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k e^{i2\pi kn/N} \quad n = 0, \dots, N-1\,. \tag{3.20}$$

The DFT is used to map a discrete signal into the frequency domain, the IDFT is used to map a signal in the frequency domain into time domain. Because, the DFT is an orthogonal transformation, the inverse is computed using the Hermitian operator .
The cost of inverting an $N \times N$ matrix is proportional to N^3, the cost of multiplying a matrix by a vector is proportional to N^2. We will further diminish the computation cost of multiplying a matrix times a vector by using the FFT (Fast Fourier Transform).

3.1.2 Zero padding

The DFT allows us to transform an N-points time series into N frequency coefficients X_k, where the index k is associated to the discrete frequency ω_k,

$$\omega_k = \frac{2\pi k}{N} = \Delta\omega k\,, k = 0, 1, \dots, N-1$$

the frequency axis is sampled every $\Delta\omega$ radians. At this point it appears that $\Delta\omega$ is controlled by the number of samples of the time series N. Zero padding can be used to decrease the frequency interval $\Delta\omega$, in this case we define a new time series that consists of the original time series followed by $M - N$ zeros,

$$x = [x_0, x_1, x_2, \dots, x_{N-1}\, \underbrace{0, 0, \dots, 0}_{M-N}]$$

The new time series is an M-points time series with a DFT given by

$$X_k = \sum_{n=0}^{N-1} x_n e^{-i2\pi nk/M} = \sum_{n=0}^{M-1} x_n e^{-i2\pi nk/M}\,, \quad k = 0, \dots, M-1 \tag{3.21}$$

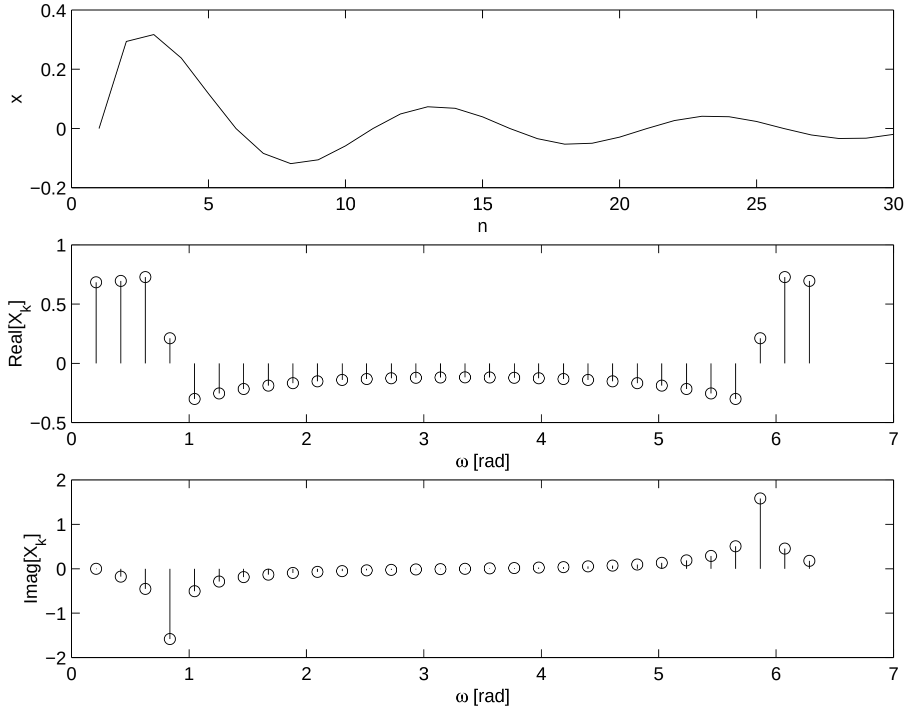


Figure 3.1: A time series and the real and imaginary parts of the DFT. Note that freq. axis is given in radians $(0, 2\pi)$

The sampling interval of the frequency axis is now

$$\Delta\omega = \frac{2\pi}{M} < \frac{2\pi}{N} \,.$$

In general, the trick of zero padding is used to oversample the frequency axis at the time of plotting the DFT. It is also important to pad with zeros at the time of performing discrete convolution using the DFT.

In Figures (3.1) and (3.2) we portray the effect of padding a time series. In Figure (3.1) we have the original time series and the associated DFT (the real and imaginary part). In Figure (3.2) the original time series after zero padding (20 zeros) is used to compute the DFT.

In the following example I show how to pad with zeros a time series. This codes was utilized to generate Figures (3.1) and (3.2).

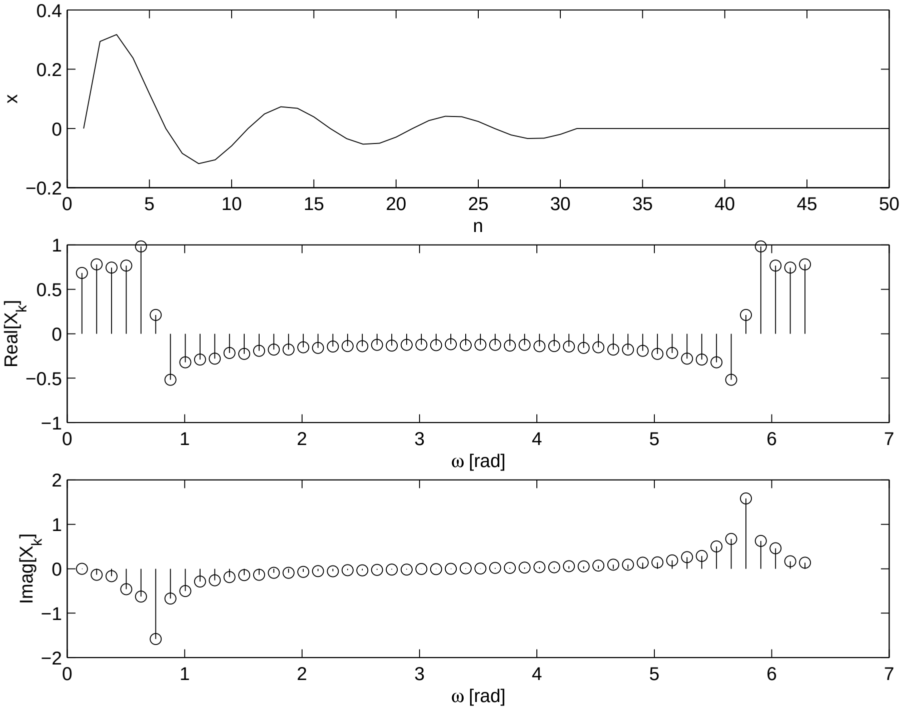


Figure 3.2: A time series and the real and imaginary parts of the DFT. In this case the time series was padded with zeros in order to decrease the frequency interval $\Delta\omega$.

```
% Zero padding - Example
%
N = 30;                          % Length of the TS
L = 20;                          % Number of zeros to pad
n = 1:1:N;                       % Prepare a TS.
x = sin(2.*pi*(n-1)*0.1);
x = x./n;
if L>=1; x = [x, zeros(1,L)];  % Pad with zeros if L>0
N = length(x);
n = 1:1:N;
end;
```

```
X = fft(x);                    % Compute the DFT
w = 2*pi*n/N;                  % Compute the freq. axis in rads.
subplot(311);                  % Plot results
plot(n,x); xlabel('n'); ylabel('x');
subplot(312);
stem(w,real(X)); xlabel('\omega [rad]');ylabel('Real[X_k]')
subplot(313);
stem(w,imag(X)); xlabel('\omega [rad]');ylabel('Imag[X_k]')
```

3.1.3 The Fast Fourier Transform (FFT)

The FFT is not a new transform; the FFT is just a fast algorithm to compute DFTs. The FFT is based on the halving trick, that is a trick to compute the DFT of length N time series using the DFT of two sub-series of length N/2. Let's start assuming that we a have a time series of length $2N$:

$$z_0, z_1, z_2, z_3, \dots, z_{2N-1} \,.$$

First, we will assume that one wants to compute the DFT of the time series z. Using the definition

$$Z_k = \sum_{n=0}^{2N-1} z_n \, e^{-i2\pi nk/(2N)}, \;\; k = 0 : 2N-1 \,, \tag{3.22}$$

we can rewrite the last equation in terms of two time series composed of even samples $x = z_0, z_2, z_4 \dots$ and odd samples $y = z_1, z_3, z_5 \dots$, respectively.

$$Z_k = \sum_{n=0}^{N-1} z_{2n} \, e^{-i2\pi 2nk/(2N)} + \sum_{n=0}^{N-1} z_{2n+1} \, e^{-i2\pi(2n+1)k/(2N)} \,. \tag{3.23}$$

It is clear that the RHS term can be written in terms of the DFTs of x (even samples) and y (odd samples)

$$Z_k = X_k + e^{-i2\pi k/(2N)} \, Y_K, \;\; k = 0 : N-1 \,. \tag{3.24}$$

The last equation provides a formula to compute the first N samples of the DFT of z based on the N samples of the DFT of x and z. Now, note that we need another formula to retrieve the second half of the samples of the DFT of z,

$$Z_k = \sum_{n=0}^{2N-1} z_n\, e^{-i2\pi nk/(2N)}\,, k = N, \ldots, 2N-1\,. \tag{3.25}$$

In the last equation we apply the following substitution: $j = k - N$

$$Z_{j+N} = \sum_{n=0}^{2N-1} z_n\, e^{-i2\pi n(j+N)/(2N)}\,, k = N, \ldots, 2N-1\,. \tag{3.26}$$

After rewriting the last expression in terms of x and y we end up with the following formula:

$$Z_{j+N} = X_j - e^{-i2\pi k/(2N)}\, Y_j\,,\;\; j = 0, N-1\,. \tag{3.27}$$

Now we have two expression to compute the DFT of a series of length $2N$ as a function of two time series of length N. Good FFT algorithms repeat this trick until the final time series are series of length 1. The recursions given in (3.24) and (3.27) are applied to recover the DFT of the original time series. It can be proved that the total number of operations of the FFT is proportional to $N \ln_2(N)$ (for a time series of length N). This is an important saving with respect to the standard DFT which involves a number of operations proportional to N^2.

A small modification to formulas (3.24) and (3.27) will permit us to compute the inverse DFT.

3.1.4 Working with the DFT/FFT

Symmetries

Let us start with a the DFT of a **real** time series of length N

$$X_k = \sum_{n=0}^{N-1} x_n e^{-i2\pi nk/N}\,,\;\; k = 0, \ldots, N-1 \tag{3.28}$$

the frequency in radians is given by

$$\omega_k = 2\pi k/N\,, k = 0, 1, \ldots, N-1\,.$$

Using the following property

$$e^{i2\pi(N-k)n/N} = e^{-i2\pi kn/N} \tag{3.29}$$

we can re-write equation (3.28) as follows:

$$X_{N-k} = \sum_{n=0}^{N-1} x_n e^{-i2\pi n(N-k)/N} = \sum_{n=0}^{N-1} x_n e^{i2\pi n(N+k)/N} \; = \; X_k^* \, . \tag{3.30}$$

The following example is used to illustrate the last point. The time series is $x = [2, 3, 1, 3, 4, 5, -1, 2]$ The DFT is given by

```
Sample k          X_k                  N-k (N=8)

  0         19.0000                      8
  1         -4.1213 - 1.2929i            7
  2          6.0000 - 3.0000i            6
  3          0.1213 + 2.7071i            5
  4         -7.0000                      4
  5          0.1213 - 2.7071i            3
  6          6.0000 + 3.0000i            2
  7         -4.1213 + 1.2929i            1
```

It is claear that the first $N/2+1$ samples are required to define the remaining samples of the DFT.

The frequency axis

In the previous example I compute the DFT, X_k in terms of samples k. We have already mentioned that k is related to angular frequency as follows: $\omega_k = 2\pi k/N$. Let us define the sampling interval in frequency as $\Delta\omega = 2\pi/N$, therefore, $\omega_k = \Delta\omega\, k,\ k = 0, \ldots, N-1$. In the previous example we have

```
k    omega_k                    X_k
0         0             19.0000
```

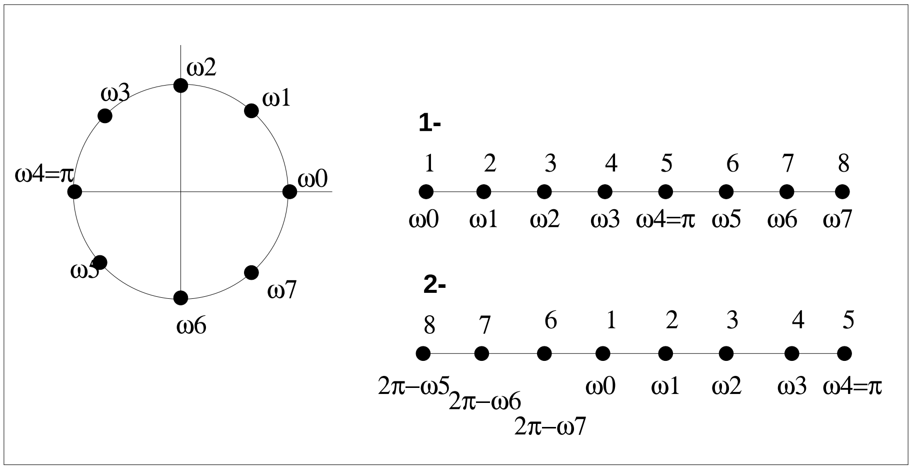


Figure 3.3: Distribution of frequencies around the unit circle. The DFT can be plotted as in the $[0, 2\pi)$ interval or in the $(-\pi, \pi]$ interval.

```
1    0.7854            -4.1213 - 1.2929i
2    1.5708             6.0000 - 3.0000i
3    2.3562             0.1213 + 2.7071i
4    3.1416            -7.0000
5    3.9270             0.1213 - 2.7071i
6    4.7124             6.0000 + 3.0000i
7    5.4978            -4.1213 + 1.2929i
```

Note that the central frequency is $\omega_4 = \pi$, the last frequency is almost 2π or $\omega_7 = 2\pi - \Delta\omega$. This is because we have discretized the unit circle in the interval $[0, 2\pi)$.

In does not make much sense to talk about frequencies above π radians. If fact the $\omega = \pi$ is the Nyquist frequency in rads. What is the meaning of frequencies above $\omega > \pi$?. Well this simple reflect the way we have discretized the unit circle when computing the DFT.

3.2 The 2D DFT

The 2D Fourier transform is defined as follows:

$$F(\omega_1,\omega_2) = \int\int f(x_1,x_2)e^{-i(\omega_1\,x_1+\omega_2\,x_2)}dx_1\,dx_2\,, \tag{3.31}$$

similarly, we can define the inversion formula

$$f(x_1,x_2) = \int\int F(\omega_1,\omega_2)e^{i(\omega_1\,x_1+\omega_2\,x_2)}d\omega_1\,d\omega_2\,. \tag{3.32}$$

Whereas the 1D FT is used to decomposed signals in a decomposition of sin and cos, one can image the 2D FT as a decomposition of a signal in terms of plane waves.
It is important to stress that for our signal processing applications we will be dealing with the 2D DFT (this is the discrete version of the FT).
Let us first consider a 2D discrete signal (i.e., a map)

$$x_{m,n},\ n=0,\dots,N-1,\ m=0,\dots,M-1\,.$$

The formulas for the forward and inverse DFT in the 2D case are given by

$$X_{k,l} = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} x_{m,n}e^{-i2\pi km/M}e^{-i2\pi ln/N}\ ,\ k=0,\dots,M,\ l=0,\dots,N\,. \tag{3.33}$$

$$x_{k,l} = \frac{1}{N\,M}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} X_{m,n}e^{i2\pi km/M}e^{i2\pi ln/N}\ ,\ k=0,\dots,M,\ l=0,\dots,N\,. \tag{3.34}$$

The 2D DFT is computed by calling two times the 1D DFT. This is very simple: you first compute the DFT of all the columns of $x_{n,m}$, then you compute the DFT to rows of the previous result. In fact, 2D DFT codes are just 1D FFT's codes working on rows and columns. The 2D DFT is important at the time of filtering 2D images (i.e., gravity maps, seismic records). Notice that in the 2D DFT we need to consider 2D symmetries.

3.3 On the Design of Finite Impulse Response filters

So far we have studied operators (filters) that are capable of collapsing a wavelet into a spike. These filters are often called spiking filters or Wiener filters. In this section we will examine the problem of designing FIR (Finite Impulse Response) filters. These are filters that are used to eliminate undesired spectral components from our data.

3.3.1 Low Pass FIR filters

In this case we want to design a filter that operates in the time domain with a amplitude spectrum with the following characteristics:

$$B(\omega) = \begin{cases} 1 & -\omega_c \le \omega \le \omega_c \\ 0 & otherwise \end{cases} \tag{3.35}$$

We will assume that the filter phase is zero. In the previous expression ω_c is the cut-off frequency. This filter can be either applied in the frequency domain or in the time domain. It is clear that if the signal to be filtered is called $X(\omega)$, then the filtered signal is given by

$$Y(\omega) = X(\omega)\,.\,F(\omega) \tag{3.36}$$

In general, it is more convenient to design *short* filters in the time domain and applied them via convolution[2]

$$y(t) = x(t) *, b(t) \tag{3.37}$$

where the sequence b_k is the Impulse Response of the filter with desired amplitude response $B(\omega)$. We can use the inverse Fourier transform to find an expression for $b(t)$,

[2] note that we are working with continuous signals.

$$b(t) = \frac{1}{2\pi} \int_{-\omega_c}^{\omega_c} B(\omega) e^{i\omega t} \, d\omega \tag{3.38}$$

Evaluating the last integral leads to the following expression for the filter $f(t)$:

$$b(t) = \frac{\omega_c}{\pi} \frac{sin(\omega_c \, t)}{\omega_c \, t} = \frac{\omega_c}{\pi} \, sinc(\omega_c \, t) \, , \qquad \infty < t < \infty \, . \tag{3.39}$$

This is the impulse response of the continues system with amplitude response $B(\omega)$. We need to discretized the previous expression to obtain the impulse response of a discrete system:

$$b_n = \Delta t \, b(t)|_{t=n \, \Delta t} \tag{3.40}$$

the factor Δt comes from equation (1.44); this is a scaling factor that allows us to say that the Fourier transform of the discrete and continuous signals are equal in $[-\pi/\Delta t, \pi/\Delta t]$.
The final expression of the digital filter is given by

$$b_n = \Delta t \frac{\omega_c}{\pi} \, sinc(\omega_c \, n \Delta t), \;\; n = \ldots, -3, -2, -1, 0, 1, 2, 3 \ldots \, . \tag{3.41}$$

It is clear that this is a IIR filter (infinite impulse response filter). A FIR filter is obtained by truncating the IIR filter:

$$b_n = \Delta t \frac{\omega_c}{\pi} \, sinc(\omega_c \, n \delta t), \;\; n = -L \ldots, -3, -2, -1, 0, 1, 2, 3 \ldots L \, . \tag{3.42}$$

In this case we have a filter of length $2\,L+1$. When the filter is truncated the actual amplitude spectrum of the filter is not equal to the desired or ideal amplitude spectrum (3.35). This point has already been studied in section (1.2.3) where we examined the spectral artifacts that are introduced when a signal is truncated in time. In Figure (3.4) we display the impulse response of a filter of cut-off frequency $f_c = 50Hz$ for filter lengths $(2L+1)$ 21, and

41. We also display the associated amplitude response. It is easy to see that the filter truncation has introduced the so called *Gibbs phenomenon* (Oscillations).
One way to minimize truncation artifacts is by smoothing the truncated impulse response with a taper or window.

$$b_n^w = b_n.w_n$$

now b_n^w is the truncated impulse response after applying a taper function. The taper is used to minimize truncation effects at the end point of the impulse response; a popular taper is the *Hamming Window*:

$$w_n = 0.54 - 0.45\, cos(2\pi(n-1)/(N-1))\,, n = 1:N$$

In figure (3.5) we analyze the effect of tapering the impulse response of the filter before computing the amplitude response. It is clear that the oscillations around the transition band have been eliminated. It is important to stress that tapering will also increase the width of the transition band; therefore filters that are too short might not quite reflect the characteristics of the desired amplitude response.

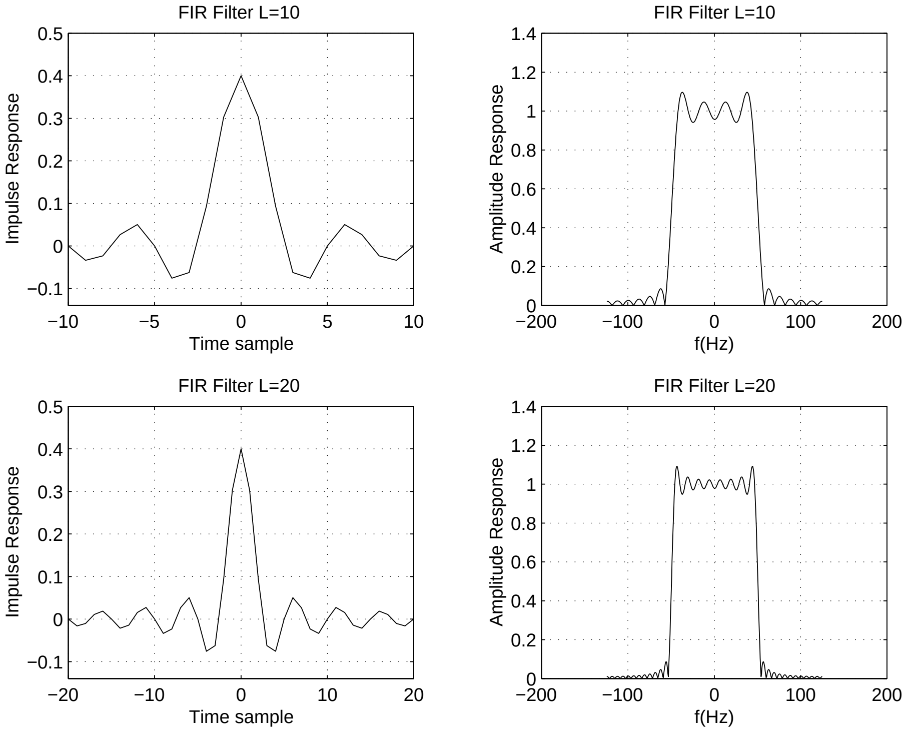


Figure 3.4: Impulse response of two finite length filters and the associated amplitude response. The filter were obtained by truncating the *ideal* infinite length impulse response sequence.

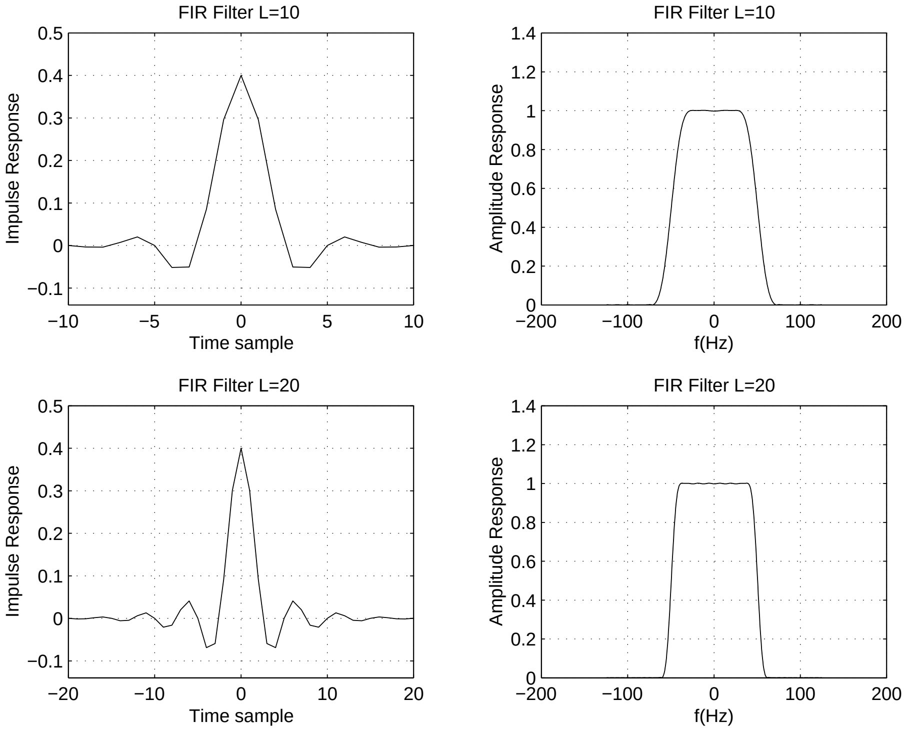


Figure 3.5: Impulse response of two finite length filters and the associated amplitude response. The filter were obtained by truncating the *ideal* infinite length impulse response sequence. In this case the truncated impulse response was *taper* with a Hamming window. Tapering helps to attenuate side-lobe artifacts (Gibbs phenomenon)

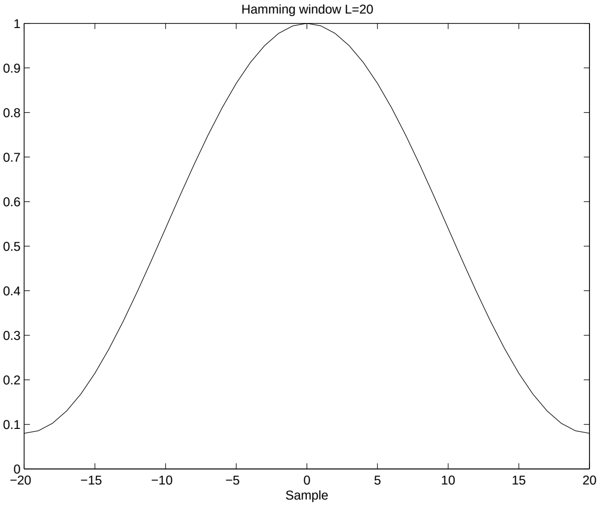


Figure 3.6: A Hamming taper (window) of length $2L + 1$.

3.3.2 High Pass filters

Knowing how to compute low pass filters allows us to compute high pass filters. If the amplitude response of a low pass filter if given by $B^L(\omega)$ we can contrstruct a high pass filter with the same cut-off frequency using the following expression:

$$B^H(\omega) = 1 - B^L(\omega) \tag{3.43}$$

that suggests that one can compute the impulse response of the low pass filter an then transform it into a high pass filter using the following expression:

$$\begin{array}{rcll} b_k^H & = & -b_k^L & k \neq 0 \\ b_k^H & = & 1 - b_k^L & k = 0 \end{array} \tag{3.44}$$

Chapter 4

Deconvolution of reflectivity series

4.1 Modeling normal incidence seismograms

In this chapter we will study the problem of computing reflection and transmition coefficients for a layered media when plane waves propagate in the earth with angle of incidence $i = 0$ (Normal incidence). We will use these concepts to derive a model for normal incidence seismograms.

4.1.1 Normal incidence

Consider a plane wave impinging at angle of propagation $i = 0$ with respect to the normal (see Figure (4.1)). In this case we have three waves:

- Incident wave: $\downarrow$ in medium 1
- Reflected wave: $\uparrow$ in medium 1
- Transmitted wave: $\downarrow$ in medium 2

Let us assume that the amount of incident wave is equal to 1, the amount of reflected wave is given by r, and the amount of transmitted wave is denoted by t. At the boundary the following condition should be satisfied (continuity of displacements)

$$1 + r = t$$

This equation has two unknowns, to compute the r and t we need an extra equation. We well consider conservation of energy. In the acoustic (vertical incidence case) conservation of energy leads to the following equation:

$$I_1 \times 1 + I_1 \times r^2 \;=\; I_2 \times t^2\,.$$

The quantities I_1 and I_2 are called **acoustic impedances**

$$I_1 = \rho_1\, v_1$$

$$I_2 = \rho_2\, v_2$$

where ρ_1 and ρ_2 are the densities of the material above and below the interface and v_1 and v_2 the P-velocities, respectively. After combining the equations of continuity of displacement and conservation of energy we obain the following expressions

$$r \;=\; \frac{I_2 - I_1}{I_2 + I_1}\text{Reflection coefficient} \tag{4.1}$$

$$t \;=\; \frac{2 I_1}{I_2 + I_1}\text{Transmition coefficienct} \tag{4.2}$$

The above analysis is valid for an incident plane wave propagating downwards. Let's consider the case of an incident wave propagating upwards (Figure (4.2)).

- Incident wave: ↑ in medium 2
- Reflected wave: ↓ in medium 2
- Transmitted wave: ↑ in medium 1

In this case the reflection and transmition coefficients are given by

$$r' \;=\; \frac{I_1 - I_2}{I_2 + I_1} \tag{4.3}$$

$$t' \;=\; \frac{2 I_2}{I_2 + I_1} \tag{4.4}$$

From the above equations it is clear that

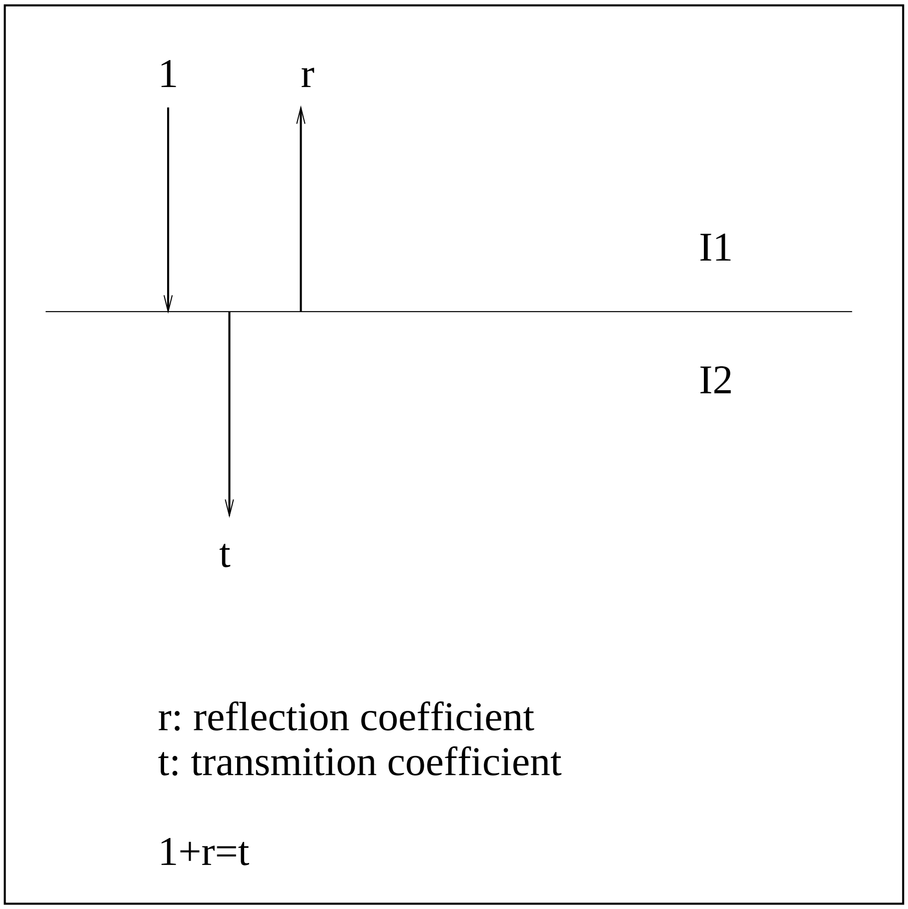


Figure 4.1: P-wave normal incidence. The incident wave is propagating downwards.

$$r' = -r \tag{4.5}$$

4.1.2 Impulse response

Let's assume that we run a zero offset experiment in a stratified earth composed of 4 layers plus a half-space of impedances given by I_1, I_2, I_3, I_4, I_5. (Figure (4.3)). At $t = 0$ a delta-like source emits energy into Earth. The energy is transmitted and reflected from the layers. If we do not consider

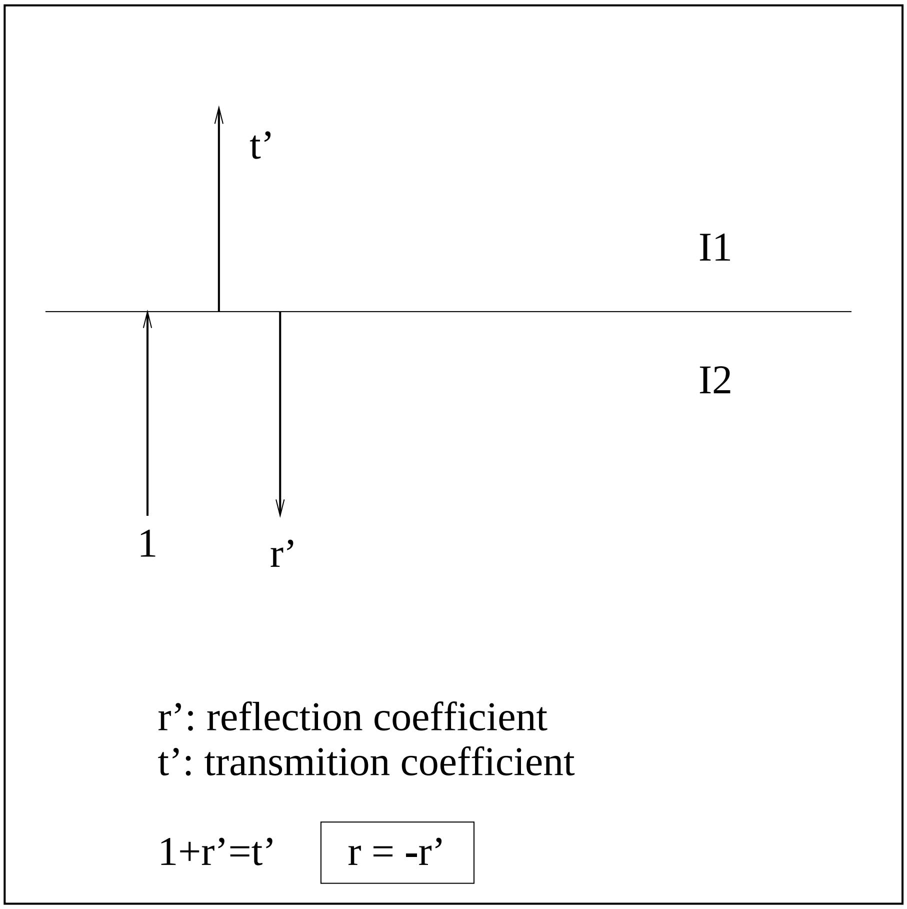


Figure 4.2: P-wave normal incidence. The incident wave is propagating upwards.

multiples reflection, our sismogram will be composed of 4 arrivlas (4 reflections).

To simplify the problem I will show how to compute the amplitude of the wave recorded at the surface of the earth generated (reflected) at the interface 4. First we have to compute the amount of amplitude transmitted to each one of the layers until reaching layer 4. This is given by the product of the transmition coefficients of each layer. In Figure (4.3) the transmition coefficients t are replaces by their equivalent expression $(1 + r)$.

The amplitude of the wave when reaches the layer 4 is

$$1 \times t_1 \times t_2 \times t_3 = (1+r_1)(1+r_2)(1+r3)$$

when the wave is reflected in the layer 4 the total amplitude at that point (last expression) needs to be multiplied by the reflection coefficient of interface 4,

$$1 \times t_1 \times t_2 \times t_3 \times r_4 = (1+r_1)(1+r_2)(1+r3)t_4$$

Note that now the wave (reflected wave) is propagating upwards, therefore, the transmition coefficients are given by terms of the form

$$1 + r' = 1 - r$$

The final amplitude after propagating the wave to the surface of the Earth (this is what the receiver is measuring!) is given by

$$\underbrace{(1+r_1)(1+r_2)(1+r_3)}_{\text{Transmition } \downarrow} \times \underbrace{r_4}_{\text{Reflection}} \times \underbrace{(1-r_1)(1-r_2)(1-r_3)}_{\text{Transmition } \uparrow}$$

The final expression for the amplitude of the wave reflected in the interface 4 can be written down as follows

$$(1-r_1^2)(1-r_2^2)(1-r_3^2)r_4\,.$$

It is clear that reflections occur at all the layers:

Amplitude of the reflection generated at intergace 1

$$A_1 \;=\; r_1$$

Amplitude of the reflection generated at intergace 2

$$A_2 \;=\; (1-r_1^2)r_2$$

Amplitude of the reflection generated at intergace 3

$$A_3 \;=\; (1-r_1^2)(1-r_2^2)r_3$$

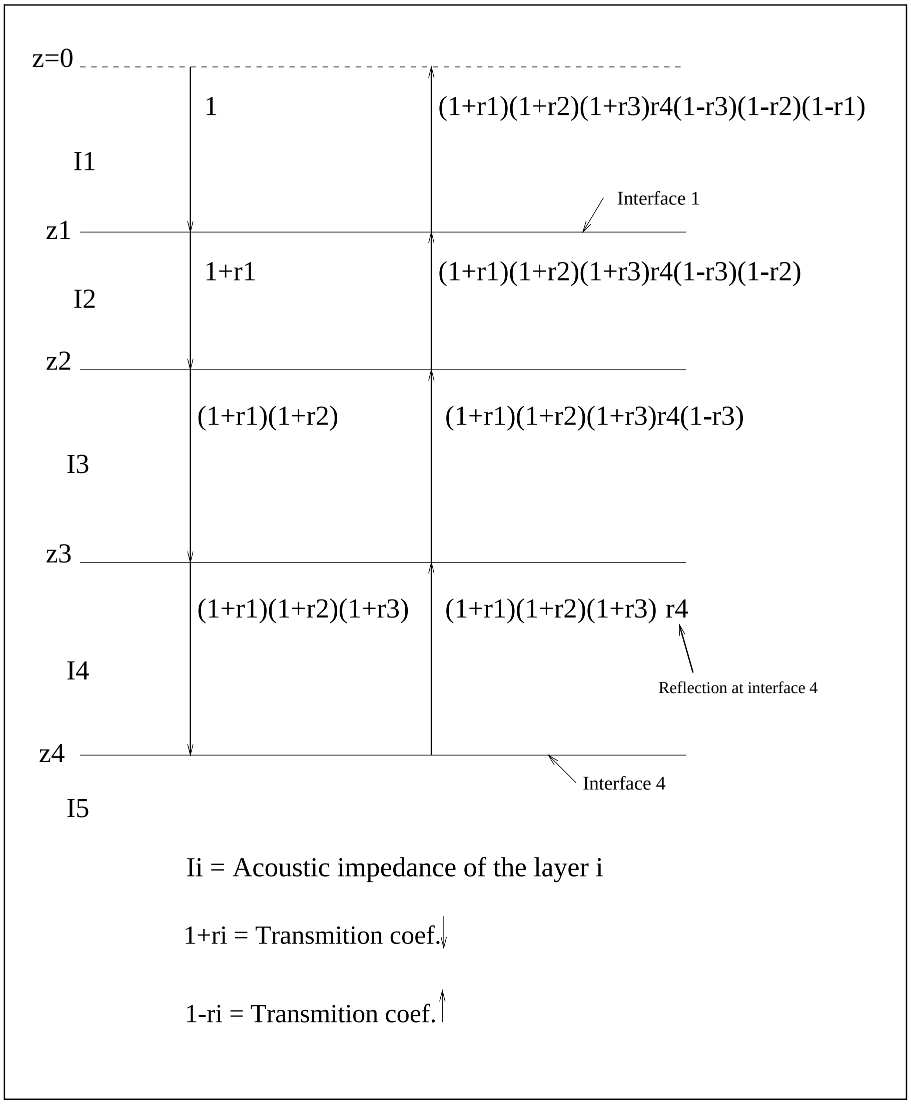

z=0
1
(1+r1)(1+r2)(1+r3)r4(1-r3)(1-r2)(1-r1)
I1
Interface 1
z1
1+r1
(1+r1)(1+r2)(1+r3)r4(1-r3)(1-r2)
I2
z2
(1+r1)(1+r2)
(1+r1)(1+r2)(1+r3)r4(1-r3)
I3
z3
(1+r1)(1+r2)(1+r3)
(1+r1)(1+r2)(1+r3) r4
Reflection at interface 4
I4
z4
Interface 4
I5
Ii = Acoustic impedance of the layer i
1+ri = Transmition coef.↓
1-ri = Transmition coef.↑

Figure 4.3: Amplitude of a wave plane wave propagating in a layered medium. Analysis of the wave reflected in the interface 4.

Amplitude of the reflection generated at intergace 4

$$A_4 \; = \; (1 - r_1^2)(1 - r_2^2)(1 - r_3^2) r_4$$

We can write a general expression for the amplitude of a reflection generated at the k-th interface:

$$A_1 = r_1$$

$$A_k = \prod_{i=1}^{k-1} (1 - r_i^2)\, r_k \;\; k = 2, 3, 4, \dots$$

How to interpret these results?. If we assume that the earth is excited with a delta function, and neglecting the presence of multiples, our zero-offset seismogram will be a collection of delta functions (spikes) at arrival times given by the two-way travel time formula. The strength of each arrival will be proportional to the amplitude A_k

However, in real exploration seismology, it is impossible to have a source that resembles a delta function. The source signature is called a wavelet. This is a finite length time function that we will denote as $w(t)$. In this case, the seismogram is represented by a superposition of wavelets arriving at different times and with amplitude proportional to A_k.

In our model with 4 interfaces (Figure (4.3)) we will have 4 arrivals of amplitude A_1, A_2, A_3 and A_4. The seismogram can be expressed as follows

$$s(t) = A_1\, w(t - t_1) + A_2\, w(t - t_2) + A_3\, w(t - t_3) + A_4\, w(t - t_4) \qquad (4.6)$$

where t_1, t_2, t_3 and t_4 are the arrival times of each reflection [1]

Notice that if we neglect transmission effects, the amplitude A_i can be replaced by the reflection coefficient r_i. In general we will assume an Earth model that conists of microlayers In this case we can write the seismic trace model a convolution between two time series: a wavelet and the reflectivity sequence

$$s_n = w_n * q_n \,. \qquad (4.7)$$

[1] Notice that $w(t - \tau)$ is $w(t)$ after being delayed τ seconds.

4.2 Deconvolution of reflectivity series

So far we have discuss the problem of designing a deconvolution operator for a seismic wavelet. We have also examined a "toy" example, the minimum phase dipole (Chapter 2).

In general, the convolutional model is a very well accepted model to describe a seismic trace. In this model we say that the seismic trace (in general a zero-offset trace) can be written down as a convolution of two signals: a seismic wavelet (this is the source function) and the reflectivity series.

The reflectivity series is our "geological" unknown. In fact, the reflectivity is a sequence of spikes (reflectors) that indicates the position (in time) of layers in the subsurface, the strength or amplitude of each spike is an indicator of how much energy is reflected back to the receivers during the seismic experiment. Let's write the seismogram as a simple convolution between a wavelet w_n and a reflectivity sequence q_n:

$$s_n = w_n * q_n \,. \tag{4.8}$$

In this simple model we have neglected the noise, in general we will assume that deterministic noise (multiples and ground roll) has been attenuated and therefore what is left is random noise

$$s_n = w_n * q_n + n_n \tag{4.9}$$

It is clear from the above equation that one has a problem with one equation (one observable) and two unknowns (the wavelet and the reflectivity). Therefore, the seismic deconvolution problem involves the solution of two subproblems:

- **Wavelet Estimation**
- **Operator design**

By wavelet estimation we refer to methods to estimate the seismic source from the seismic trace. In general these methods are statistical techniques that explode some properties of the remaining unknown (the reflectivity). We do have deterministic techniques based on the wave equation to estimate the seismic source in the marine case. These methods are beyond the scope of this course.

4.2.1 The autocorrelation sequence and the white reflectivity assumption

We have seen that the design of a Wiener filter involves the inversion of an autocorrelation matrix with Toeplitz structure. This matrix arises from the fact that we have represented our convolution model as a matrix times vector multiplication. To clarify the problem, let us assume that we have a 3 point wavelet and we compute the autocorrelation matrix. We first write down the convolution matrix[2]:

$$\mathbf{C} = \begin{pmatrix} w_0 & 0 \\ w_1 & w_0 \\ w_2 & w_1 \\ 0 & w_2 \end{pmatrix} . \tag{4.10}$$

The autocorrelation matrix is given by

$$\mathbf{R} = \mathbf{C^T C} = \begin{pmatrix} r_0 & r_1 \\ r_1 & r_0 \end{pmatrix} . \tag{4.11}$$

Now we can try to write the autocorrelation coefficients in terms of the sample of the wavelet w_n, in this case we have:

$$r_0^w = w_0^2 + w_1^2 + w_2^2 \tag{4.12}$$

$$r_1^w = w_0 w_1 + w_1 w_2 \tag{4.13}$$

[2] This is the matrix you would have used to design a 2 points spiking or Wiener filter

The first coefficient is the zero-lag correlation coefficient, this is also a measure of the energy of the wavelet. The second coefficient [3] r_1^w is the first lag of the correlation sequence.
The correlation coefficients can be written using the following expression:

$$r_j^w = \sum_k w_k w_{k+j}\,, \quad j = 0, \pm 1, \pm 2 \ldots\,. \tag{4.14}$$

In the Wiener filter the matrix $\mathbf{R}$ is an $N \times N$ matrix where N is the length of the filter, in this case we will need to compute N autocorrelation coefficients:

$$r_j^w, j = 0, N-1\,.$$

In order to design a Wiener or spiking filter, we first need to know the wavelet. Unfortunately, the seismic wavelet is unknown. To solve this problem we use the white reflectivity assumption. Under this assumption the seismic reflectivity (the "geology") is considered a zero mean white process (Robinson and Treitel, 1980).
A zero-mean white process is an uncorrelated process, in other words if r_j^q is the autocorrelation function of the reflectivity, then

$$r_j^q = \left\{ \begin{array}{cc} P_q & j = 0 \\ 0 & j = \pm 1, \pm 2, \pm 3, \ldots \end{array} \right. . \tag{4.15}$$

The autocorrelation is a measure of similarity of a time series with itself. The zero lag coefficient measure the power of the signal (P_q), the first coefficient ($j = 1$) measures the similarity of the signal with a one-sample shifted version of itself.
If the reflectivity is a zero-mean white noise process, the following remarkable property is true:

$$r_j^s = P_q \; r_j^w\,. \tag{4.16}$$

[3] Please, note that the supra-script w is used to stress that this is the autocorrelation of the wavelet

In other words: *the autocorrelation function of the trace is an estimate (within a scale factor) of the autocorrelation of the wavelet.* It is clear that now we can estimate the autocorrelation of the wavelet from the autocorrelation of our observable: the seismic trace.
We have managed to compute the autocorrelation function of the wavelet, but what about the wavelet. It turns out that the Z-transform of the autocorrelation sequence of the wavelet can be used to compute the seismic wavelet. In this case we need to make a new assumption, we will assume that the wavelet is a minimum phase wavelet. In general, this is a good assumption to deal with sources generated by explosions (dynamite).
It is esay to show that the z transform of the autocorrelation sequence can be decomposed as follows:

$$R^w(z) = \sum_j r^w_j z^j = W(z)\, W(z^-1) \tag{4.17}$$

(this is for a real wavelet). In this case, the autocorrelation function provides information about the wavelet but cannot define the phase of the wavelet. After factorizing the above equation, one can select the zeros that lie outside the unit circle (the minimum phase dipoles!!). In this way we can recover a wavelet with minimum phase features consistent with a spectrum given by $R^w(z)$.
The estimation of a minimum phase wavelet from the autocorrelation function is ofter refered as the spectral factorization problem. It can be solved using different techniques, in particilar the Hilbert Transform provides a fast algoritihm to compute the minimum phase wavelet.

4.2.2 What to do with the noise?

We start with our noisy seismogram:

$$s_n = w_n * q_n + n_n \tag{4.18}$$

The goal of the deconvolution process is to recover q_n from the data, s_n. In order to achieve this objective, a filter f_k must be computed such that

$f_k * w_k = \delta_k$. Generally, of course, we can only compute an estimate of the filter $\hat{f}_k$ and $\hat{f}_k * w_k = a_k$, where a_k is called the averaging function and resembles a delta function only in the ideal case. Applying $\hat{f}_k$ to both sides of equation (4.18), yields the estimated output of the deconvolution process

$$\begin{aligned} \hat{q}_k &= a_k * q_k + \hat{f}_k * n_k \\ &= q_k + (a_k - \delta_k) * q_k + \hat{f}_k * n_k \,. \end{aligned} \tag{4.19}$$

Since our main requirement is to estimate a reliable model $\hat{q}_t$ which is close to the true reflectivity, it is important to design a filter such that the error terms in equation (4.19) are as small as possible. Or in other words, one seeks a solution with the following properties:

$$a_k = w_k * f_k \approx \delta_k \,, \tag{4.20}$$

and

$$f_k * n_k \approx 0 \tag{4.21}$$

The last two expression can also be written in matrix form

$$\mathbf{C}_w \mathbf{f} \approx \mathbf{d} \tag{4.22}$$

and

$$\mathbf{C}_n \mathbf{f} \approx \mathbf{0} \tag{4.23}$$

where $\mathbf{C}_w$ and $\mathbf{C}_n$ denote the convolution matrices for the wavelet and the noise, respectively. Both equations are honored when we minimize the following objective function:

$$J = ||\mathbf{C}_w \mathbf{f} - \mathbf{d}||^2 + \beta ||\mathbf{C}_n \mathbf{f}||^2 \,, \tag{4.24}$$

where β is a tradeoff parameter. The second term in the last equation can be written as

$$||\mathbf{C}_n\mathbf{f}||^2 = \mathbf{f}^T\mathbf{C}_n^T\mathbf{C}_n\mathbf{f}\,, \tag{4.25}$$

where the matrix $\mathbf{C}_n^T\mathbf{C}_n$ is the noise autocorrelation matrix. If the noise is uncorrelated, we can replace $\mathbf{C}_n^T\mathbf{C}_n$ by its estimator

$$E[\mathbf{C}_n^T\mathbf{C}_n] = \sigma_n^2\mathbf{I}\,. \tag{4.26}$$

where σ_n^2 is an estimate of the variance of the noise. Now the objective function J is given by,

$$J = ||\mathbf{C}_w\mathbf{f} - \mathbf{d}||^2 + \mu||\mathbf{f}||^2\,, \tag{4.27}$$

where $\mu = \sigma_n^2 \times \beta$. This is the objective function used to design an inverse filter, and the solution given by

$$\mathbf{f} = (\mathbf{R}_w + \mu\mathbf{I})^{-1}\mathbf{C}_w^T\mathbf{d}\,. \tag{4.28}$$

In Figures (4.4), (4.5) and (4.6) we test the performance of the least squares inversion when dealing with noise free and noisy data. It is clear, that the pre-whitening parameter plays a key role in the deconvolution of noisy data.

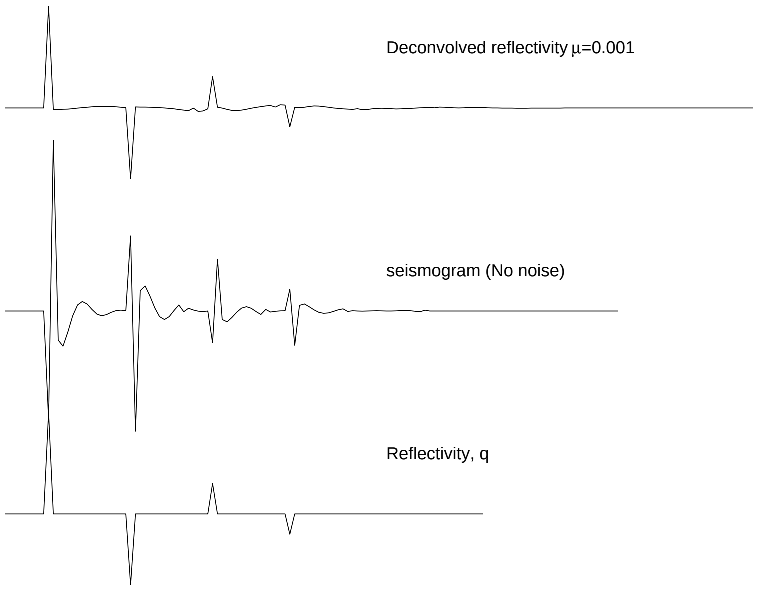


Figure 4.4: Deconvolution of a ”clean” seismogram.

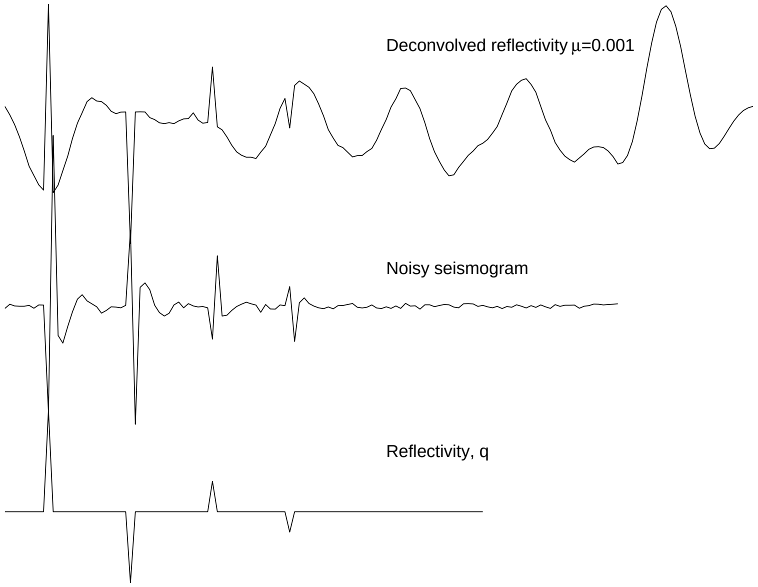


Figure 4.5: Deconvolution of a "noisy" seismogram. The tradeoff parameter is too small; the result is too unstable.

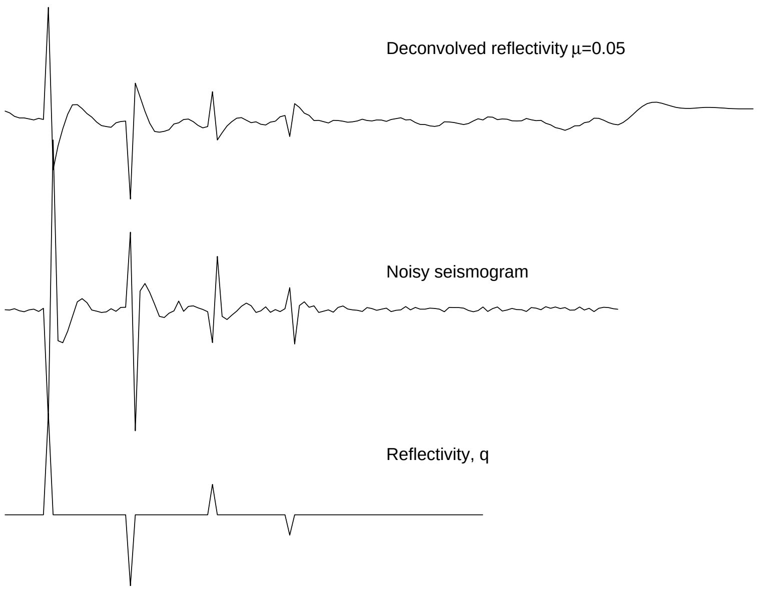


Figure 4.6: Deconvolution of a ”noisy” seismogram. The tradeoff has been increased to stabilize the solution

4.2.3 Deconvolution in the frequency domain

A procedure similar to the one outlined in the previous section can be used to deconvolved data in the frequency domain. Taking the discrete Fourier Transform of equation (4.19) yields

$$\hat{Q}_k = Q_k + (A_k - 1)Q_k + \hat{F}_k N_k. \tag{4.29}$$

Since a_k should be a good approximation to a delta function, it turns out that the filter should be designed satisfying the following requirement

$$W_K F_k = A_k \approx 1 \qquad \forall k. \tag{4.30}$$

Furthermore, in order to maintain the noise at a small level

$$F_k\, N_k \approx 0 \qquad \forall k. \tag{4.31}$$

We can combine these two requirements into a single one. In order to achieve this, let us construct the following objective or cost function

$$J = \sum_k |A_k - 1|^2 + \alpha \sum_k |F_k N_k|^2. \tag{4.32}$$

Minimizing the objective function with respect to the filter coefficients leads to

$$\hat{F}_k = \frac{W_k^*}{|W_k|^2 + \alpha |N_k|^2}. \tag{4.33}$$

Finally, the reflectivity estimate is given by

$$\begin{aligned} \hat{Q}_k &= D_k \tfrac{W_k^*}{|W_k|^2+\alpha|N_k|^2} \\ &= D_k \tfrac{W_k^*}{|W_k|^2+\mu} \,. \end{aligned} \tag{4.34}$$

Since the noise has a flat spectrum ($|N_k|^2 = \sigma_n^2$) we can replace $\alpha|N_k|^2$ by another constant μ. An estimate of the variance of the reflectivity estimate in the frequency domain is given by

$$Var(\hat{Q}_k) = |\hat{F}_k|^2 {\sigma_N}^2. \tag{4.35}$$

after a few manipulation we end up with

$$Var(\hat{Q}_k) = \frac{|W_k|^2 {\sigma_N}^2}{(|W_k|^2 + \mu)^2} \, . \tag{4.36}$$

When $\mu = 0$ the variance can be extremely high at the frequencies at which the wavelet power is small. Similarly we can find a expression for the norm of the reflectivity estimate in the frequency domain

$$\begin{array}{rl} N = & \sum_k |\hat{Q}_k|^2 \\ = & \frac{1}{{\sigma_N}^2} \sum_k |S_k|^2 Var(\hat{Q}_k) \end{array} \tag{4.37}$$

The misfit function is

$$\begin{array}{rl} \Phi & = \sum_k |S_k - W_k \hat{Q}_k|^2 \\ & = \frac{1}{{\sigma_N}^2} \sum_k |S_k|^2 (\frac{\mu}{|W_k|^2 + \mu})^2 \end{array} \tag{4.38}$$

Regularization error and noise magnification

If E_k denotes the deviation of the filter from the true inverse filter, defined by

$$E_k = 1 - \hat{F}_k W_k \tag{4.39}$$

we can write equation (4.12) as follows

$$\begin{array}{rl} \hat{Q}_k & = \hat{F}_k W_k Q_k + \hat{F}_k N_k \\ & = (1 - E_k) Q_k + \frac{1 - E_k}{W_k} N_k \end{array} \tag{4.40}$$

then, the difference between the true reflectivity Q_k and the reflectivity estimate $\hat{Q}_k$ is given by

$$\hat{Q}_k - Q_k = \underbrace{-E_k Q_k}_{RE} + \underbrace{\frac{1 - E_k}{W_k} N_k}_{NAE} \tag{4.41}$$

where RE stands for regularization error and NAE for noise amplification error. The NAE is independent of the data and can be expressed as a function of the wavelet:

$$\frac{W_k^*}{|W_k|^2 + \mu}$$

It is clear that the more the filter resembles the i nverse the wavelet W_k^{-1}, the larger this error will be. The RE introduce data-dependent degradation (i.e., ringing).

4.3 Sparse deconvolution and Bayesian analysis

The deconvolution operator is usually stabilized by adding a small perturbation to the diagonal of the autocorrelation matrix (Robinson and Treitel, 1980). The latter is equivalent to solving the problem by means of zero order quadratic regularization.

The regularization technique is used to estimate a unique and stable solution by introducing some type of prior information. In this part of the course, I will examine different regularization strategies that may be used to improve the deconvolution of seismic records. Specifically, I will use the Huber and the Cauchy criteria to retrieve a sparse reflectivity sequence. These criteria are related to "long tail" probability distributions (Huber, 1981) which, in a Bayesian context, are used as a prior distribution for the reflectivity sequence. The seismic wavelet is assumed to be known or to be well approximated by an estimator.

Re-weighting strategies have been used in conjunction with least squares type estimators to diminish the influence of outliers in inverse problems (Scales et al., 1988). In robust statistics the influence function (Huber, 1981) that measures the influence of the residuals on the estimators is constructed so as to attenuate outliers. A similar manipulation can be applied to the regularization function. In this case the goal is to attenuate the sidelobe artifacts which are introduced in the deconvolution process. In this context, the regularization strategy is used to drive the output of the deconvolution to a prescribed part of the model space, quite to the contrary to the classical application in robust statistics where the chief goal is to attenuate the influence of gross errors. When the problem is properly regularized the resolution of close seismic arrivals can be significantly enhanced. The described procedure is used to overcome the poor resolution associated to quadratic regularization strategies. We have to realize, however, that only when the seismogram is composed of a finite superposition of seismic wavelets (sparse reflectivity assumption) that these techniques may provide a substantial improvement with respect to conventional deconvolution.

4.3.1 Norms for sparse deconvolution

The deconvolution problem may be stated as follows. Consider

$$s_k = \sum_j w_j q_{k-j}, \qquad k = 1, n_y \tag{4.42}$$

where $q_k,\ k = 1, n_q$ and $s_k,\ k = 1, n_s$ are the input and the output series to the convolution process, respectively. The series $w_k,\ k = 1, n_w$ is the "blurring function" or source wavelet. In time domain deconvolution, the goal is to find $\hat{x}_k$ such that the residuals ϵ_k are minimized

$$\epsilon_k = s_k - \sum_j w_j \hat{q}_{k-j}, \qquad k = 1, n_s\,. \tag{4.43}$$

In the least squares approach the following objective function of the residuals is minimized:

$$J\ = \sum_k \rho_1(\frac{\epsilon_k}{\sigma_k})\,, \tag{4.44}$$

where

$$\rho_1(u) = \frac{1}{2}u^2\,. \tag{4.45}$$

The residuals are weighted according to the data accuracy that is given by the inverse of the standard error of each observation σ_k. For simplicity we shall assume that $\sigma_k = \sigma_n,\ k = 1, n_s$. The minimization of J is accomplished by solving the following system:

$$\frac{\partial J}{\partial q_l} = \sum_k \psi(\frac{\epsilon}{\sigma_n}) w_{k-l} = 0\,, \qquad l = 1, n_q \tag{4.46}$$

where $\psi(u) = \frac{d\rho(u)}{du}$. In robust statistics the function ψ is called the influence function. This function measures the influence of the residuals on the parameter estimators. The minimization of J leads to the following system of normal equations

$$\sum_k \sum_j w_{k-j} w_{k-l} q_j = \sum_k w_{k-l} s_k \,, \tag{4.47}$$

or in matrix form

$$\mathbf{Rq} = \mathbf{g} \,. \tag{4.48}$$

Equation (4.48) is stabilized by adding a small perturbation to the diagonal of the matrix $\mathbf{R}$. This is equivalent to minimize the following modified objective or cost function

$$J = J + J_q \,, \tag{4.49}$$

where $J_q = \sum_k \rho_1(\frac{q_k}{\sigma_q})$. The term J_q is the regulariser of the problem. This particular procedure is called zero order quadratic regularization. The cost function J which I will denote $J_{1\,1}$ is given by

$$J_{1\,1} = \sum_k \rho_1(\frac{\epsilon_k}{\sigma_n}) + \sum_i \rho_1(\frac{q_i}{\sigma_q}) \,. \tag{4.50}$$

The minimum of equation (4.50) is reached at the point

$$\hat{\mathbf{q}} = (\mathbf{R} + \mu \mathbf{I})^{-1} \mathbf{g} \,, \tag{4.51}$$

where $\mu = \sigma_n^2/\sigma_q^2$ is the damping parameter of the problem. In filter theory, μ is also called the pre-whitening parameter (Robinson and Treitel, 1980).

4.3.2 Modifying J_q

A standard procedure in robust statistics is based on redesigning the influence function in order to attenuate the influence of outliers. A similar modification can be used to design the regularization function, J_q.

The data misfit is modeled using the functional ρ_1, while for the regularization term I will introduce the following modification (Huber, 1981)

$$\rho_2(u) = \begin{cases} u^2/2 & \text{if } |u| \leq a \\ a\,|u| - a^2/2 & \text{if } |u| > a \end{cases} . \tag{4.52}$$

The deconvolution problem is solved by minimizing the following cost function

$$J_{1\,2} = \sum_k \rho_1(\frac{\epsilon_k}{\sigma_n}) + \sum_i \rho_2(\frac{q_i}{\sigma_q})\,. \tag{4.53}$$

The influence function for ρ_2 becomes:

$$\psi_2(u) = \begin{cases} u & \text{if } |u| \leq a \\ a\,\text{sign}(u) & \text{if } |u| > a \end{cases} \tag{4.54}$$

The function ρ_2 behaves identically to ρ_1 for small values of u. When $|u| > a$, ρ_2 defines a line and its associated influence function becomes a constant. We can define another function with similar behavior

$$\rho_3(u) = \ln(\frac{u^2}{2} + 1)\,. \tag{4.55}$$

When u is small $\rho_3 \to \rho_1$. The influence function corresponding to ρ_3 is given by

$$\psi_3(u) = \frac{u}{\frac{u^2}{2} + 1}\,. \tag{4.56}$$

If ρ_3 is adopted the deconvolution problem is solved by minimizing a cost function designated by $J_{1\,3}$

$$J_{1\,3} = \sum_k \rho_1(\frac{\epsilon_k}{\sigma_n}) + \sum_i \rho_3(\frac{q_i}{\sigma_q})\,. \tag{4.57}$$

The cost function $J_{1\,3}$ can be derived using Bayes' rule by assuming Gaussian errors and a Cauchy prior probability to model the unknown parameters (Sacchi and Ulrych, 1995).
Figure (4.7) shows the functions ρ_1, ρ_2 and ρ_3. In Figure (4.8) the corresponding influence functions ψ_1, ψ_2 and ψ_3 are displayed. The functions ρ_2 and ψ_2 were calculated for $a = 1$ and $a = 2$. When the parameter a is small compared to the normalized variable u the width of the transition zone $-a < u < a$ becomes very narrow. In this case the cost function $\sum_i \rho_2(u_i)$ behaves like a l_1 norm, $l_1 = \sum_i |u_i|$.

4.3.3 Iterative solution

The solution to the least squares problem with zero order quadratic regularization expressed in equation (4.51) can be modified to introduce into the regularization the functionals ρ_2 (Huber criterion) and/or ρ_3 (Cauchy criterion). In this case the system of normal equations is obtained by equating to zero the gradient of the objective function, $J_{1\,2}$ or $J_{1\,3}$,

$$(\mathbf{R} + \mu \mathbf{Q})\mathbf{q} = \mathbf{g}\,. \tag{4.58}$$

If the problem is regularized with ρ_2, the matrix $\mathbf{Q}$, which I will call $\mathbf{Q}_2$, has the following diagonal elements

$$Q_{2_{i\,i}} = \begin{cases} 1 & \text{if } |\frac{q_i}{\sigma_q}| \le a \\ \frac{a}{|\frac{q_i}{\sigma_q}|} & \text{if } |\frac{q_i}{\sigma_q}| > a \end{cases} . \tag{4.59}$$

When ρ_3 is adopted, the matrix $\mathbf{Q}$ in equation (4.58), which I will denote $\mathbf{Q}_3$, has the following diagonal elements

$$Q_{3_{i\,i}} = \frac{1}{1 + \frac{q_i^2}{2\sigma_q^2}}\,. \tag{4.60}$$

We can make an analogy with the zero order quadratic regularization to understand the effect of $\mathbf{Q}$ on solving the system expressed by equation (4.58). In the zero order quadratic regularization the damping term μ in equation (4.51) corresponds to the ratio of two variances, $\mu = \sigma_n^2/\sigma_q^2$. When ρ_2 or ρ_3 are used, the damping term becomes a ratio of the variance of the noise to a model dependent variance which I will designate $\sigma^2(q_i)$. This variable takes the following form when the problem is regularized with ρ_2:

$$\sigma^2(q_i) = \begin{cases} \sigma_q & \text{if } |\frac{q_i}{\sigma_q}| \leq a \\ \frac{\sigma_q |q_i|}{a} & \text{if } |\frac{q_i}{\sigma_q}| > a \end{cases} . \tag{4.61}$$

The last equation shows that above a threshold the variance of x_i is proportional to $|x_i|$. A similar argument leads to the variance for the regularization with ρ_3

$$\sigma^2(q_i) = \sigma_q^2 + \frac{q_i^2}{2} , \tag{4.62}$$

in which case the variance has a parabolic growth with the amplitude of q. Equation (4.58) is solved using the following iterative scheme:

1. Start with an initial reflectivity sequence $\mathbf{q}^0$

2. Select the hyperparameters of the problem σ_n, σ_q, and a (Huber criterion) or σ_n and σ_q (Cauchy criterion).

3. Compute $\mu = \sigma_n^2/\sigma_q^2$, $\mathbf{Q}^{(0)}$, and the source autocorrelation matrix $\mathbf{R}$.

4. Iteratively solve equation (4.58) using the following algorithm

$$\mathbf{q}^{(k)} = (\mu \mathbf{Q}^{(k-1)} + \mathbf{R})^{-1} \mathbf{g} \tag{4.63}$$

   where k is the iteration number.

5. The procedure is stopped when the following tolerance criterion is satisfied

$$\frac{|J^{(k)} - J^{(k-1)}|}{(|J^{(k)}| + |J^{(k-1)}|)/2} < tolerance \tag{4.64}$$

where $J = J_{1\,2}$ or $J_{1\,3}$ depending on the regularization criterion.

6. Compute the data misfit. Select new hyperparameters if the misfit is not satisfactory. The strategy for hyperparameter selection is discussed in the following section.

Each iteration demands one matrix inversion. The procedure can be accelerated by using an iterative solver like the conjugate gradient (CG) algorithm. The advantage of using CG is that an approximate solution can be computed by truncating the number of iterations.

The effect of $\mathbf{Q}$ can be summarized as follows. In each iteration, the nonlinearity produces a solution that has the minimum amount of structure or maximum sparseness. The validity of this type of solutions is subject to the validity of the sparse reflectivity assumption.

4.3.4 Hyperparameter selection

The determination of the parameter μ in equation (4.58) is crucial, but unfortunately it cannot be determined a priori. A wrong selection of μ may yield a solution that is unreasonable. We will assume that the variance of the noise σ_n^2 is known. If the Cauchy criterion is adopted, only one independent parameter σ_x must be determined ($\mu = \sigma_n^2/\sigma_q^2$). When the Huber criterion is used, two independent parameters are needed: σ_q and a. The parameter a is assigned as follows $a = c \times \sigma_q$, where c is a scalar ($c = 0.1 - 1.$). If c is large ($c > 2$) the Huber criterion behaves like the standard quadratic form ρ_1.

We adopt the discrepancy principle which determines the parameter σ_q from the requirement that the data misfit matches the power of the noise. Since we have assumed that the noise is normally distributed the data misfit obeys a χ^2 statistics

$$\chi^2 = \frac{1}{\sigma_n^2} \sum_{k=1}^{n_s} \epsilon_k^2 \, . \tag{4.65}$$

The expected value of the χ^2 statistic is used as a target misfit, $E[\chi^2] = n_s$ (n_s is the number of observations), where the largest acceptable value at

99% confidence limit is $\approx n_s + 3.3\sqrt{n_s}$.

Figure (4.9a) portrays a simulated reflectivity impulse response for a simple earth model, the seismogram generated by convolving the impulse response with the source, and the seismic source. Gaussian noise was added to the synthetic seismogram with standard deviation $\sigma_n = 5 \times 10^{-2}$. This represents a relative noise amplitude of 17.5%. The relative noise magnitude is specified by a percentage of the maximum noise-free seismogram amplitude that its standard deviation represents $(\sigma_n / \max(s_k) \times 100)$.

The deconvolution was carried out using zero order quadratic regularization (minimizing $J_{1\,1}$), the Huber criterion ρ_2 (minimizing $J_{1\,2}$) and the Cauchy criterion ρ_3 (minimizing $J_{1\,3}$). The estimated impulse responses are displayed in Figures 4.9b, c, and d together with the reconstructed seismograms and residuals (original minus reconstructed data). The parameter σ_s was selected according to the χ^2 criterion.

The solution with ρ_1 (zero order quadratic regularization) does not allow us to properly identify each arrival. The re-weighted strategy yields highly resolved estimates of the position and amplitude of each seismic reflection.

In general, about $5-10$ iterations are sufficient to find a good approximation to the minimum of the cost function.

The portion of stacked seismic section shown in Figure (4.10a) is used to test the performance of the re-weighted deconvolution procedure when dealing with field data. The stacked section is obtained by applying normal moveout correction, and summing traces from common mid point (CMP) gathers. The data consist of 24 traces which delineate several seismic horizons. In particular, we are interested in the coherent events at $\approx$ 0.95sec which may represent a thin layer. The seismic wavelet (4.11) was extracted using a cepstrum-cumulant approach. The wavelet is retrieved using two different techniques which are radically different in formulation: a cepstrum decomposition and a fourth-order cumulant matching approach. Since the recovered wavelets were very similar, the confidence in the wavelet estimators increases. It is important to stress that, unlike in many deconvolution scenarios, in seismic deconvolution the kernel function or source wavelet is unknown. The deconvolved data are shown in Figure (4.10). In this example, the problem is regularized by means of the Cauchy criterion (ρ_3).

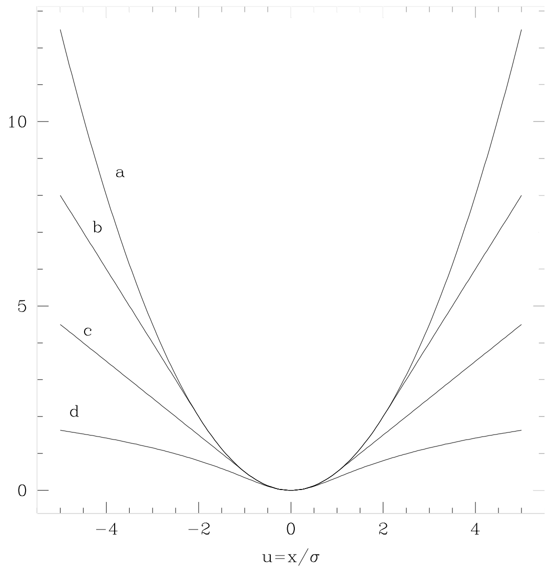


Figure 4.7: Cost functions. a) $\rho_1(u)$, b) $\rho_2(u)\, a = 1$, c) $\rho_2(u)\, a = 2$, and d) $\rho_3(u)$.

The χ^2 criterion was used to estimate the parameter σ_x. A relative noise amplitude of 2% was assumed. The latter was used to estimate the standard error of the noise σ_n. Similar results were obtained using Huber's weights.

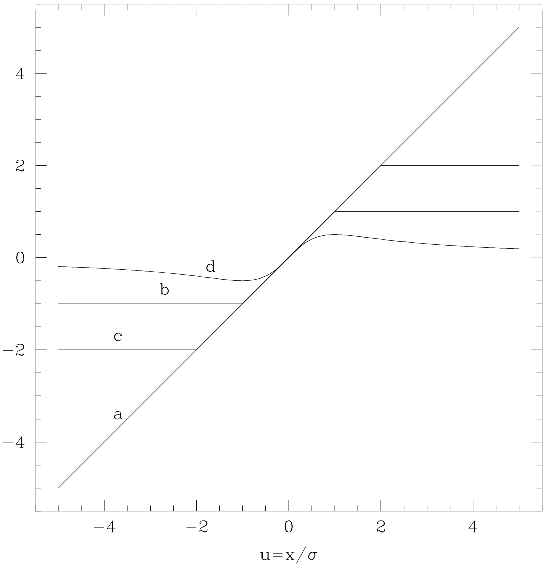


Figure 4.8: Influence functions. a) $\psi_1(u)$, b) $\psi_2(u)\,a = 1$, c) $\psi_2(u)\,a = 2$,and d) $\psi_3(u)$.

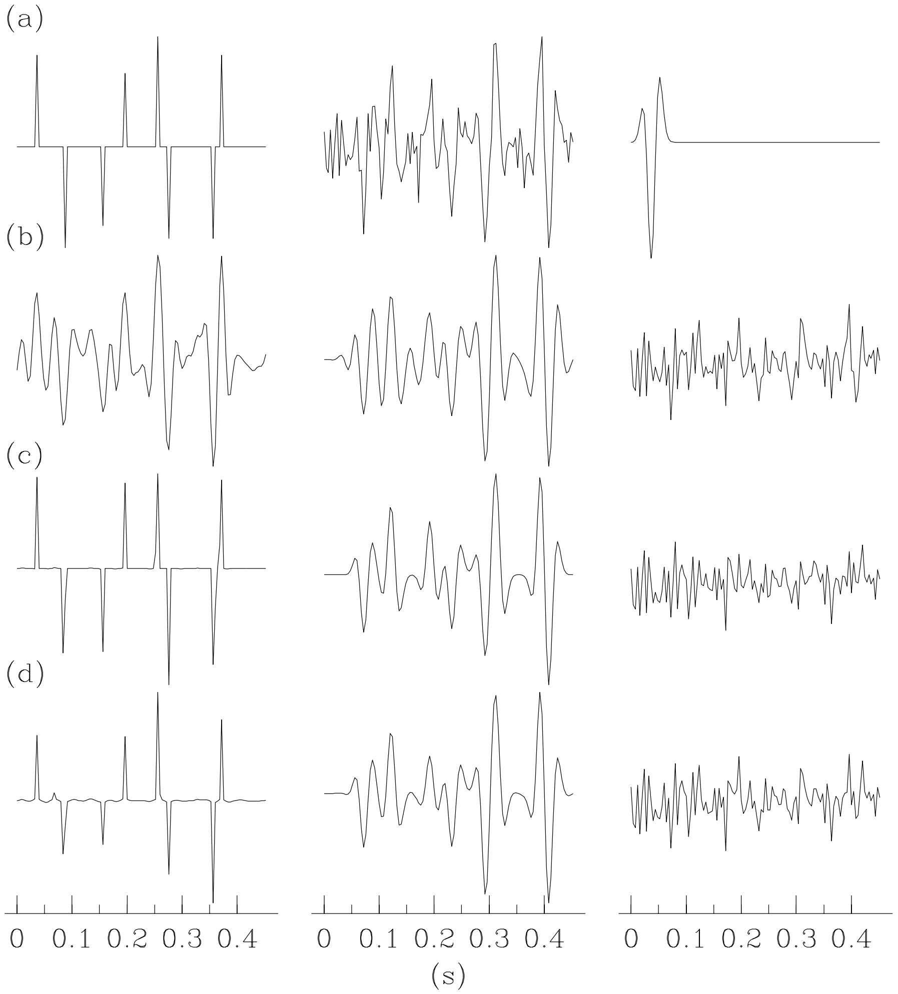


Figure 4.9: a) Synthetic impulse response (left) , seismogram (center), and source wavelet (right). The seismogram was contaminated with Gaussian noise. b) Deconvolution using zero order quadratic regularization: estimated impulse response (left), reconstructed seismogram (center), and residuals (original minus reconstructed data). c) Deconvolution by means of the Huber criterion ($a = 1$). d) Deconvolution by means of the Cauchy criterion.

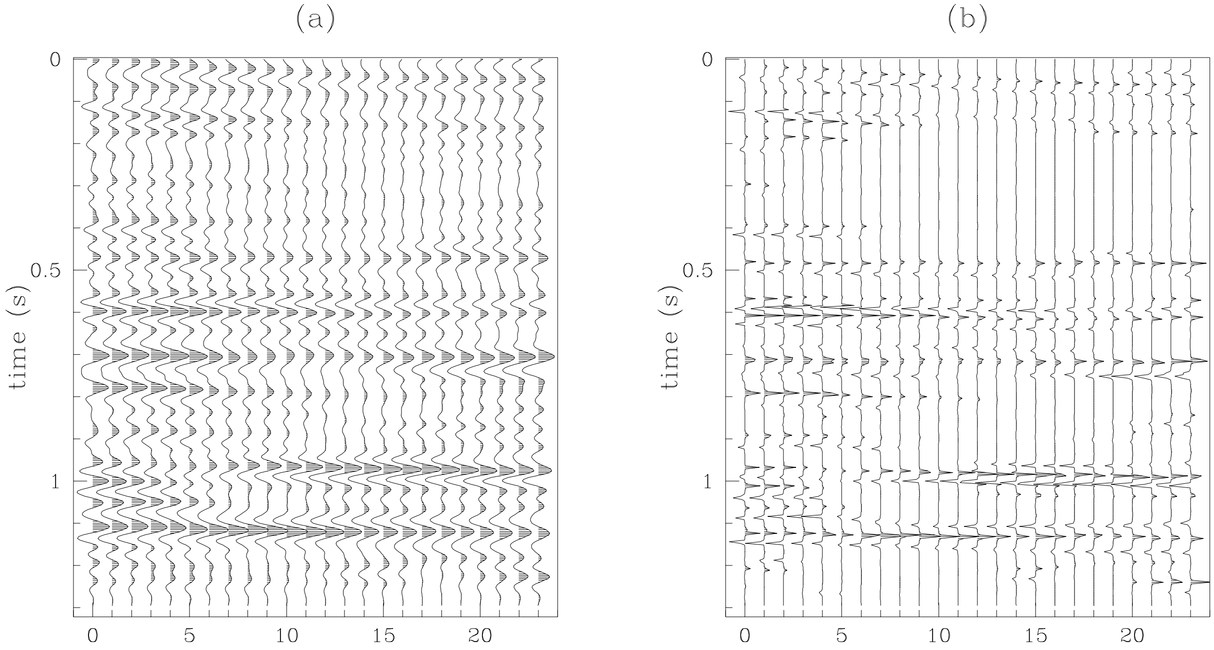


Figure 4.10: (a) Original seismic section. (b) Deconvolved seismic section using the Cauchy criterion to regularize the inversion. The source wavelet was retrieved using a combined cepstrum-cumulant approach.

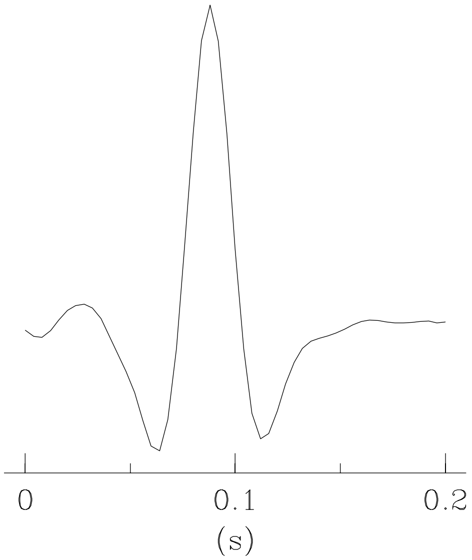


Figure 4.11: Seismic wavelet corresponding to the data inverted in the previous Figure.

4.4 Bayesian inversion of impedance

In this section we discuss the problem of estimating a "blocky" impedance model from normal incidence data. We will adopt again the convolution model

$$s_k = w_k * q_k \,. \tag{4.66}$$

The reflectivity is denoted by q_k and the wavelet by w_k. The goal is to recover q_k from a noisy version of s_k. We assume that impedance constraints are provided at different time levels. After writing the usual logarithmic approximation for the impedance

$$\xi_k = \frac{1}{2}\ln(z_k/z_0) = \sum_{i=1}^{N_k} x_i \,, \tag{4.67}$$

we are now in condition of writing equations (4.66) and (4.67) as two linear system of equations. In matrix notation

$$\mathbf{Wq} = \mathbf{s} + \mathbf{n} \tag{4.68}$$

$$\mathbf{Cx} = \xi + \epsilon \,, \tag{4.69}$$

where the matrices $\mathbf{W}$ and $\mathbf{C}$ correspond to the wavelet matrix and to the impedance constraint matrix, respectively. The matrix $\mathbf{W}$ contains the wavelet properly padded with zeros in order to express discrete convolution in matrix form. The matrix $\mathbf{C}$ is a simple integrator operator. Bayes' rule is used to incorporate the prior probability of the unknown reflectivity $\mathbf{q}$ into the problem. In many applications we want to estimate a blocky impedance profile. In this case, a 'long tailed' distribution may be used to estimate a sparse reflectivity sequence.

Noise in the trace is modelled by means of the usual Gaussian assumption. The uncertainties of the constraints are also assumed Gaussian (note that we are using the variable ξ which can take positive and negative values and not z which is strictly positive).

Bayes' rule is used to define the posteriori probability of the reflectivity sequence. The solution is computed by maximizing the posteriori probability (MAP solution). This is equivalent to minimize the following cost function:

$$J = \alpha \underbrace{J_q}_{1} + \underbrace{\frac{1}{2}||\frac{1}{\sigma}(\mathbf{Wq}-\mathbf{s})||^2}_{2} + \underbrace{\frac{1}{2}||\mathbf{S^{-1}}(\mathbf{Cq}-\xi)||^2}_{3} \tag{4.70}$$

where σ^2 is the variance of the noise in the seismic trace and the matrix $\mathbf{S}$ is a diagonal matrix with the following elements

$$S_{ii} = \sigma_{c_i} \,. \tag{4.71}$$

In equation (4.70) we are specifying three different features that the solution must satisfied:

- 1 - The solution must be sparse.
- 2 - The solution must honour the seismic trace.
- 3 - The solution must honour a set of impedance constraints.

The parameter α is the weighting parameter or hyperparameter that determines the relative amount of sparseness that can be brought into the inversion. The term J_q is derived using four different priors which induce the associated regularization criteria for sparse spike inversion. The four sparseness criteria that we have studied are the L_p criterion, the *Cauchy* criterion, the *Sech* criterion, and *Huber* criterion. These criteria were all used in robust statistics to diminish the influence of outliers in estimation problems (Huber, 1981). In our application, these criteria are used to impose sparseness into the reflectivity estimate. J_q in equation (4.70) is given by one of the following regularization terms

$$J_p = \frac{1}{p}\sum_i |q_i|^p \tag{4.72}$$

$$J_{Cauhcy} = \frac{1}{2}\sum_i \ln(1+\frac{q_i^2}{\sigma_q^2}) \tag{4.73}$$

$$J_{Sech} = \sum_i \ln(\cosh \frac{q_i}{\sigma_q}) \tag{4.74}$$

$$J_{Huber} = \sum_i \left\{ \begin{array}{ll} q_i^2/2 & \text{if } |q_i| \le q_c \\ a\,|q_i| - q_c^2/2 & \text{if } |q_i| > q_c \end{array} \right. . \tag{4.75}$$

A non-linear conjugate gradient algorithm is used to minimize the cost function. The parameter α is continuously adapted to satisfy a misfit criterion. As an example we show the inversion of two traces with the L_p ($p = 1.2$) prior and with the *Cauchy* criterion. The hyperparameters of the problem were fitted using a χ^2 criterion. The same misfit target was used in both examples, $\chi^2 =$ number of traces + number of constraints.

In Figure (4.12) we portray a window of a sesimic section pre-processed for impedance inversion. The inversion was carried out using the *Huber* and L_p criteria ($p = 1.1$). The reflectivities estimates are shown in Figures (4.13) and (4.15). The constrained inversion of the impedance profile is dipected in Figures (4.14) and (4.16).

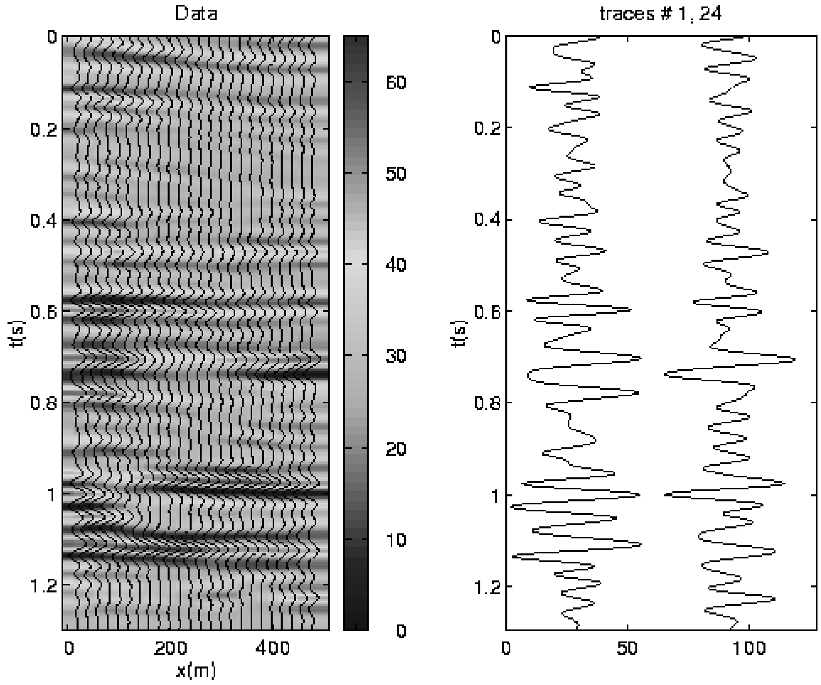


Figure 4.12: data

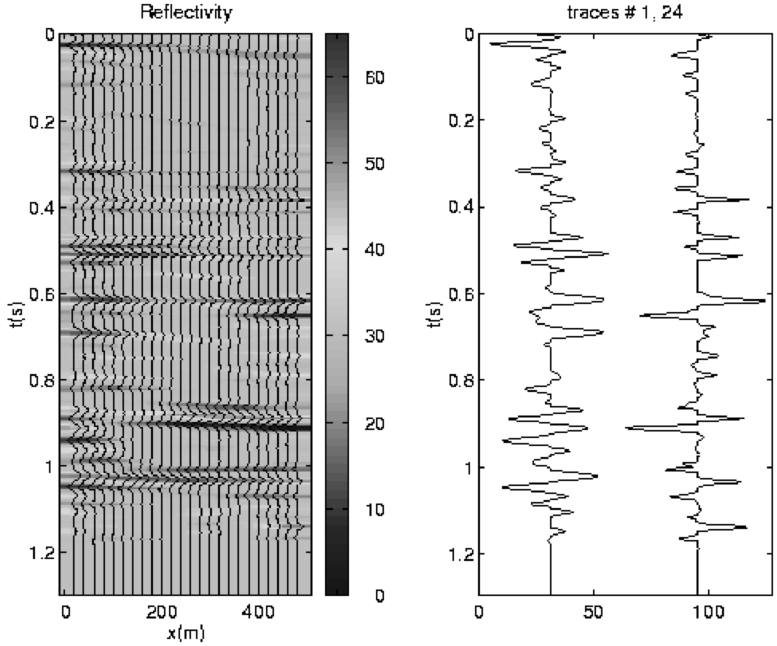


Figure 4.13: Reflectivity inversion using the L_p norm, $p = 1.1$

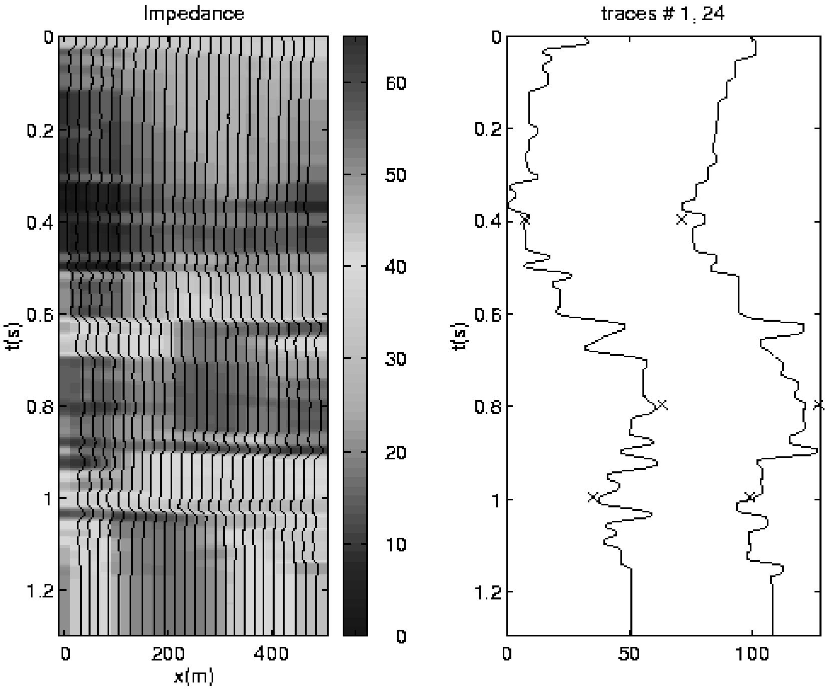


Figure 4.14: Constrained impedance inversion using the L_p norm, $p = 1.1$.

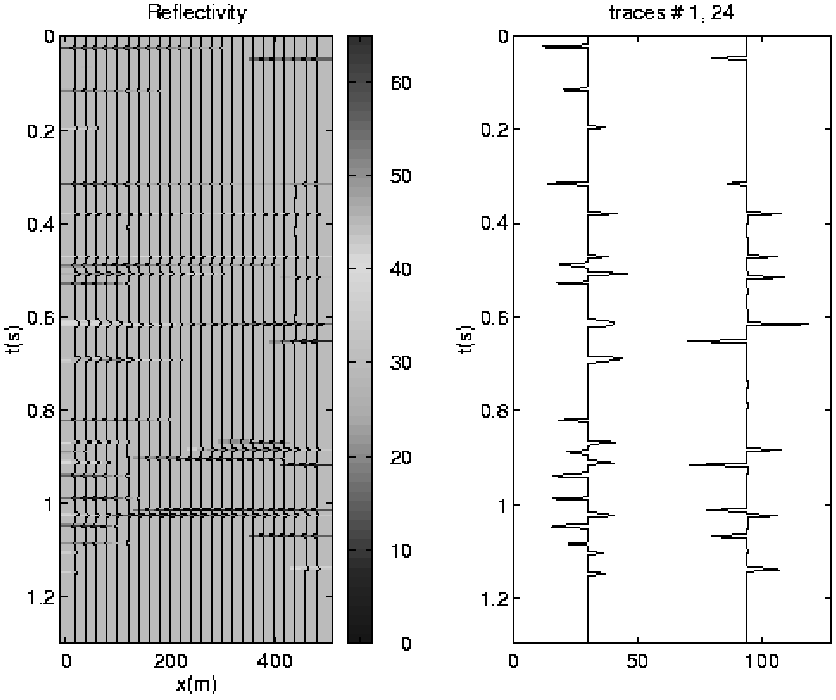


Figure 4.15: Reflectivity inversion *Huber* norm.

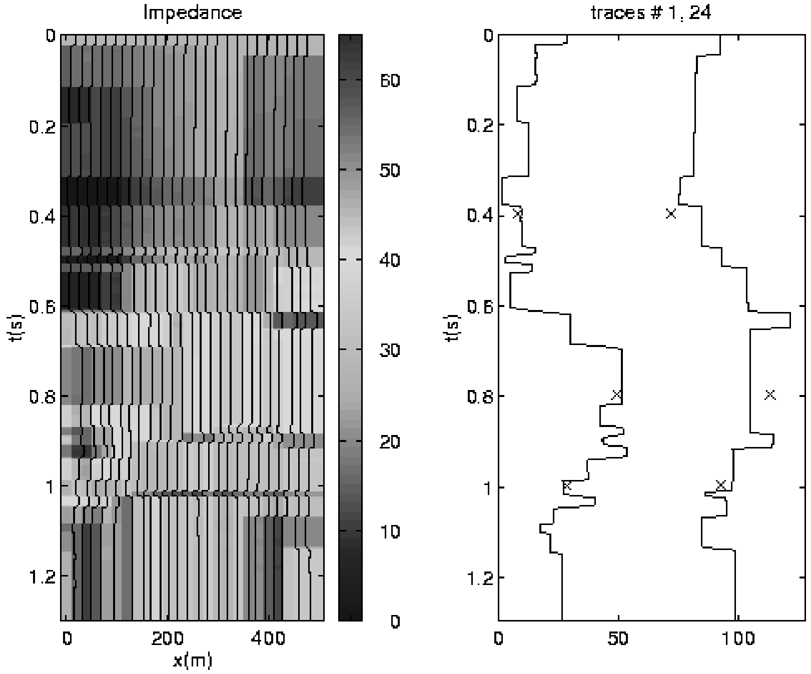


Figure 4.16: Constrained impedance inversion using *Huber* norm

4.5 Linear programming impedance inversion

In this section I will discuss the classical approach to impedance inversion proposed by Oldemburg et al. (1983).

I will also provide a subroutine to perform the L_1 inversion of a seismogram using the a Linear programming solver.

The subroutine `l1_inv`[4] is designed to perform 1-D inversion of the acoustic impedance. The algorithm permits the incorporation of impedance constraints at different times in the form of upper and lower bounds. The convolution model is adopted:

$$s_k = w_k * q_k \,. \tag{4.76}$$

The wavelet is assumed to be known or to be well approximated by an estimate. The reflectivity is denoted by q_k and the wavelet by w_k. The goal is to recover q_k from a noisy version of s_k. The algorithm also assumes that impedances constraints are provided at different time levels. In such a case we can write the usual logarithmic approximation for the impedance

$$\xi_k = \frac{1}{2}\ln(z_k/z_0) = \sum_{i=1}^{N_k} q_i \,, \tag{4.77}$$

In matrix notation we can write

$$\mathbf{Wq} = \mathbf{s} + \mathbf{e} \tag{4.78}$$

$$\mathbf{Cq} = \xi \tag{4.79}$$

where the matrices $\mathbf{W}$ and $\mathbf{C}$ correspond to the wavelet matrix and to the impedance constraint matrix, respectively. The matrix $\mathbf{W}$ contains the wavelet properly padded with zeros in order to express discrete convolution in matrix form. The matrix $\mathbf{C}$ is a simple integrator operator.

The procedure proposed to recover the reflectivity is based on the minimization of the l_1 cost function of the problem. Instead of using a Conjugate

[4] Let me know if you want to have the complete f77 source code

Gradient technique to minimize the cost function (like in the Bayesian approach discussed in the previous section.) we adopt the linear programming approach.

4.5.1 Constrained minimization using linear programming

The cost function of the problem is defined as

$$J = \alpha|\mathbf{q}|_1 + |\mathbf{e}|_1 \,. \tag{4.80}$$

and the constraint minimization problem is set as follows

$$\text{Minimize} \quad J = \alpha|\mathbf{q}|_1 + |\mathbf{e}|_1 \tag{4.81}$$

$$\text{subject to} \quad \mathbf{Wq} = \mathbf{s} + \mathbf{e} \tag{4.82}$$

$$\text{and} \quad \xi_l < \mathbf{Cq} < \xi_{\mathbf{u}} \,. \tag{4.83}$$

The last problem is solved using the linear programming approach. We first specify a tableau (`subroutine tableau`) where we load the objective function, the data constraint and the inequality constraints. The Linear programming solution is then invoked to retrieve the solution (`subroutine cl1`). The parameter α is the tradeoff parameter of the problem. In the program `alpha` is given in percentage of the l_1 norm of the wavelet. This is analogous to the pre-whitening parameter in spiking deconvolution which is given as percentage of the zero lag autocorrelation coefficient.

Usually, $1-10\%$ should be enough to stabilize the inversion. The parameter also controls the sparseness of the solution (number of nonzero reflectivity samples). When $\alpha \to 0$ the inversion is too rough (unstable). When α is too large (50%) the inversion is too sparse.

4.5.2 Example

A sparse reflectivity series convolved with a 20Hz Ricker wavelet is used to test the algorithm. The length of the trace is 150 samples. The additive noise is white and Gaussian (SNR=20). In this example we retrieved impedance bounds every 10 samples. The total number of impedance bounds to be honoured is $nic = 15$. The results are shown in Figure (4.17).

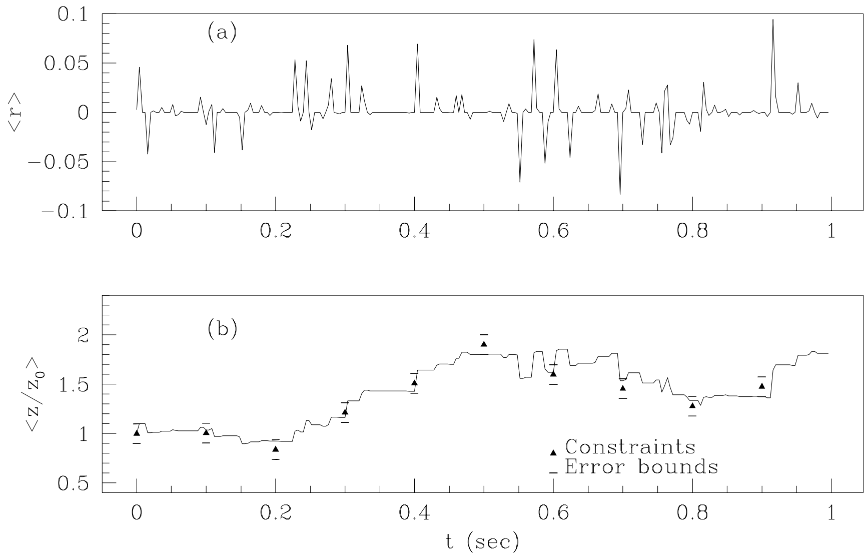


Figure 4.17: (a) Seismic reflectivity. (b) Input seismic trace. (c) L_1 inversion of the reflectivity. (d) Predicted trace (convolution of the wavelet with the inverted reflectivity). (e) True impedance profile. The bars indicate the impedance bounds (hard constraints). (f) Impedance recovered with the L_1 inversion, note that the bounds are properly honoured

4.5.3 Linear programming code

The subroutine `l1_inv` is the basic program that you need for sparse inversion. Some variables were set with default values. The subroutine `tableu` is used to load the objective function of the problem, the data constraints and the impedance bounds.

The code is too long, this is why I only print the part containing the comments. If you wish to try with this code, I have a ready-to-use version in a disk.

```
      subroutine l1_inv(t,n1,w,nw,
     *             nic,ncon,z_upper,z_lower,
     *             alpha,r,tp,iter,e1,e2)


c This subroutine does the l1 inversion of a seismogram
c using the linear programming approach


c INPUT
c    n1    :  number of samples of the trace
c    t(n1) :  the seismic trace (properly scaled)
c    nw    :  number of points of the wavelet
c    w(nw) :  the wavelet (properly scaled)
c    nic   :  number of impedance constraints
c    ncon(nic) : position of each impedance constraint in samples
c    z_upper(nic): upper impedance constraint at the position ncon
c    z_lower(nic): lower impedance constraint at the position ncon
c                        the impedance bounds are for the variable
c                        0.5ln(z/z0) (see text)
c    alpha : regularization parameter.
c            alpha is a percentage of the l1 norm of the wavelet.
c            In spiking deconv.(l2) is a percentage of the main diagonal
c            of the autocorrelation matrix. The strategy that I have
c            adopted is similar. Use 1-10%.
c
c OUTPUT
c    r(n1) : the reflectivity estimate
c    tp(n1): the predicted seismic trace, i.e., r(t)*w(t)
c    iter  : number of iterations that the linear programing routine
c            used to find a feasible solution
```

```
c    e1    : the l1 error
c    e2    : the rms error
c            e1,e2 are the misfit figure.


c  SUBR:   tableau : make the tableau for the lprog.
c          cl1     : solve the lprog. problem

c  NOTES: the varialbe KODE serves to check is the lprog. routine
c         has found a feasible solution. KODE=0 (always) before
c         calling to cl1 (The linear prog).
c         KODE=0 (always) after calling cl1 for normal execution.


      parameter (nrd=400,nwd=101,ntd=500,ncd=400,
     *           klmd=nrd+2*ntd+2*ncd,
     *          klm2d=nrd+2*ntd+2*ncd+2,
     *           nklmd=2*(ntd+nrd)+ntd+2*ncd,
     *           n2d  =ntd+nrd+2)

      real * 8   q(klm2d,n2d), x(n2d), res(klmd)
      real * 8   cu(2,nklmd)
      integer    iu(2,nklmd), s(klmd)

      real * 8   w(nwd),r(nrd),t(ntd),tp(ntd)
      real * 8   z_upper(nrd),z_lower(nrd)
      integer    ncon(nrd)

      real * 8   toler
      real * 8   alpha

      real * 8   e1,e2

      nr=n1
      nt=n1+nw-1
      do i=nr+1,nt
      t(i)=0.
      enddo

c   Prepare the tableau for the simplex
      call tableau(nw,w,nr,nt,t,alpha,nic,ncon,zcon,
     #              z_upper,z_lower,q,
     #              k,l,m,n)


     .
     .
     .
     .
```

4.6 Non-minimum phase wavelet estimation

In this section we will analyse the problem of estimating a non-minimum phase wavelet for seismic deconvolution/inversion. The wavelet we have used to process our data in the preceeding section was estimated using a cumulant approach. We will discuss here how to estimate the seismic wavelet using the aforementioned procedure.

One of the early techniques for nonminimum phase signal decomposition has been homomorphic filtering based on the complex cepstrum (Ulrych,1971) or differential cepstrum (Bednar and Watt, 1985). Anyone exposed to homomorphic filtering has probably experienced the gap that exits between theory and practice. The approach is elegant and general (Hilbert transformation and spectral factorization can be implemented by means of the homomorphic transform.) However, practical problems have provided serious obstacles for the cepstral approach in the deconvolution arena. In particular cepstral decomposition assumes that the reflectivity is an impulsive train that can be completely separated from the wavelet in the cepstral domain (Ulrych, 1971). In fact, the technique does not even consider the reflectivity sequence as a stationary process. Another problem with the cepstrum is that the homomorphic system does not allow for the presence of additive noise in the formulation of the problem. In fact, the highly nonlinear nature of the cepstrum complicates the effect of additive noise.

The definition of the cepstrum in terms of higher order cumulants enables us to retrieve the cepstrum of the source wavelet (analytically) when the reflectivity does not consist of a train of spikes. Since the bispectrum and the trispectrum conserve the phase characteristic of the wavelet it is evident that the cepstrum derived from these polyspectra will also conserve phase information.

4.6.1 Non-minimum phase system identification

The classical system identification literature has been primarily dominated by least squares approaches. Least squares based techniques are attractive since they yield maximum likelihood estimates of parameters where the observable are a linear combination of a white sequence. These identification procedures use the autocorrelation function of the data. However, the autocorrelation function annihilates the phase of the system (c.f. the wavelet), and therefore these techniques are useful primarily to identify minimum phase systems, or systems where the phase can be specified a priori i.e., a zero phase assumption.

We begin with Wold's decomposition theorem. According to this celebrated theorem any discrete stationary process can be expressed as the sum of two independent processes, one purely deterministic, and another purely non-deterministic. The decomposition theorem also states that the purely non-deterministic part can be written in terms of a linear transformation of a white process, ϵ_k. In other words as long as the process is stationary, there is always a representation of the process given by

$$x_t = \sum_k \epsilon_k h_{t-k} \, , \tag{4.84}$$

where for simplicity we have omitted the deterministic part of the process which is not needed in the seismic deconvolution scenario. Clearly, the Wold decomposition theorem dictates that a decomposition exists where the wavelet h_k is causal and minimum phase. It is clear that one of the consequences of this theorem is that the description of the process is non-unique. One can generate a stationary nonlinear process and according to the theorem this process will have a MA representation given by equation (4.84).

We will use the following model to describe the seismic trace

$$\begin{aligned} x_k = &\quad r_k * w_k \\ s_k = &\quad q_k + n_k \end{aligned} \tag{4.85}$$

where x_k is the noise-free seismic trace, and s_k is the trace corrupted with noise. The source wavelet is denoted by w_k and the reflectivity sequence by q_k. We will assume that q_k is white and iid. The noise n_k is considered zero-mean, Gaussian, and independent of x_k. The transfer function of the system is given by

$$W(z) = |W(z)|e^{i\Phi(z)} \,. \tag{4.86}$$

The seismic deconvolution problem is to reconstruct the magnitude and the phase of the transfer function from the output data s_k. Thus far, we have only considered that q_k is white. To continue with the analysis the following comments are in order:

1. If q_k is Gaussian and $W(z)$ is minimum phase, autocorrelation based methods will correctly identify both the amplitude and the phase of the system.

2. If q_k is Gaussian and $W(z)$ is nonminimum phase, no technique will correctly identify the phase of the system.

3. If q_k is non-Gaussian and $W(z)$ is nonminimum phase, true magnitude and phase of the system transfer function can be recovered by knowing the actual distribution of q_k. For MA processes of order one, it has been demonstrated that a L_1 optimization provides an estimate of the amplitude and phase of the system when the driving noise of the process is non-Gaussian (Scargle, 1977).

The above statement (3) is very important. In fact, it suggests that we can still have faith in stochastic wavelet estimation procedures. It is also clear that non-Gaussianity plays a key role in nonminimum phase wavelet estimation. In fact, MED-type estimators are an example where departure from normality is postulated. The non-uniqueness expressed by Wold theorem is eliminated by restricting q_k to be non-Gaussian.
Higher order spectra are defined in terms of higher order cumulants and contain information regarding deviation from Gaussianity. Quite contrary to power spectral density estimates, higher order spectra retain phase information which allows nonminimum phase system identification/estimation.

4.6.2 The bicepstrum

If $W(z)$ is stable and if q_k is non-Gaussian, white, iid., with skewness $\beta \neq 0$ then the bispectrum of x_k is given by

$$B_x(z_1, z_2) = \sum_m \sum_n r_x^{(3)}(m, n) z_1^n z_2^m \tag{4.87}$$

where

$$r_x^{(3)}(m, n) = E[x_k x_{k+m} x_{k+n}]\,, \tag{4.88}$$

is the third order moment of the data. Since the third order moment of a Gaussian signal vanishes, we can write $B_s(z_1, z_2) = B_x(z_1, z_2)$. The bispectrum can be written in terms of the transfer function $W(z)$ as follows (Nikias and Raghuveer, 1987)

$$B_x(z_1, z_2) = \beta W(z_1) W(z_2) W(z_1^{-1} z_2^{-1})\,. \tag{4.89}$$

Since the wavelet is, in general, nonminimum phase, the transfer function can be written as

$$W(z) = A z^l W^{min}(z) W^{max}(z^{-1}) \tag{4.90}$$

where A is a constant, l an integer associated with a linear phase shift, and $W^{min}(z)$, $W^{max}(z^{-1})$ are the minimum and maximum phase components of the wavelet, respectively. Substituting equation (4.90) into (4.89) the bispectrum becomes

$$\begin{aligned} B_x(z_1, z_2) = \beta W^{min}(z_1) W^{min}(z_2) \\ W^{max}(z_1^{-1}) W^{max}(z_2^{-1}) \\ W^{min}(z_1^{-1} z_2^{-1}) W^{max}(z_1 z_2)\,. \end{aligned} \tag{4.91}$$

Now, we define the bicepstrum as

$$\hat{b}(n, m) = Z^{-1}[\ln\, B_x(z_1, z_2)] \tag{4.92}$$

where Z^{-1} stands for the inverse 2-D z-transform. We note that

$$\begin{aligned}\ln[B_x(z_1,z_2)] &= \ln[\beta] \\ &+ \ln[W^{min}(z_1)] + \ln[W^{min}(z_2)] \\ &+ \ln[W^{max}(z_1^{-1})] + \ln[W^{max}(z_2^{-1})] \\ &+ \ln[W^{min}(z_1^{-1}z_2^{-1})] + \ln[W^{max}(z_1z_2)]\,. \end{aligned} \tag{4.93}$$

The inversion of equation (4.93) yields a complete separation into minimum phase and maximum phase components. Since the homomorphic transform maps minimum phase components into the positive semi-axis of the cepstrum and maximum phase components into the negative semi-axis, it is easy to verify that

$$\hat{b}(m,0) = \hat{w}^{min}(m), \quad m > 0 \tag{4.94}$$

$$\hat{b}(m,0) = \hat{w}^{max}(m), \quad m < 0\,. \tag{4.95}$$

where $\hat{w}(n)$ indicates the cepstrum of the wavelet. It is clear that with only two semi-axes of the bicepstrum we can completely define the cepstrum of the wavelet (the value $\hat{w}(0)$ cannot be recovered but represents only a scale factor).

We use a similar approach below to compute the tricepstrum (the cepstrum of the fourth order cumulant).

4.6.3 The tricepstrum

Defining the trispectrum as

$$T_x(z_1,z_2,z_3) = \sum_m \sum_n \sum_l r_x^{(4)}(n,m,l) z_1^n z_2^m z_3^l \tag{4.96}$$

where $r_x^{(4)}(n,m,l)$ is now the fourth order cumulant of x_k. The trispectrum can be written in terms of the transfer function of the system as follows

$$\begin{aligned} T_x(z_1,z_2,z_3) = \gamma A^4 W^{min}(z_1)W^{min}(z_2)W^{min}(z_3) \\ W^{max}(z_1^{-1})W^{max}(z_2^{-1})W^{max}(z_3^{-1}) \\ W^{min}(z_1^{-1}z_2^{-1}z_3^{-1})W^{max}(z_1z_2z_3) \end{aligned} \tag{4.97}$$

After taking logarithm of the trispectrum we end up with

$$\begin{aligned} \ln[T_x(z_1,z_2,z_3)] = \;& \ln[\gamma A^4] + \ln[W^{min}(z_1)] + \ln[W^{min}(z_2)] + \ln[W^{min}(z_3)] \\ & + \ln[W^{max}(z_1^{-1})] + \ln[W^{max}(z_2^{-1})] + \ln[W^{max}(z_3^{-1})] \\ & \ln[W^{min}(z_1^{-1}z_2^{-1}z_3^{-1})] + \ln[W^{max}(z_1z_2z_3)]\,. \end{aligned} \tag{4.98}$$

The inversion of the last expression will map minimum and phase components into the tricepstrum domain as follows:

$$\hat{t}(m,n,l) = \begin{cases} \ln[\gamma A^4] & m=n=l=0 \\ \hat{w}_{min}(m) & m>0, \quad n=l=0 \\ \hat{w}_{min}(n) & n>0, \quad m=l=0 \\ \hat{w}_{min}(l) & l>0, \quad m=n=0 \\ \hat{w}_{max}(m) & m<0, \quad n=l=0 \\ \hat{w}_{max}(n) & n<0, \quad m=l=0 \\ \hat{w}_{max}(l) & l<0, \quad m=n=0 \\ \hat{w}_{min}(m) & m=n=l<0 \\ \hat{w}_{max}(m) & m=n=l>0\,. \end{cases}$$

The origin of tricepstrum, $\hat{t}(0,0,0) = \ln(A^4\gamma)$ represents a scale factor and, therefore, can be ignored.

In general, we estimate the wavelet from 2 semi-axis of the tricepstrum:

$$\hat{t}(m,0,0) = \hat{w}^{min}(m), \quad m>0 \tag{4.99}$$

$$\hat{t}(m,0,0) = \hat{w}^{max}(m), \quad m<0\,. \tag{4.100}$$

4.6.4 Computing the bicepstrum and the tricepstrum

We first note that if $z_1 = e^{-i\omega_1}$ and $z_2 = e^{-i\omega_2}$ we have

$$\begin{aligned} \hat{B}_x(\omega_1, \omega_2) &= \ln[B_x(\omega_1, \omega_2)] \\ &= \ln|B_x(\omega_1, \omega_2)| + i[2k\pi + \Phi(\omega_1, \omega_2)]\,. \end{aligned} \tag{4.101}$$

Since the phase of the complex variable $\hat{B}_x(\omega_1, \omega_2)$ is undefined it appears that an unwrapping procedure is mandatory . Fortunately, the unwrapping can be omitted by defining the differential cepstrum, as follows

$$\frac{d\hat{B}_x(\omega_1, \omega_2)}{d\omega_1} = \frac{1}{B_x(\omega_1, \omega_2)} \frac{dB_x(\omega_1, \omega_2)}{d\omega_1} \tag{4.102}$$

where the derivatives are obtained by

$$\frac{dB_x(\omega_1, \omega_2)}{d\omega_1} = \mathcal{F}_2[-imr_{x_3}(m, n)] \tag{4.103}$$

and we can write (4.102) as

$$\hat{b}(m, n) = \frac{1}{m} \mathcal{F}_2^{-1}[\frac{\mathcal{F}_2[mr_{x_3}(m, n)]}{\mathcal{F}_2[r_{x_3}(m, n)]}]\,. \tag{4.104}$$

A similar algorithm is used to estimate the tricepstrum

$$\hat{t}(m, n, l) = \frac{1}{m} \mathcal{F}_3^{-1}[\frac{\mathcal{F}_3[mr_{x_4}(m, n, l)]}{\mathcal{F}_3[r_{x_4}(m, n, l)]}]\,. \tag{4.105}$$

4.6.5 Examples

In the first simulation we convolved a nonminimum phase wavelet with an exponentially distributed sequence with skewness $\beta = 1$. The additive noise is white and Gaussian. The standard deviation of the noise represents 1% of the maximum amplitude of the signal. We used 6 records of 600 samples to estimate the third order cumulant. The bicepstrum was computed using equation (4.104). Figure (4.18) shows the true cepstrum of the wavelet and

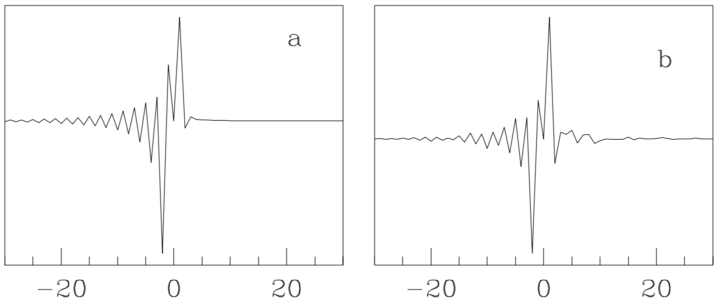


Figure 4.18: (a) Cepstrum of the true wavelet. (b) Cepstrum of wavelet derived from the bicepstrum (equations (11) and (12)).

the cepstrum extracted form the bicepstrum using. Figure (4.19) shows the true and the estimated wavelets together with the associated minimum and maximum phase components. The maximum phase component has a zero close to the unit circle that is manifested in the negative semi-axis of the cepstrum as a ‘long’ decaying oscillatory signal.

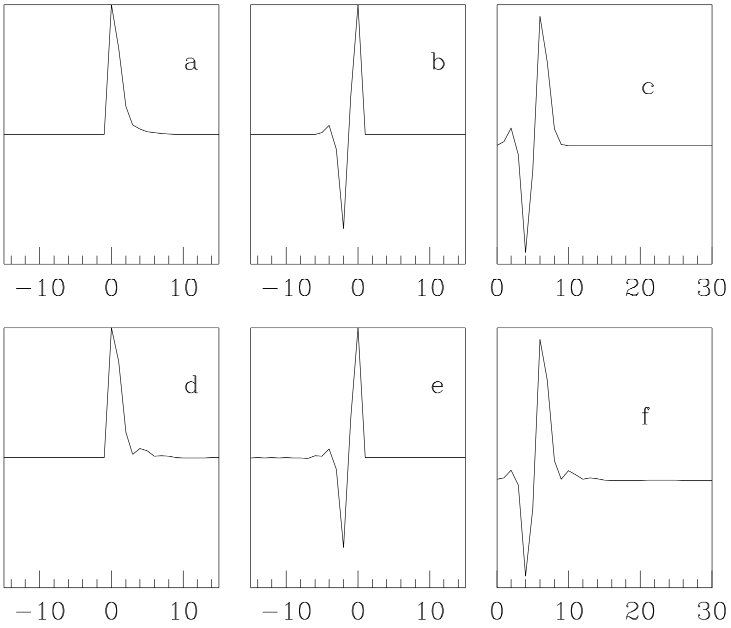


Figure 4.19: (a) and (b): True minimum phase and maximum phase components of the wavelet (c). (d), (e), and (f): Estimators of (a), (b), and (c) computed form the bicepstrum.

The same wavelet was convolved with a white, non-Gaussian, reflectivity with non vanishing kurtosis. The fourth order cumulant was computed from 4 records of 1000 samples each. The cepstrum of the wavelet was estimated from the tricepstrum. The results are shown in Figure (4.20). The time signature and its maximum and minimum phase components are displayed in Figure (4.21). The technique permits the recovery of the cepstral coefficients of the wavelet with an accuracy proportional to the accuracy of the estimation of the cumulant.

In this section we provide an heuristic analysis of the performance of the algorithm.

We simulate 20 realizations of a non-Gaussian ($\gamma \neq 0$) process which is convolved with the source wavelet portrayed in Figure (4.22). Figure (4.23) shows the wavelet retrieved from the tricepstrum. The fourth-order cumulant was estimated using 4 segments of 1000 samples each. In figures 4.24) and (4.25), we used 4 segments of 500 and 250 samples , respectively. These results indicate that a fairly good reconstruction of the amplitude and phase can be achieved for large data sets.

Figure (4.26) portrays a segment of a seismic section pre-processed for impedance inversion. The segment is composed of 24 traces of 300 samples each. The fourth order cumulant is estimated from each trace and the average cumulant is used to identify the wavelet. Figure (4.27) shows the cepstrum of the wavelet retrieved from one of the axis of the tricepstrum. The minimum and maximum phase components of the wavelet are shown in Figure (4.28).

The tricepstrum estimator of the wavelet is illustrated in Figure (4.29). For comparison we also show the estimator of the wavelet computed using a cumulant matching approach (Velis and Ulrych, 1996). The later uses a global optimization procedure (simulated annealing) to find the wavelet that best reproduces the cumulant of the data.

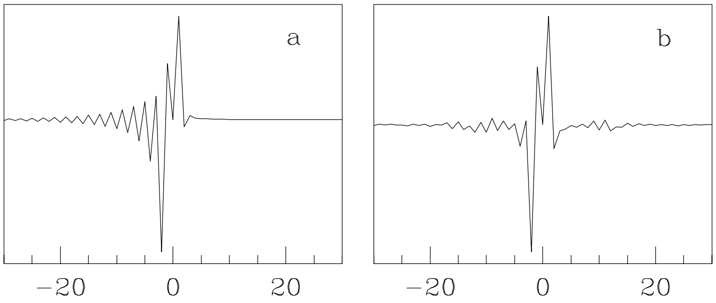


Figure 4.20: (a) Cepstrum of the true wavelet. (b) Cepstrum of wavelet derived from the tricepstrum.

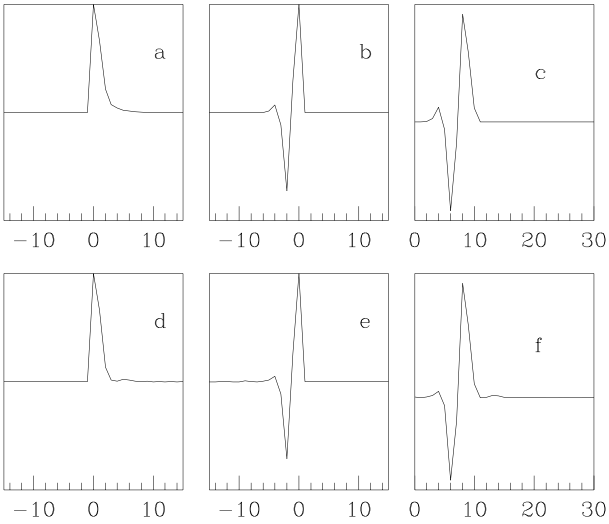


Figure 4.21: (a) and (b): True minimum phase and maximum phase components of the wavelet (c). (d), (e), and (f): Estimators of (a), (b), and (c) computed from the tricepstrum.

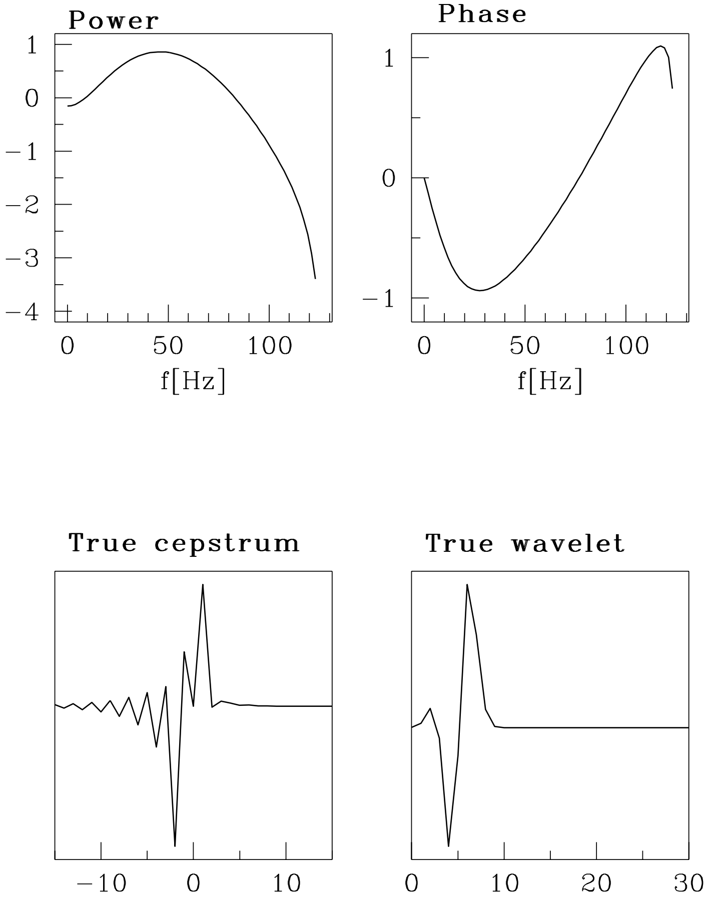


Figure 4.22: Synthetic wavelet.

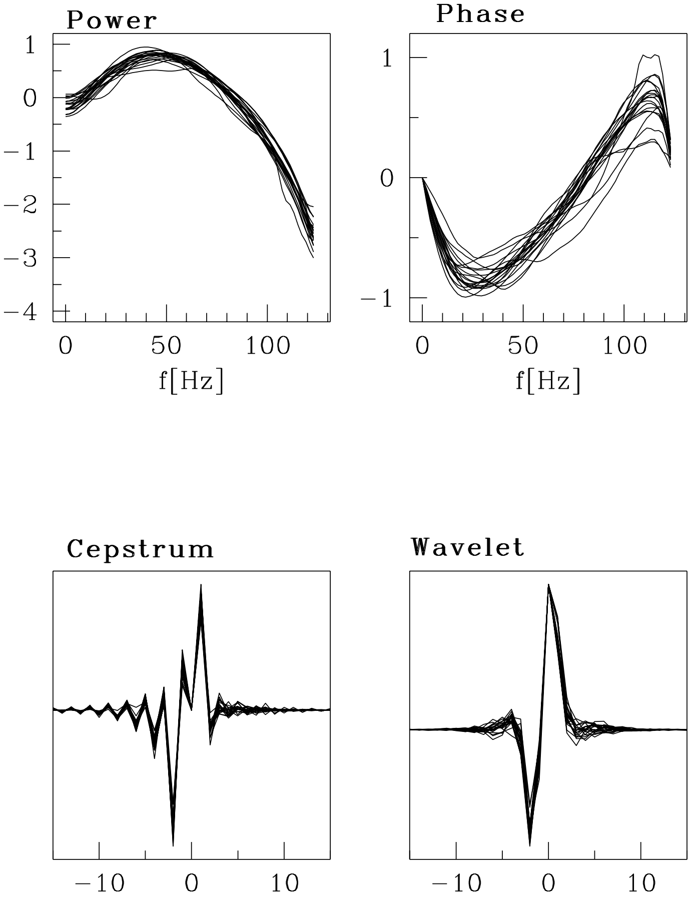


Figure 4.23: Wavelet estimation using the tricepstrum. The fourth order cumulant was estimated from 4 segments of 1000 samples each. The figures correspond to 20 realizations of the process.

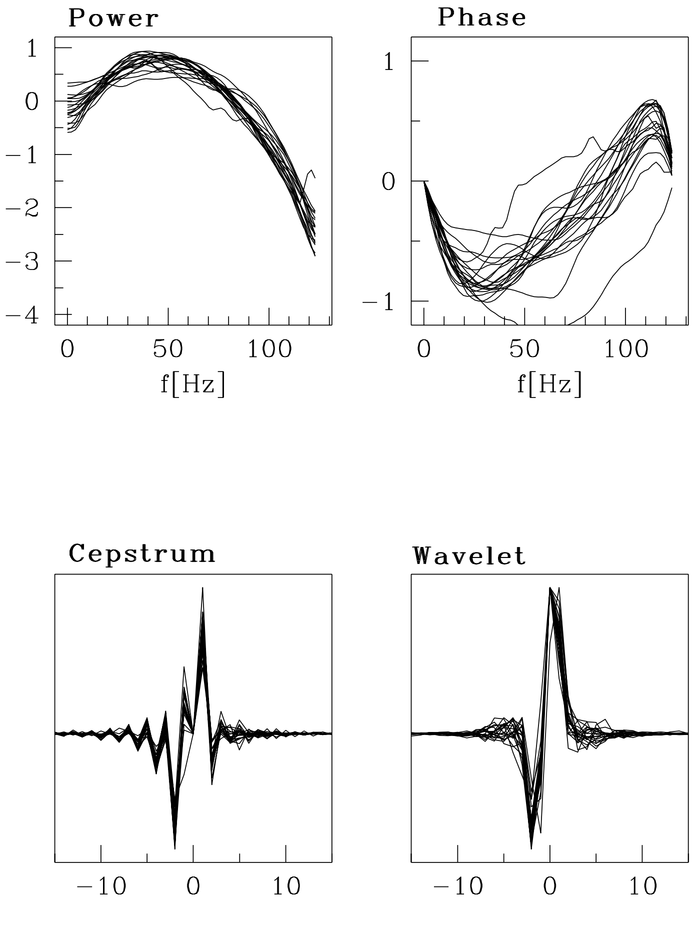


Figure 4.24: Wavelet estimation using the tricepstrum. The fourth order cumulant was estimated from 4 segments of 500 samples each. The figures correspond to 20 realizations of the process.

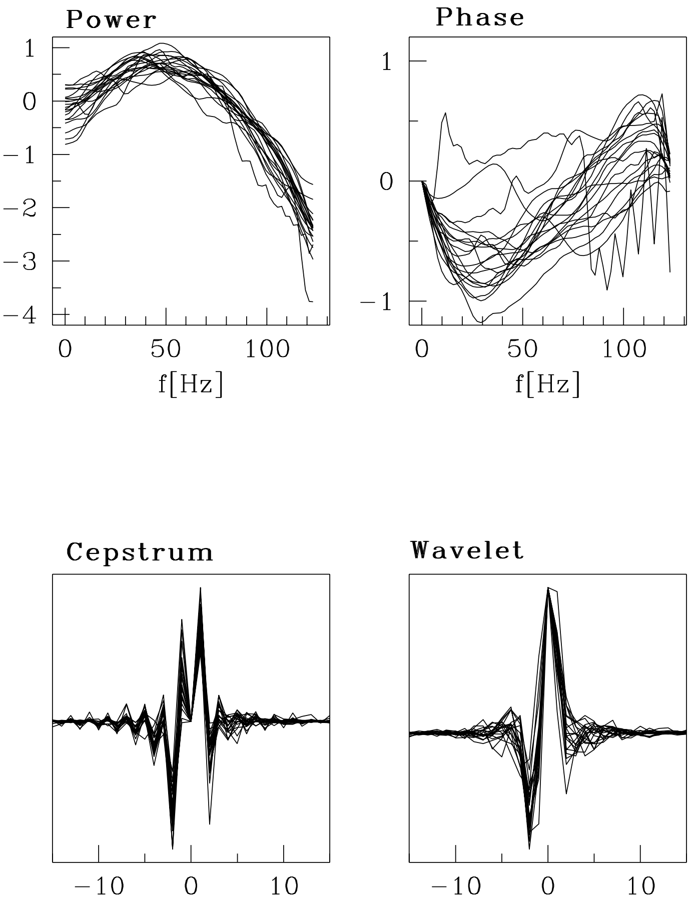


Figure 4.25: Wavelet estimation using the tricepstrum. The fourth order cumulant was estimated from 4 segments of 250 samples each. The figures correspond to 20 realizations of the process.

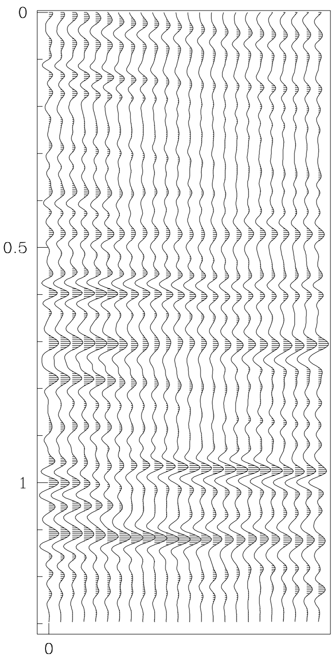


Figure 4.26: Segment of seismic section pre-processed for impedance inversion.

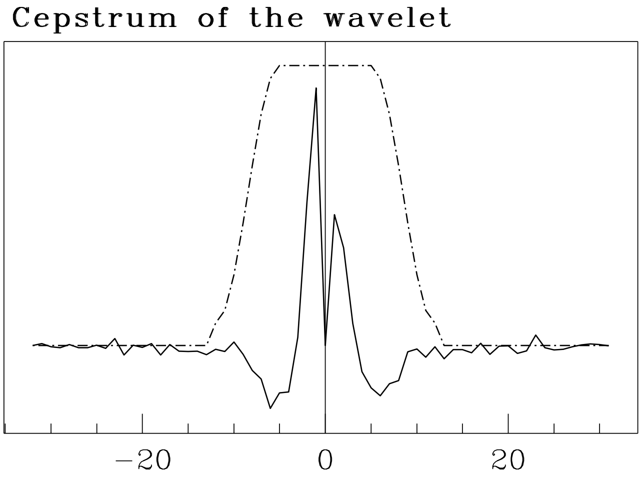


Figure 4.27: Cepstrum of wavelet estimated from the tricepstrum of the data. An average fourth-order cumulant derived from 24 traces was used to retrieve the tricepstrum.

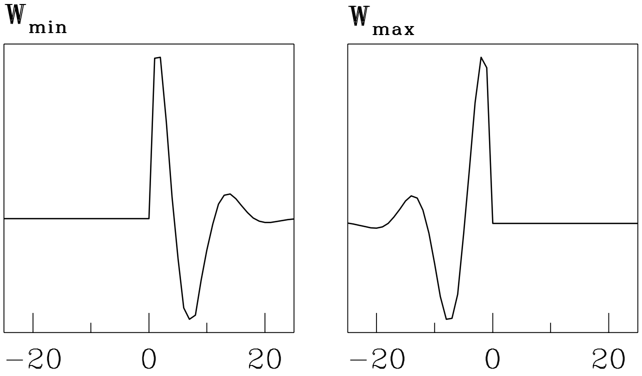


Figure 4.28: Minimum and maximum phase decomposition of the wavelet after cepstral liftering.

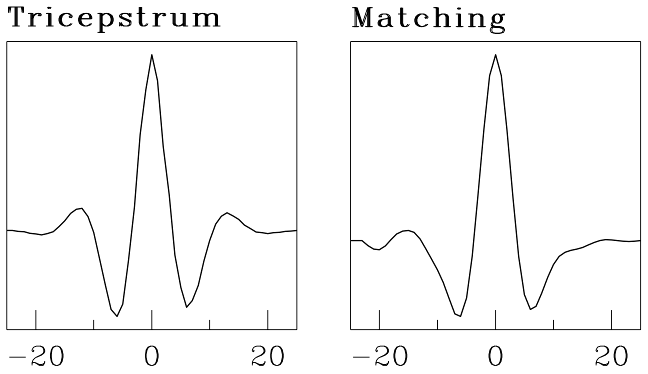


Figure 4.29: Wavelet estimates computed using the tricepstrum (left) and cumulant matching plus non-linear optimization (right).

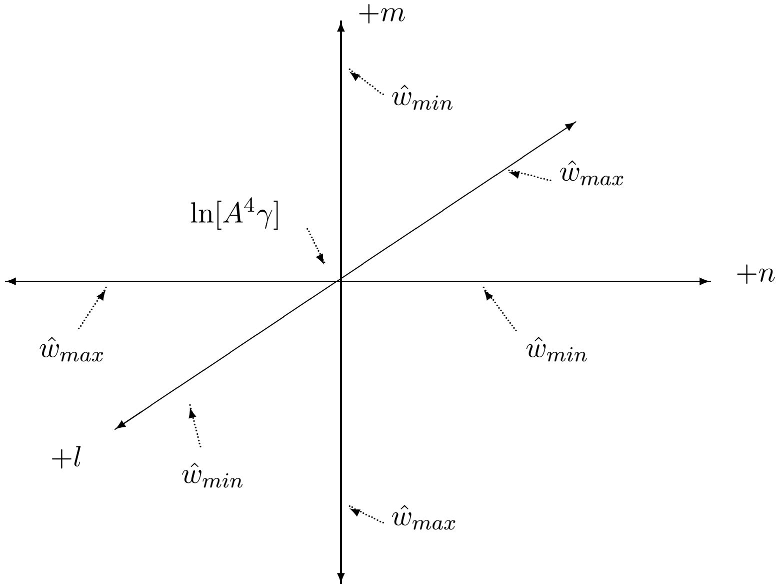


Figure 4.30: Schematic representation of the tricepstrum for a non-Gaussian MA process.

4.7 Minimum entropy deconvolution

The MED technique (minimum entropy deconvolution) proposed by Wiggins (1978), offers a different approach to seismic deconvolution. While the classical methods such as spiking and predictive deconvolution (Robinson and Treitel, 1980) seek to whiten the spectra, the MED seeks the smallest number of large spikes which are consistent with the data.

Despite these differences, both methods constitute a linear approach to seismic deconvolution. The spiking and predictive filters are obtained by inverting the Toeplitz matrix; the MED filter is calculated in an iterative procedure in which the Toeplitz matrix is inverted at each step (Wiggins, 1978). These filters are quite different in nature, but as they are linear operators none of them can handle band-limited data properly. This limitation is difficult to overcome when dealing with noisy data.

In this work a frequency-domain version of the MED scheme is developed. This approach involves maximizing a generalized entropy norm with respect to the seismic reflectivity. The particular norm in which we are going to focus our attention (the logarithmic norm) has also been used for deconvolution and wavelet extraction by Postic et al. (1980) in an attempt to overcome the limitations of the classical MED method (Wiggins, 1978).

For band-limited signals the deconvolution can be achieved by reconstruction of the reflectivity spectrum. Two main procedures have been developed to reach the latter goal. The first method (Levy and Fullagar, 1981) is based on a linear programming (LP) approach. This method attempts to find the reflectivity series with minimum absolute norm remaining consistent with the data. The second approach (Lines and Clayton, 1977) fits a complex autoregressive (AR) model to the data spectrum and from the information available in the actual band attempts to extrapolate the missing low and high frequencies of the reflectivity. Both methods have been improved and expanded to cope with acoustic impedance inversion from band-limited reflection seismograms. In the LP approach, Oldenburg et al. (1983) incorporated impedance constraints to the problem.

Our method can be mainly compared with the LP approach. This is because both methods seek for an extremum point of a given norm which is a function of the underlying unknown function: the reflectivity. The main advantage of an entropy norm over the absolute norm (l_1) is that the minimization procedure leads to an easy to handle algorithm, avoiding the computationally expensive cost of linear programming routines. It must be stressed that the proposed method provides a unifying thread between the LP (Oldenburg et al., 1983) and the MED approach (Wiggins, 1978).

4.7.1 Minimum Entropy estimators

The normal incidence seismogram model can be expressed as the convolution between two basic components: the reflectivity y_t and the wavelet w_t. If we denote the noise-free seismic trace by x_t then

$$s_t = w_t * q_t \tag{4.106}$$

where * denotes discrete convolution. The goal of the deconvolution process is to recover q_t from s_t. If we adopt a linear scheme, an operator f_t such that

$$q_t = s_t * f_t \tag{4.107}$$

must be obtained. Note that if s_t is a band-limited signal, only a part of q_t can be recovered.
Usually the signal is contaminated with noise, then the normal incidence seismogram model is

$$s_t = w_t * q_t + n_t. \tag{4.108}$$

We want to compute a filter f_t such that $f_t * w_t = \delta_t$, but usually we have an estimate of the filter $\hat{f}_t$. Then $\hat{f}_t * w_t = a_t$, where a_t is called the averaging function which in the ideal case should resemble a delta function (Oldenburg, 1981). Operating with the filter $\hat{f}_t$ on the seismic trace

$$\begin{aligned} \hat{q}_t &= a_t * q_t + \hat{f}_t * n_t & (4.109) \\ &= q_t + (a_t - \delta_t) * q_t + \hat{f}_t * n_t. & (4.110) \end{aligned}$$

Equation (4.110) shows that the filter not only has to make a_t close to a delta, but also has to keep the noise level as small as possible. It follows that we are faced with the usual trade-off between resolution and statistical reliability.

At this point it must be said that even with the best seismic field and processing techniques, the band-pass nature of the "earth system" is always present, i.e. after removing the wavelet, only a portion of the reflectivity spectrum is available. In other words $\hat{q}_t = q_t * a_t$ is band-limited. For further developments $\hat{q}_t$ will be called the band-limited reflectivity and q_t the full-band reflectivity. We will assume that the wavelet has been removed, therefore a_t has zero phase with constant amplitude in the frequency range $[\omega_L, \omega_H]$ and zero amplitude outside that interval.

Estimating the full-band reflectivity from the band-limited reflectivity is a nonunique linear inverse problem. Neglecting the noise term, the last assessment may be easily confirmed taking the Fourier transform of equation (4.102)

$$\hat{Q}(\omega) = A(\omega) \cdot Q(\omega). \tag{4.111}$$

It is easy to see that $Q(\omega)$ can take any value at those frequencies at which $A(\omega)$ vanishes. The knowledge of $\hat{Q}(\omega)$ is not enough to estimate the portion of $Q(\omega)$ outside the non-zero band of $A(\omega)$. Hence, there exists an infinite number of models q_t that satisfy equation (4.110). In other words, $\hat{Q}(\omega)$ gives no information about the parts of $Q(\omega)$ which belong to the null space. In the next analysis we will discuss how to limit the nonuniqueness of the problem.

4.7.2 Entropy norms and simplicity

Among all the possible solutions to the problem stated in the previous section, we will look for those particular solutions in which a reasonable feature of the reflectivity is reached. Usually, parsimony is a required feature of an acceptable model. "Minimum structure" or "simple solution" are terms often used for a model with parsimonious behaviour. In Wiggins' original approach, the term "minimum entropy" is used as synonymous with "maximum order". The term is appropriate to set up the main difference between MED and spiking or predictive deconvolution. While spiking or predictive deconvolution attempt to whiten the data (minimum order), the MED approach seeks for a solution consisting mainly of isolated spikes. Wiggins' entropy was inspired by factor analysis techniques, and can be regarded as a particular member of a broad family of norms of the form

$$V(\mathbf{q}) = \frac{1}{NF(N)} \sum_{i=1}^{N} q'_i . F(q'_i), \tag{4.112}$$

where the vector $\mathbf{q}$ represents the reflection coefficient series of length N, and q'_i an amplitude normalized measure given by

$$q'_i = \frac{{y_i}^2}{\sum_k {y_k}^2/N}. \tag{4.113}$$

In formula (4.112), $F(q_i)$ is a monotonically increasing function of q_i , which is often called the entropy function. Having defined $F(q_i)$, the following inequality can be established:

$$F(1)/F(N) \leq V \leq 1. \tag{4.114}$$

The normalization factor in equation (6) guarantees the same upper limit for any entropy function. Note that for the most simple case, a series with all zeros and one spike, the norm reaches the upper bound $V = 1$. When all the samples are equal V reaches the lower bound.
The original MED norm is obtained when $F(q'_i) = q'_i$. In many synthetic examples we found that this norm is very sensitive to strong reflections. To avoid such inconvenient we have tested other norms concluding that better results are achieved with the logarithmic norm in which $F(q_i) = ln(q'_i)$

4.7.3 Wiggins' algorithm

A trivial solution to the problem stated in equation (4.107) is

$$\begin{aligned} \hat{f}_t = &\quad s_t^{-1}, \\ \hat{q}_t = &\quad \delta_t. \end{aligned} \tag{4.115}$$

Where ${s_t}^{-1}$ stands for the inverse of s_i if it exists. To avoid such solution a fixed length must be imposed to the operator f_i, then

$$q_n = \sum_{l=1}^{LF} f_l . s_{n-l}. \tag{4.116}$$

The criterion for designing the operator f_k may be set as

$$\begin{aligned} \frac{\partial V}{\partial f_k} &= 0, \quad k = 1, 2, ..., LF; \\ \frac{\partial V}{\partial f_k} &= \tfrac{1}{NF(N)} \textstyle\sum_i (F(q'_i) + q'_i \tfrac{\partial F(q'_i)}{\partial q'_i}) \tfrac{\partial q'_i}{\partial f_k}. \end{aligned} \tag{4.117}$$

From equation (4.107) it follows that $\frac{\partial q_n}{\partial f_k} = s_{n-k}$, and after some algebraic manipulations equation (7.19) becomes

$$\sum_l f_l \sum_n s_{n-k} s_{n-l} = \sum_i b_i s_{i-k}, \tag{4.118}$$

where

$$b_i = \frac{G(q'_i) q_i}{\frac{1}{N} \sum_j G(q'_j) q'_j} \tag{4.119}$$

and

$$G(q'_i) = F(q'_i) + q'_i \frac{\partial F(q'_i)}{\partial q'_i}. \tag{4.120}$$

We can use two criteria $F(q'_i) = q'_i$ (Wiggins' entropy function) or $F(q'_i) = \ln(q'_i)$ (logarithmic entropy function). Numerical experiments suggests that the second one appears to be a better choise for seismic deconvolution (Sacchi et. al, 1994).

Expression (4.118) corresponds to the system used to design the well known Wiener or shaping filter (Robinson and Treitel, 1980). This filter seeks to convert an input signal $\mathbf{x}$ into a desired output $\mathbf{b}$. In matrix notation:

$$\mathbf{R}\,\mathbf{f} = \mathbf{g}(\mathbf{f}), \tag{4.121}$$

where $\mathbf{R}$ is the Toeplitz matrix of the data and the vector $\mathbf{g(f)}$ is the cross-correlation between $\mathbf{b}$ and $\mathbf{s}$. The system must be solved through an iterative algorithm:

$$\mathbf{f^{(n)}} = \mathbf{R^{-1}} \cdot \mathbf{g}(\mathbf{f^{(n-1)}}), \tag{4.122}$$

where the upper index n denotes iteration number. In each iteration the system is solved with Levinson's algorithm (Robinson and Treitel, 1980). The initial value for this system is $\mathbf{f^{(0)}} = (\mathbf{0}, \mathbf{0}, \mathbf{0}, \cdots, \mathbf{1}, \cdots, \mathbf{0}, \mathbf{0}, \mathbf{0})$. Note that in each iteration the system attempts to reproduce the features of the reflectivity series. If the proper length is chosen, the system leads to a useful maximum and the main reflections can be estimated (Wiggins, 1978).

4.7.4 Frequency domian algorithm (Sacchi et. al, 1994)

In the frequency domain, the maximization of the entropy is subjected to the following constraint:

$$Q(\omega) = \hat{Q}(\omega), \quad \omega \in [\omega_L, \omega_H]. \tag{4.123}$$

For practical purposes let us define equation (4.110) by means of the discrete Fourier transform

$$\hat{Q}_k = A_k \cdot Q_k, \tag{4.124}$$

where the lower index k denotes frequency sample. So the maximization of V with mid-band constraints can be written down as

$$\text{Maximize} \quad V, \tag{4.125}$$

subjected to

$$Q_k = \sum_{n=0}^{N-1} \hat{q}_n e^{\frac{-i2\pi kn}{N}}, \quad k = k_L, \cdots, k_H, \tag{4.126}$$

where k_L and k_H are the samples that correspond to ω_L and ω_H, respectively. It is easy to see that the mid-band $[\omega_L, \omega_H]$ must be kept unchanged throughout the algorithm.

The solution of the last problem can be achieved solving the following system of equations:

$$\frac{\partial V}{\partial Q_k + \sum_{l=k_L}^{k_H} \lambda_l \frac{\partial Q_l}{\partial Q_k}(Q_l - \sum_{n=0}^{N-1} \hat{q}_n e^{\frac{-i2\pi nl}{N}}) = 0} \tag{4.127}$$

$$Q_k - \sum_{n=0}^{N-1} \hat{q}_n e^{\frac{-i2\pi nk}{N}} = 0, \quad k = k_L, \cdots, k_H \tag{4.128}$$

where λ_l are the Lagrange multipliers of the problem. Taking the derivative, inserting Q_k in the constraint and then the multipliers in equation (18), the following result is obtained:

$$Q_k = \begin{cases} \sum_{n=0}^{N-1} \frac{G(q'_n) q_n e^{-i2\pi kn/N}}{\sum_j G(q'_j) q'_j / N}, & k \notin [k_L, k_H]; \\ \hat{Q}_k, & k \in [k_L, k_H]. \end{cases}$$

From equations (4.119) and (4.123) it is easy to see that

$$Q_k = \begin{cases} B_k, & k \notin [k_L, k_H]; \\ \hat{Q}_k, & k \in [k_L, k_H], \end{cases} \tag{4.129}$$

where B_k is the discrete Fourier transform of b_t (4.119). Because b_t is a nonlinear function of q_t, the problem must be solved as follows:

1. The algorithm is initialized by letting $q_t = \hat{q}_t$

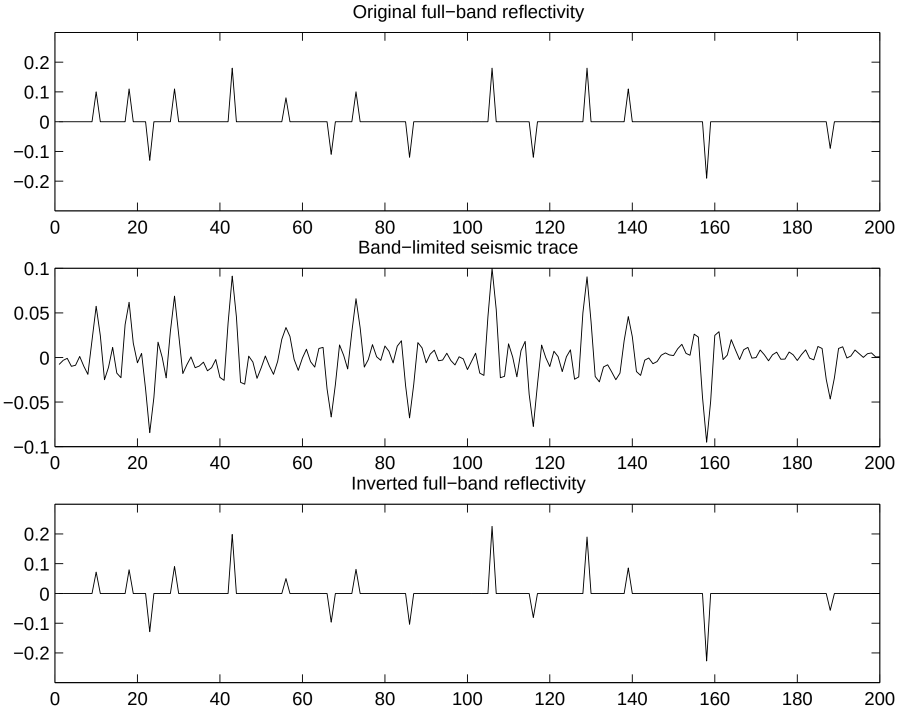


Figure 4.31: True reflectivity (top), band-limited reflectivity (center), and inverted reflectivity using FMED (bottom)

2. b_t and B_k are computed.

3. The missing low and high frequencies are replaced by B_k.

4. From the inverse Fourier transform, an estimate of the reflectivity is calculated. The norm V is also evaluated to check convergence and a new iteration starts in step 2.

The algorithm, as we can see, is very simple and can be efficiently coded used the FFT.

The subroutine to run FMED

We have developed a FMED library to recosntruct missing low and high frequencies after zero-phase deconvolution[5].
The subroutine `fmed` is called as follows:

```
subroutine FMED(s,ns,fl,fh,dt,nfft,cw1,cw2,iter_max,th,io,fmax
```

where the following parameters are provided to the subroutine:
`s(n1)` A seismic trace of `ns` samples. The trace is a band-limited version of the reflectivity
`fl,fh` min and max frequency that define the band-limited reflectivity (in Hz).
`dt` sampling rate in sec.
`nfft` Number of freq. samples for the fft
`cw1(nfft),cw2(nfft)` pre-computed tables of exponentials for the FFT. See subroutine `fftcw`.
`iter_max` Maximum number of iterations (≤ 10)
`th` Values of reflectivity below `th` are set to zero , if you doubt `th=0`.
`io` If `io.eq.0` the algorithm completes all the high frequencies up to the the Nyquist frequency. If `io.ne.0` the algorithm will complete the high frequencies up to `fmax`.
`fmax` maximum frequency to complete when `io.ne.0`

[5] Let me know if you are interested in a copy of this library

4.8 References

Bednar, J. B., and Watt, T. L., 1985, Calculating the complex cepstrum without phase unwrapping or integration, *Proc. IEEE*, vol. ASSP-33, No.4,

Huber, P. J, 1981. Robust Statistics, John Wiley and Sons, Inc.

Levy, S., and Fullagar, P. K., 1981, Reconstruction of a sparse spike train from a portion of its spectrum and application to high-resolution deconvolution: Geophysics, v.46, p.1235-1243.

Lines, L. R., and Clayton, R. W., 1977, A new approach to Vibroseis deconvolution: Geophys. Prosp., v.25., p.417-433.

Nikias C. L. and Raghuveer, M. R., 1987, Bispectrum estimation: a digital signal processing framework, *Proc. IEEE*, Vol.75, No.7, 869-891.

Oldenburg, D. W., Scheuer, T., and Levy, S., 1983, Recovery of the acoustic impedance from reflection seismograms: Geophysics, v.48, p.1318-1337.

Robinson, E. A. & Treitel, S., 1980. Geophysical signal analysis, Prentice-Hall, Inc.

Sacchi, M.D. & Ulrych,T.J, 1995. Improving resolution of Radon operators using a model re-weighted least squares procedure, *Journal of Seismic Exploration*, **4**, 315-328.

Sacchi,M.D., Velis D., & Cominguez, A.H., 1994, Frequecny domain Minimum Entropy deconvolution: Geophysics, vol. 59, No. 6.

Scales, J. A., Gersztenkorn, A. & Treitel, S., 1988. Fast l_p solution of large, sparse, linear systems: Application to seismic travel time tomography, *Journal of Comp. Phys.*, **75**, 314-333.

Scargle, J. D., 1977, Absolute value optimization to estimate phase properties of stochastic time series, *IEEE Trans. Inform. Theory*, vol.IT-23, 140-143.

Ulrych, T. J., 1971, Application of homomorphic deconvolution to seismology, *Geophysics*, vol.39, 650-660.

Velis D. R., and Ulrych T. J., 1996, Simulated annealing wavelet estimation via fourth-order cumulant, *Geophysics*, vol.61, 1939-1948.

Wiggins, R. A., 1978, Minimum entropy deconvolution: Geoexpl., v.16, p.21-35.

Chapter 5

Signal-to-noise-ratio Enhancement

5.1 FX filters

Signal-to-noise-ratio enhancemnet in the FX domain has been proposed by Canales (1994) as a method for random noise attenuation. The technique is widely accepted and used in the industry. The method is very effective in attenuating randon noise, it is easy to implement and very efficient in the computational sence.

Signal predictability has been extensively studied in the context of AR filter and in harmonic retrival via ARMA models (see for instance, Ulrych and Clayton, 1976).

The idea is quite simple, and can be summiraized as follows. In the FX domain linear events or quasilinear events manisfest themselves as a superposition of harmonics. If noise is taken into account, an optimal model to predict a superposition of harmonics is an ARMA model. However, given the fact that ARMA models might not be very stable (they involve the solution of an eigenvalues problem), we will propose to replace the ARMA model by a long AR filter. In this case, the predicbility is not optimal, but the problem can be easily solved using predictor error filters of the type we have already analyzed in the context of deconvolution.

We will start this lecture by introducing the concept of predictivilty via a very simple model composed of a superposition of harmonics.

Later we will discuss a novel approach proposed by Soubaras (1994, 1995) which is based on the concept of quasi-predictivilty.

5.1.1 The signal model

The signal model is based on the assumption that seismic data can be represented as a superposition of events with linear moveout. In general, a seismic section can be divided into overlapping windows where this assumption is valid.

We first consider a seismic section that consists of a single waveform. The frequency domain representation of $s(f, x)$ is given by

$$S(f,x) = A(f)\, e^{i2\pi f\theta x}\,, \tag{5.1}$$

where $A(f)$ indicates the source spectrum, f the temporal frequency, x the spatial variable or offset and θ the apparent slowness along x. We will assume that the spatial variable x is regularly discretized according to $x = (k-1)\Delta x$, $k = 1 : N$. For any temporal frequency, f, we can write

$$S_n = A\, e^{i\alpha n}, \qquad n = 1, N \tag{5.2}$$

where $\alpha = 2\pi f\theta\Delta x$. The following recursion is obtained by combining S_n and S_{n-1}

$$S_n = a_1 S_{n-1}\,. \tag{5.3}$$

where $a_1 = exp(i\alpha)$. The last equation is a first order difference equation that allows us to recursively predict the signal along the spatial variable x. Similarly, it can be shown that the superposition of p complex harmonics (p linear events in $x - t$) can be represented by a difference equation of order p

$$S_n = a_1 S_{n-1} + a_2 S_{n-2} + \ldots a_p S_{n-p}\,. \tag{5.4}$$

The last equation can be written in prediction error form as follows

$$\sum_{k=0}^{p} g_k S_{n-k} = 0\,, \tag{5.5}$$

where the coefficients of the prediction error filter are related to the coefficients a_k in equation (4) by the following expressions

$$g_0 = 1, \quad g_k = -a_k, \quad k = 1, p\,.$$

So far we have been able to define a recursive expression to predict a noise-free superposition of complex harmonics. In real applications, however, additive noise will corrupt the data

$$Y_n = S_n + W_n\,, \tag{5.6}$$

where W_n represents a white noise sequence. Substituting $S_{n-k} = Y_{n-k} - W_{n-k}$ into equation (6.19) leads to the following system of equations that defines the signal model in terms of the prediction error filter

$$\begin{aligned} \textstyle\sum_{k=0}^{p} g_k Y_{n-k} &= \textstyle\sum_{k=0}^{p} g_k W_{n-k} \\ &= e_n\,. \end{aligned} \tag{5.7}$$

The latter is an ARMA(p,p) process in which the AR and MA components are identical. The signal e_n in equation (5.7) designates the non-white innovation sequence $\sum_{k=0}^{p} g_k W_{n-k}$.

5.1.2 AR FX Filters

Rather than trying to solve the ARMA equations one can replace the ARMA model by a long AR (autoreregressive) model:

$$Y_n - f_1 Y_{n-1} + f_2 Y_{n-2} \ldots + f_p Y_{n-p} = W_n\,. \tag{5.8}$$

Where f_k are the coefficients of the AR(p). The parameter p is the order of the model, which in this case should be large enough to represent the original ARMA model. This kind of model has been extensively studied in the spectral analysis enviroment and in regression analysis. The determination of p is not an easy task, and demands some statistical criterion. Akaike (1974) proposed a criterion for order selection, the AIC (Automatic infromation criterion also known as Akaike information criterion).

The last eqaution can be writen in matrix form as follows (assume $p = 3$),

$$\begin{pmatrix} y_1 & 0 & 0 \\ y_2 & y_1 & 0 \\ y_3 & y_2 & y_1 \\ y_4 & y_3 & y_2 \\ 0 & y_4 & y_3 \\ 0 & 0 & y_4 \end{pmatrix} \begin{pmatrix} f_1 \\ f_2 \\ f_3 \end{pmatrix} - \begin{pmatrix} y_2 \\ y_3 \\ y_4 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} w_2 \\ w_3 \\ w_4 \\ 0 \\ 0 \end{pmatrix} . \tag{5.9}$$

We wave already seen a similar equation in the previous chapter when we analysze inverse filters. This is a convolution matrix.
Last equation can be written as

$$\mathbf{Y}\mathbf{f} - \mathbf{d} = \mathbf{w} \tag{5.10}$$

The least squares filter $\mathbf{f}$ is computed by minimizing the power of the innovation $\mathbf{w}$, in this case our cost or objective funcion is given by:

$$J = ||\mathbf{Y}\,\mathbf{f} - \mathbf{d}||^2 \,. \tag{5.11}$$

Taking derivatives of the cost function with respect and equating the result to zero leads to

$$\mathbf{Y}^H\,\mathbf{Y}\,\mathbf{f} \;=\; \mathbf{Y}^H\,\mathbf{d} \tag{5.12}$$

Note that the matrix $\mathbf{Y}^H\,\mathbf{Y}$ is a Toeplitz form which can be efficiently solved using Levinson's recursion. Then, the estimated filter $\hat{\mathbf{f}}$ is given by

$$\hat{\mathbf{f}} \;=\; (\mathbf{Y}^H\,\mathbf{Y})^{-1}\,\mathbf{Y}^H\,\mathbf{d} \,. \tag{5.13}$$

Once the filter has been estimated we apply it to the data vector $\mathbf{d}$ to obtain the "clean" data vector $\hat{\mathbf{d}}$

$$\hat{\mathbf{d}} \;=\; \mathbf{Y}\,\hat{\mathbf{f}} \,. \tag{5.14}$$

The estimate $\hat{\mathbf{d}}$ is the predicted data, the predicted noise sequence is given by:

$$\hat{\mathbf{w}} = \hat{\mathbf{d}} - \mathbf{d}\,. \tag{5.15}$$

In general we need to regulirize the filter by adding a small perutbation to the diagonal of the Toeplitz matrix,

$$\hat{\mathbf{f}} = (\mathbf{Y}^H\,\mathbf{Y} + \mu\mathbf{I})^{-1}\,\mathbf{Y}^H\,\mathbf{d}\,. \tag{5.16}$$

5.1.3 Data resolution matrix

We will analyze the performance of the filter by means of the Singular Value Decompistion (SVD) of the matrix. We will first consider the system given by $\mathbf{Y}\,\mathbf{f} = \mathbf{d}$ and the SVD decompostion of the convolution matrix $\mathbf{Y}$

$$\mathbf{Y} = \mathbf{U}\,\mathbf{\Sigma}\,\mathbf{V}^H\,, \tag{5.17}$$

where $\mathbf{U}$ and $\mathbf{V}$ are matrices containing the eigenvectors of $\mathbf{Y}\,\mathbf{Y}^H$ and $\mathbf{Y}^H\,\mathbf{Y}$, respectively. The diagonal form $\mathbf{\Sigma}$ is the matrix of singular values. $\mathbf{U}$ and $\mathbf{V}$ are othogonal matrices of eigenvectors and therefore the satisfiy the following equations:

$$\mathbf{U}^H\,\mathbf{U} = \mathbf{I}\,, \quad \mathbf{V}^H\,\mathbf{V} = \mathbf{I}\,. \tag{5.18}$$

The SVD solution to $\mathbf{Y}\,\mathbf{f} = \mathbf{d}$ is given by

$$\begin{aligned}\hat{\mathbf{f}} &= (\mathbf{Y}^H\,\mathbf{Y})^{-1}\mathbf{Y}^H\,\mathbf{d} \\ &= \mathbf{V}\,\mathbf{\Sigma}^2\,\mathbf{Y}^H\,\mathbf{U}\,\mathbf{\Sigma}\,\mathbf{V}^H\,\mathbf{d}\,.\end{aligned} \tag{5.19}$$

The predicted data $\hat{\mathbf{d}}$ can be written as follows:

$$\begin{aligned}\hat{\mathbf{d}} &= \mathbf{Y}\,\mathbf{d} \\ &= \mathbf{U}\,\mathbf{U}^H\,\mathbf{d}\,.\end{aligned} \tag{5.20}$$

The operator $\mathbf{U}\,\mathbf{U}^H$ is called the Data Resolution Matrix. The data resoltion matrix gives and input/output relatioship for our signal-to-noise-ratio enhancement problem.
We will come back to similar concepts when we look at the signa-to-noise-enhancement problems by means of the KL transform.

5.1.4 The convolution matrix

We have adopted a very simple convolution matrix in order to design our filter via the Levinson's recursion. But bear in ming that other data matrices can be used to estimate the data prediction filter.
Canales (1984) original formulation uses the following model

$$\begin{pmatrix} y_1 & 0 & 0 \\ y_2 & y_1 & 0 \\ y_3 & y_2 & y_1 \\ y_4 & y_3 & y_2 \\ 0 & y_4 & y_3 \\ 0 & 0 & y_4 \end{pmatrix} \begin{pmatrix} f_1 \\ f_2 \\ f_3 \end{pmatrix} = \begin{pmatrix} y_2 \\ y_3 \\ y_4 \\ 0 \\ 0 \end{pmatrix} . \tag{5.21}$$

Ulrych and Clayton (1976) proposed the transiient-free convolution matrix. This is matrix where zero extensioned is avoided. In our simple example ($p = 3$) the transient-free matrix formulation is given by

$$\begin{pmatrix} y_3 & y_2 & y_1 \\ y_4 & y_3 & y_2 \\ y_5 & y_4 & y_3 \\ y_6 & y_5 & y_4 \end{pmatrix} \begin{pmatrix} f_1 \\ f_2 \\ f_3 \end{pmatrix} = \begin{pmatrix} y_4 \\ y_5 \\ y_6 \\ y_7 \end{pmatrix} . \tag{5.22}$$

The solution of the above system gives a filter free of truncation errors. However, the martic $\mathbf{Y}^H\,\mathbf{Y}$ is not longer a Toepiltz form.
It is important to mention that the above analysis only involved forward prediction filters. In other words we are trying to predict the fututre samples of signal based upon past values the signal. A more sophyisticated schemme involves the simultaneous minimization of a forward and a backward prediction error. In this case, one can show that the system of equations in the

transient-free case) has the following aspect (we assume a filter of lenght p and a signal composed of N samples)

$$\begin{pmatrix} y_p & \cdots & y_1 \\ . & \cdots & . \\ . & \cdots & . \\ . & \cdots & . \\ y_{N-1} & \cdots & y_{N-p} \\ y_2^* & \cdots & y_{p+1}^* \\ . & \cdots & . \\ . & \cdots & . \\ . & \cdots & . \\ y_{N-p+1}^* & \cdots & .y_N^* \end{pmatrix} \begin{pmatrix} f_1 \\ f_2 \\ . \\ . \\ . \\ f_p \end{pmatrix} = \begin{pmatrix} y_{p+1} \\ . \\ . \\ . \\ y_N \\ y_1^* \\ . \\ . \\ . \\ y_{N-p}^* \end{pmatrix} . \tag{5.23}$$

5.1.5 Examples

In Figures 5.1 and 5.2 two synthetic simulatations where we examine the predictiviliy of a single ahrmonic component in the time or spcae domian. Two dimensional sumulations are displayed in Figures 5.3 and 5.4.
The algorithm to peform the FX noise attenuation is summirized as follows:

1. transform the data into the FX domain

$$Data(t, x) \rightarrow Data(f, x)$$

2. for each frequecy f solve the AR prediction problem ourline in the precedoing section to estimate the AR prediction filter.

3. Apply the filter to the data (convolution of the filter witht the data).

4. Transform back to TX

$$Data(f, x) \rightarrow Data(t, x)$$

5. end

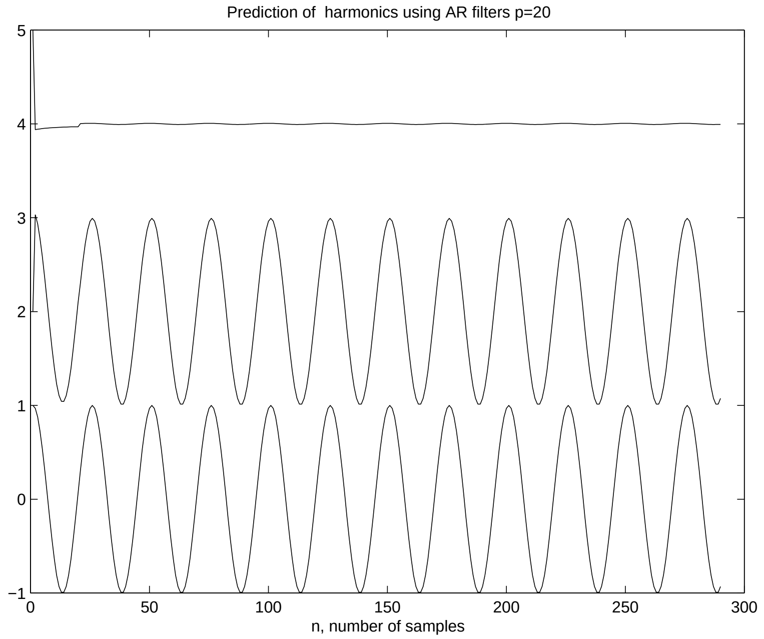


Figure 5.1: Prediction of a single harmonic (no noise) using AR filters

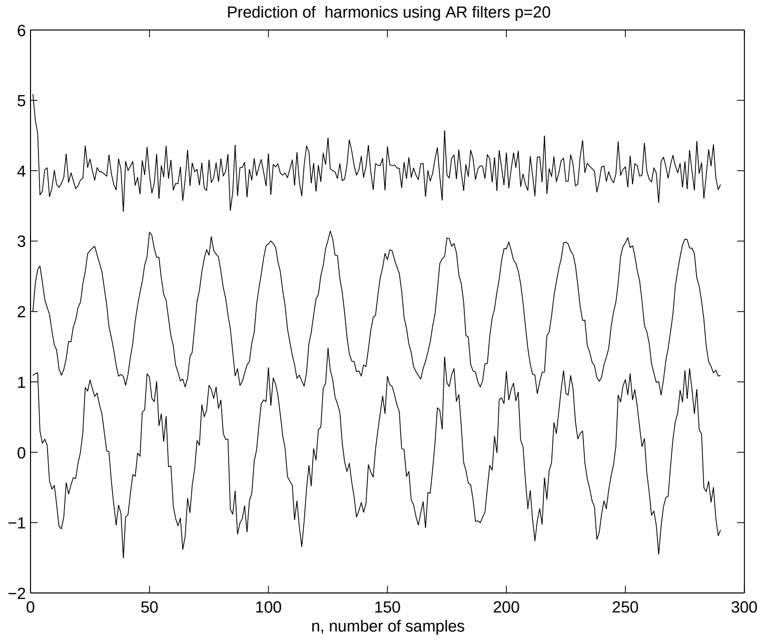


Figure 5.2: Prediction of a single harmonic ($\sigma_{noise} = 0.2$) using AR filters

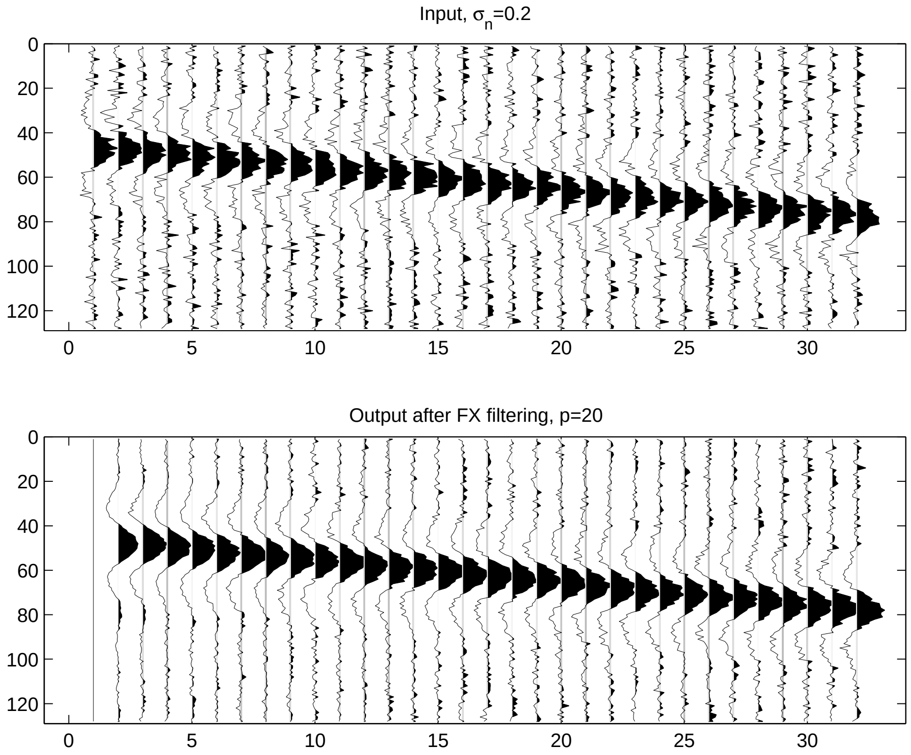


Figure 5.3: FX filtering of a single linear event inmersed in noise

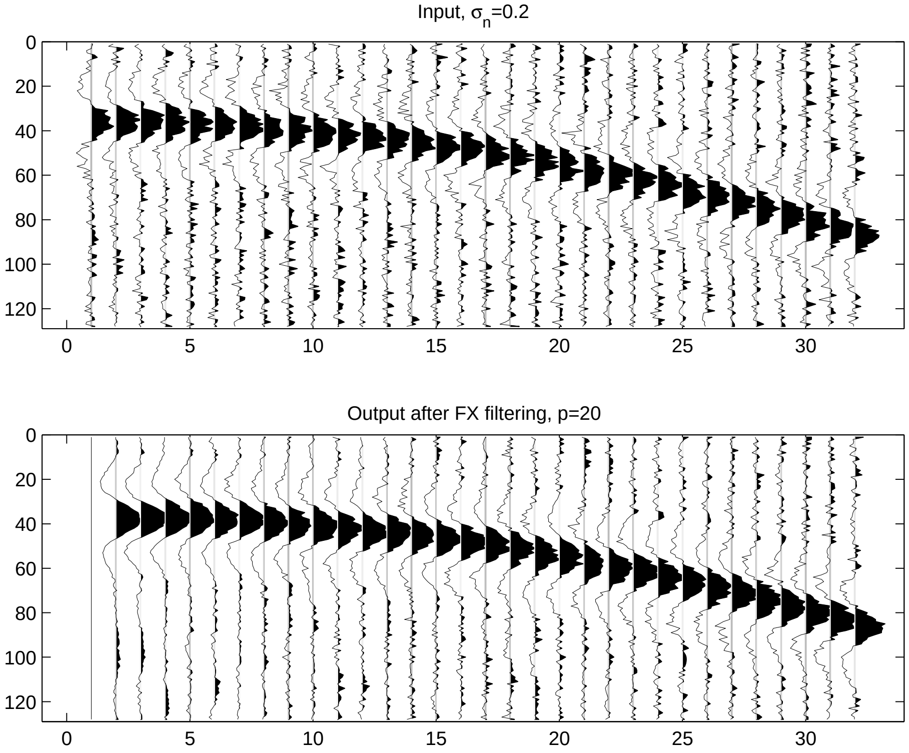


Figure 5.4: FX filtering of a single linear hyperbolic event inmersed in noise

5.1.6 Non-linear events: Chirps in $f - x$?

What happends when we do not have linear events and, therefore ARMA models. Let's study the problem of finding a recursion for a single event with parabolic moveout:

$$s(t,x) = a(t) * \delta(t + qx^2)\,. \tag{5.24}$$

In $f - x$ we can write

$$S_n = A \exp(\beta n^2) \tag{5.25}$$

where the parameter $\beta = 2\pi f q \Delta x^2$ is the coefficient of the chirped signal. In this case a simple recursion via a difference equation is not possible, yet we can combine S_n S_{n-1} and S_{n-2} to obtain the following recursive relation

$$S_n = \frac{bS_{n-1}^2}{S_{n-2}} \tag{5.26}$$

The non-linearity is evident. This type of signals cannot be optimally modelled via ARMA or AR models. So far, we have been unable to find a recursion (non-linear) in the $f - x$ domain for a superposition of parabolic events immersed in white noise.

5.1.7 Gap filling and recovery of near offset traces

The recovery of near offset traces is an important aspect in many procedures associated with seismic data processing. Prediction of missing traces in the frequency-offset domain depends on the assumption of linear moveout. When the signal exhibits non-linear moveout the $f - x$ domain is no longer composed of a finite set of harmonics, but is in fact, composed of a superposition of "chirped" signals, which makes conventional autoregressive prediction a suboptimal solution. Specifically, for a single signal with parabolic moveout a non-linear recursion would be required. The latter is a very simple scenario, yet it provides an interesting framework to analyse the complexity of the problem.

We have developed a gap filling technique which, although based on linear prediction ideas, has shown considerable promise in the resolution of this difficult problem.

This technique was first used by Wiggins and Miller (1972) for the prediction of glitches in earthquake records. Subsequently, Walker and Ulrych (1983) and Fahlman and Ulrych (1982) used this approach for band-limited impedance recovery and prediction of astronomical data, respectively.

In this section we present a suboptimal solution to the problem of gap filling. First, we will assume that the data in $f-x$ can be modelled by an AR model of length P. In principle, we will assume that the AR parameters are known. In this case, we can turn the gap filling problem into an inverse problem. The prediction error for the AR model in terms of the AR coefficients is given by

$$E_n = g_0 Y_n + g_1 Y_{n-1} + g_2 Y_{n-2} + \dots g_p Y_{n-p} \tag{5.27}$$

Since the gap comprises the traces $Y_{n_1}, Y_{n_1+1}, \dots Y_{n_2}$, we will minimize the squared sum of the prediction error within the gap

$$\sum_{k=n_1}^{n_2} E_k {E_k}^* \tag{5.28}$$

Taking derivatives with respect to the samples in the gap we end up with the following system

$$\mathbf{R}_g (Y_{n_1} Y_{n_1+1} \dots Y_{n_2})^T = \mathbf{b} \tag{5.29}$$

where $\mathbf{R}_g$ is the autocorrelation matrix of the prediction error filter and $\mathbf{b}$ is a vector that depends on the data outside the gap and the PEO and is known. The matrix $\mathbf{R}_g$ is of Toeplitz form, consequently a fast solver like the Levinson recursion can be adopted.

So far we have assumed that the PEO is known. In diverse synthetic and field data experiments we have found that a good strategy is to estimate the PEO from consecutive traces to the left and to the right of the gap. The PEO that is utilized in the gap filling is the average PEO. A first pass can be used to fill the gap and a second pass, using the complete data, can be used to recalculate the PEO and the traces within the gap.

Real data experiments are portrayed in Figures (5.6) and (5.7). In this case we have tested the algorithm with two land data shots. It is convenient to balance the energy of each trace to a common value before proceeding with the gap filling algorithm. In these examples the traces $49, 50, \ldots, 56$ are missing. A 15 points PEO was used and a regularization parameter that represents 1% of the trace of the autocorrelation matrix of the PEO was inserted to stabilize the inversion. The stabilization will minimize the energy brought into the gap, in other words the recovered amplitudes within the gap may have to be balanced.

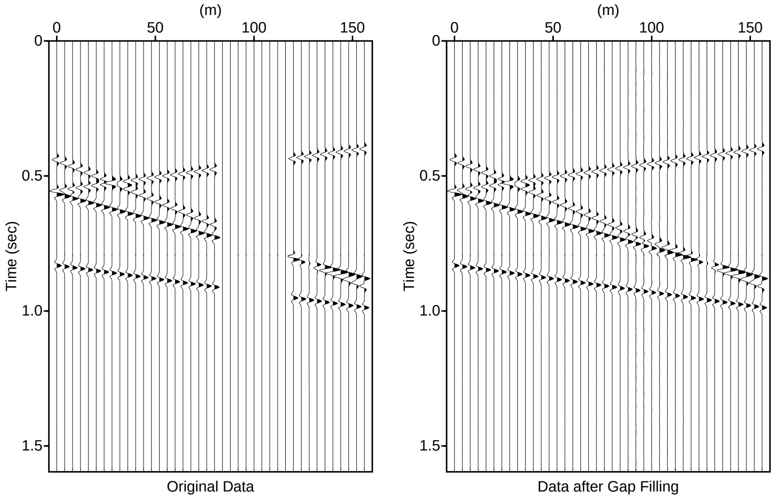


Figure 5.5: Predictivility of linear events. The AR gap filling tehcniques is used to estimate a PEO wich is latter utilized to fill the gap.

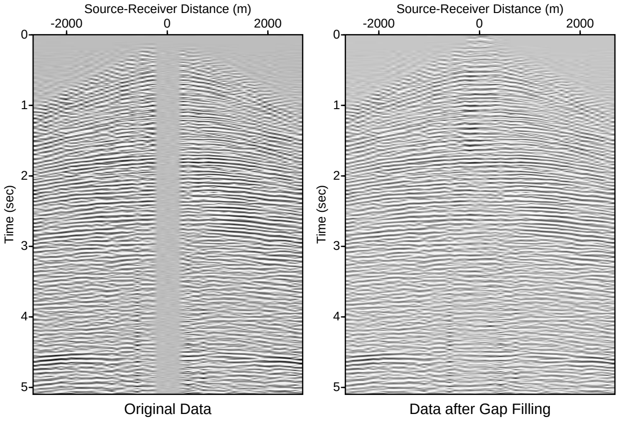


Figure 5.6: oz01.dat Original (left), after gap filling with a PEO(3)

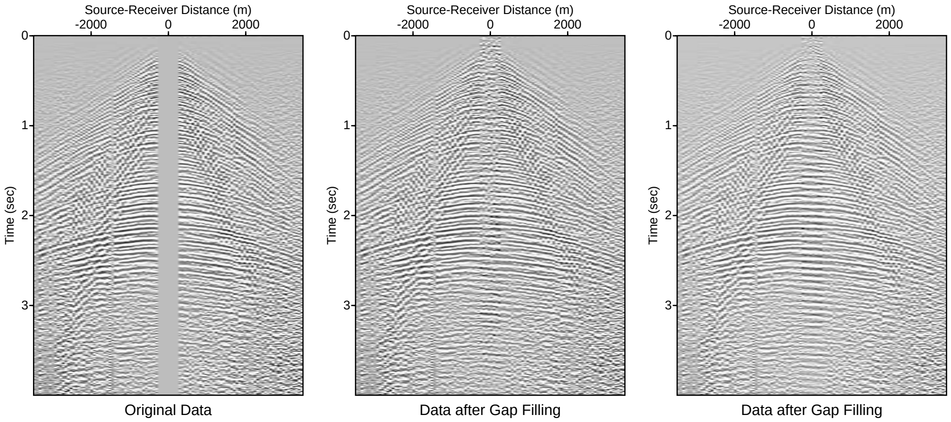


Figure 5.7: oz09.dat Original (left), after gap filling with a PEO(5) and PEO(15)

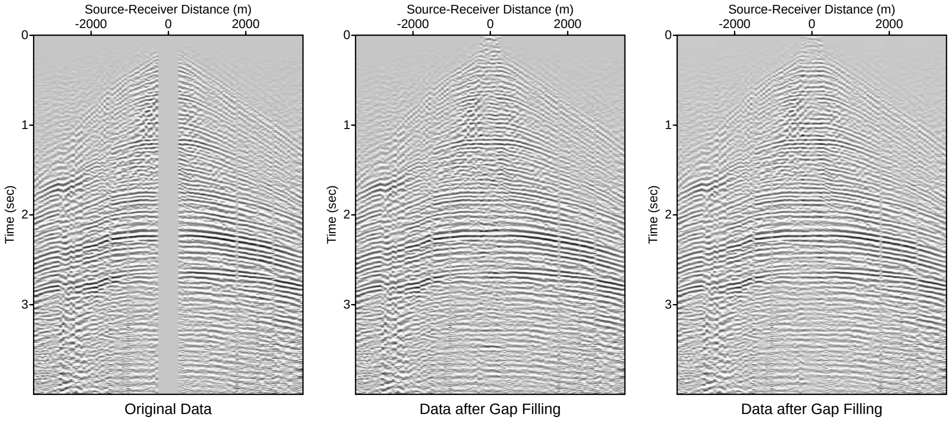


Figure 5.8: oz11.dat Original (left), after gap filling with a PEO(5) and PEO(15)

5.1.8 Pre-stack surface consistent FX filters

So far we have analyzed the problem of designing a 1D predictor filter in the FX domain for signal-to-noise-ratio enhancement. It in important to mention that a pre-stack version of the FX deconvolution was proposed by Xi-Shuo Wang at the CSEG convention in 1995. The idea is to perform N-dimensional prediction filterting in order to take into account that there is a commonality of sources, recievers, CMPs and midpoint positions in the pre-stack data volume.
Let assume that the data can be represented as follows:

$$D(s,r,h,m,f) \;\; s=1:NS,\, r=1:NR,\, h=1:NH,\;\, m=1:NM$$

where

- s is the source index
- r is the receiver index
- h is the offset
- m is the midpoint index
- f indicates the frequency

In this case the $F-SRHM$ filter is a $4-D$ operator that attemps to predict the data in the follwoing fashion:

$$\hat{D}(s,r,h,m) = \sum_i \sum_j \sum_l \sum_l F(i,j,k,l,f) D(s-i,r-j,h-k,m-l)$$

The filter is derived by minimizing the following cost function

$$J = \sum_{i,j,k,l} (D(s,r,h,m) - \sum_i \sum_j \sum_l \sum_l F(i,j,k,l,f) D(s-i,r-j,h-k,m-l))^2$$

The system of normal equations cannot be formed (it's huge!!). Therefore, a solver like Conjugate Gradients or Gauss-Seidel is needed. In this case, the system can be inverted *in the flight* (without forming the matrix).

5.2 FX Projection Filters

In this section we will study the problem of quasi-predictivity and the solution via projection filters. This part of my notes are based on the work presented by Soubaras at the SEG (1994, 1995) and EAEG (1995).

5.2.1 Wavenumber domain formulation

As we have already mentioned FX prediction is a technique proposed by Canales (1984) to attenuate Random noise in seismic section. In small windows, the seismic data can be approximated by a superposition of linear events. The later gives rise to harmonic models in the FX domain. Soubaras introduces the idea of quasi-predictibility to attenuate additive noise in the FX domain.
We denote a signal in the FX domain as $s(f, x)$

$$s(f, x) = s_0(f, x) + n(f, x)$$

where $S_0(f, x)$ is the noiseless signal and $n(f, x)$ the additive noise. Let consider the spatial prediction problem for a fix frequency f . In the wavenumber domain equation (5.2.1) can be written as

$$S(k) = S_0(k) + N(k)$$

where I have omitted f for simplicity. Let assume that $S_0(k)$ is a signal that can be predicted using a PEO operator, $A(k)$:

$$A(k)S_0(k) = I(k) \tag{5.30}$$

where $I(k)$ is the prediction error (innovation term if you think in an AR model). If $A(k)$ is a known PE-filter we can construct the following cost function to find the noise sequence

$$J = |N(k)|^2 + \mu|I(k)|^2 \tag{5.31}$$

J combines two wishes: additive noise attenuation and predictability.

$$J = |S(K) - S_0(k)|^2 + \mu|A(k)S_0(k)|^2 \tag{5.32}$$

note that if $\mu = 0$ we get the trivial solution $N(k) = 0.$ and $S_0(k) = S(k)$ where noise attenuation is not achieved.
Minimizing J (taking derivatives with respect to $S_0(k)$ and setting them to zero yields):

$$S_0(k) = \frac{S(k)}{1 + \mu A(k)A(k)^*} \tag{5.33}$$

the noise term becomes

$$N(k) = S(k) - S_0(k) = \frac{A(k)A(k)^*}{\mu^{-1} + A(k)A(k)^*} S(k) \tag{5.34}$$

equation (5.34) is Soubaras's expression for the projection filter $M(f)$ where $N(k) = M(k)S(k)$.

$$M(k) = \frac{A(k)A(k)^*}{\mu^{-1} + A(k)A(k)^*} \tag{5.35}$$

1. when $\mu = \longrightarrow 0$

$$N(k) = 0, \quad S_0(k) = S(k), \quad I(k) = A(k)S(k)$$

2. when $\mu = \longrightarrow \infty$

$$N(k) = S(k), \quad S_0(k) = 0., \quad I(k) = 0.$$

All intermediate cases, $\mu \in [0, \infty)$, corresponds to quasi-predictability (trade-off between additive noise attenuation and predictability of signal).

5.2.2 Space domain formulation

Soubaras also derived space domain expression for the projection filter (small bolds are vectors, capital bolds denote matrices),

$$\mathbf{s} = \mathbf{s_0} + \mathbf{n} \tag{5.36}$$

$$\mathbf{A s_0} = \mathbf{i}\,. \tag{5.37}$$

In this case the vectors denote the spatial series of observations $\mathbf{s}$ and the unknowns $\mathbf{s_0}$ and n. The operation As_0 also corresponds to convolution with the PEO. The cost function for the problem becomes:

$$J = (\mathbf{s} - \mathbf{s_0})^T(\mathbf{s} - \mathbf{s_0}) + \mu \mathbf{s_0}^T \mathbf{A}^T \mathbf{A} \mathbf{s_0} \tag{5.38}$$

taking derivatives yields to

$$\mathbf{s} = (\mathbf{I} + \mu \mathbf{A}^T \mathbf{A})\mathbf{s_0} \tag{5.39}$$

since $\mathbf{s} = \mathbf{s_0} + \mathbf{n}$

$$\mathbf{s} = (\mathbf{I} + \mu \mathbf{A}^T \mathbf{A})(\mathbf{s} + \mathbf{n}) \tag{5.40}$$

$$\mathbf{n} = \mu(\mathbf{I} + \mu \mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{A} \mathbf{s} \tag{5.41}$$

where after using the identity $\mathbf{A}^T(\mathbf{I} + \mu \mathbf{A}\mathbf{A}^T)^{-1} = (\mathbf{I} + \mu \mathbf{A}\mathbf{A}^T)^{-1}\mathbf{A}^T$,

$$\mathbf{n} = \mathbf{A}^T(\mu^{-1}\mathbf{I} + \mathbf{A}\mathbf{A}^T)^{-1}\mathbf{A}\mathbf{s}\,. \tag{5.42}$$

The projection filter deconvolves the PEO filter. To prove this assessment suppose that we know the filter $\mathbf{A}$. Unfortunately is not true that we know the $\mathbf{s_0}$, but suppose that we apply the filter to $\mathbf{s}$

$$\mathbf{A}\mathbf{s} = \mathbf{A}(\mathbf{s_0} + \mathbf{n}) = \mathbf{A}\mathbf{s_0} + \mathbf{A}\mathbf{n}\,. \tag{5.43}$$

It is clear from the last expression that the filter output has two components, the innovation, $\mathbf{A}\mathbf{s_0}$, which is zero if the noiseless signal is perfectible predictable plus the filtered noise sequence $\mathbf{A}\mathbf{n}$. If one looks carefully at equation (5.42) it is clear that the projection filter does the deconvolution of the PEO signature from the noise sequence,

$$\mathbf{n} = \underbrace{\mathbf{A}^T(\mu^{-1}\mathbf{I} + \mathbf{A}\mathbf{A}^T)^{-1}}_{2}\underbrace{(\mathbf{A}\mathbf{s})}_{1} \tag{5.44}$$

The operation 1 does the filtering of the signal and the noise, meanwhile the operation 2 removes the colour that the filter has introduced into the noise (deconvolution of the PEO).

The term denoted 2 in expression (5.44) is also the pseudo-inverse of the PEO filter $\mathbf{A}$, which I will define $\mathbf{A}^+$.

5.2.3 Wrong formulation of the problem

Suppose, again, that we know the PEO filter. The PEO when applied to the noise data does the following

$$\mathbf{As} = \mathbf{A}(\mathbf{s_0} + \mathbf{n}) = \underbrace{\mathbf{As_0}}_{1} + \underbrace{\mathbf{An}}_{2}. \tag{5.45}$$

It is evident that we want the term denote 1 to be small (predictability) and the term denote 2 to be small too!! (noise attenuation). We can try to conciliate both wishes by defining the following objective function:

$$J = \mathbf{s_0}\mathbf{A}^T\mathbf{As} + \mu(\mathbf{s} - \mathbf{s_0})^T\mathbf{A}^T\mathbf{A}(\mathbf{s} - \mathbf{s_0}) \tag{5.46}$$

taking derivatives with respect to $\mathbf{s_0}$ yields

$$\mathbf{s_0} = \frac{\mu}{\mathbf{1} + \mu}\mathbf{s} \tag{5.47}$$

Ooohps, it is clear that there is something wrong with equation (5.47). This is a consequence of written an incorrect objective function. In fact, what we want to minimize is the term denote 1 in equation (5.45) plus the actual noise term $\mathbf{n} = \mathbf{s} - \mathbf{s_0}$ and not the the filtered noise $\mathbf{An}$. So it is clear that the proper objective function of the problem is given by equation (14) and not by equation (5.46).

Soubaras's projection filter is an attempt to estimate from a given PEO an optimum noise sequence. Or in other words, the projection filter is the solution of a inverse problem where an unfiltered noise sequence is the unknown.

5.3 ARMA formulation of Projection filters

It is important to note that undelying signal model in $f - X$ is an ARMA model. So far, we have done prediction and noise removal using two concepts: 1- Conventing the ARMA problem into an AR problem and performing conventional linear prediction (Canales' method) and 2- Invoke the

concept of quasi-predictability and solve a problem where both PEF and the additive noise are unknowns (Soubaras' mnethod)
One of the problems of AR filtering (Canales' method) is that the noise enters into the probblem as a innovation rather than as additive noise. Soubaras volved this problem by introducing the theory of projection filters. In this section of the course we will show that projection filters can be computed by solving the original ARMA problem without introducing the concept of quas-predictability.
Let's go back to the orginal system of equation for our ARMA problem (equation (5.7)):

$$\begin{aligned} \textstyle\sum_{k=0}^{p} g_k Y_{n-k} &= \textstyle\sum_{k=0}^{p} g_k W_{n-k} \\ &= e_n \,. \end{aligned} \tag{5.48}$$

we have mentioned that the latter is an ARMA(p,p) process in which the AR and MA components are identical. The signal e_n in equation designates the non-white innovation sequence $\sum_{k=0}^{p} g_k W_{n-k}$.
The problem now can be summarized as follows, given the ARMA representation of the noise signal:

1. How do we estimate the prediction error filter g_k?

2. How do we use g_k to estimate the additive noise sequence W_k.?

This two fundamental points are discussed in the following pages.

5.3.1 Estimation of the ARMA prediction error filter

Equation (5.7) can be written in matrix form as follows:

$$\begin{aligned} \mathbf{Yg} &= \mathbf{Wg} \\ &= \mathbf{e}\,, \end{aligned} \tag{5.49}$$

where $\mathbf{Y}$ is the convolution matrix of the signal with entries given by the noisy sequence Y_k properly shifted and padded with zeros in order to represent discrete convolution. For a signal of length $N = 4$ samples and a prediction error filter of length 3 ($p = 2$) the convolution matrix takes the following form:

$$\mathbf{Y} = \begin{pmatrix} Y_0 & 0 & 0 \\ Y_1 & Y_0 & 0 \\ Y_2 & Y_1 & Y_0 \\ Y_3 & Y_2 & Y_1 \\ 0 & Y_3 & Y_2 \\ 0 & 0 & Y_3 \end{pmatrix} . \tag{5.50}$$

It is important to stress that the transient-free convolution matrix proposed by Ulrych and Clayton (1976) can also be adopted (Harris and White, 1997; Wang, 1999). The matrix $\mathbf{W}$ in equation (5.49) is the convolution matrix of the noise with entries given by the unknown noise sequence W_k.

If we assume that the noise is spatially uncorrelated and stationary, the prediction error filter $\mathbf{g}$ can be estimated by transforming equation (5.49) into an eigenvalue problem. For this purpose we first multiply both sides of equation (5.49) by $\mathbf{Y}^H$ (the Hermitian transpose of the matrix $\mathbf{Y}$) and after considering that noise and signal are spatially uncorrelated we arrive to the following expression

$$\mathbf{R}_Y \mathbf{g} = P_W \mathbf{g} \, . \tag{5.51}$$

The matrix $\mathbf{R}_Y = \mathbf{Y}^H\mathbf{Y}$ is the Toeplitz correlation matrix of the noisy data. The scalar P_W represents the noise power.

The matrix $\mathbf{R}_Y$ is positive definite with $p+1$ eigenvalues and eigenvectors (Marple, 1987). Equation (5.51) admits $p+1$ solutions. Each solution corresponds to an eigenvector and its associated eigenvalue. Let us assume the that i-th eigenvector is the solution of our problem, $\mathbf{g} = \mathbf{v}_i$. In this case the power of the colored noise sequence $\mathbf{e}$ can be written as

$$\begin{aligned} \mathbf{e}^H\mathbf{e} &= \mathbf{v}_i^H \mathbf{R}_Y \mathbf{v}_i \\ &= P_W \mathbf{v}_i^H \mathbf{v}_i \\ &= \lambda_i \, , \end{aligned} \tag{5.52}$$

it is clear that since the eigenvectors are normalized $\lambda_i = P_W$. In other words the eigenvector corresponding to the minimum eigenvalue minimizes the power of the additive white noise. If the eigenvalues are sorted in descendent order our final estimator of the prediction error filter is given by $\mathbf{g} = \mathbf{v}_{p+1}$ and the minimum eigenvalue λ_{p+1} gives an estimate of the noise power. The other eigenvectors provide a solution that corresponds to local minima of the quadratic form $\mathbf{e}^H\mathbf{e}$.

Equation (5.5.1) is the basis of the Pisarenko Harmonic Decomposition (PHD) (Pisarenko, 1973). In the PHD the frequencies or wavenumbers are determined by factoring the z-transform of the prediction error filter. The roots of the prediction error filter are guaranteed to lie on the unit circle (Marple, 1987).

5.3.2 Noise estimation

After estimating $\mathbf{g}$ the remaining problem is to estimate the noise sequence $\hat{W}_k$ which will be used to estimate the "clean" signal $\hat{S}_k = Y_k - \hat{W}_k$. The noise is estimated by deconvolving the prediction error filter from the non-white innovation in equation (5.49). In order to facilitate the algebra we rewrite equation (5.49) by commuting the sequences involved in the convolution

$$\mathbf{G}\mathbf{y} = \mathbf{G}\mathbf{w}\,. \tag{5.53}$$

The matrix $\mathbf{G}$ is the convolution matrix of the prediction error filter, $\mathbf{y}$ and $\mathbf{w}$ are vectors containing the observations and the white noise sequence, respectively. We must stress that after computing the prediction error filter, the right hand side term in equation (5.53) is known. This equation admits a trivial solution of the form $\hat{\mathbf{w}} = \mathbf{y}$. This solution implies that the signal $\mathbf{s} = 0$. This shortcoming is avoided by solving a constrained minimization problem. We minimize the following quadratic form

$$J = [\mathbf{G}(\mathbf{y} - \mathbf{w})]^H [\mathbf{G}(\mathbf{y} - \mathbf{w})] \tag{5.54}$$

subject to

$$\mathbf{w}^H\mathbf{w} = P_w\,. \tag{5.55}$$

The constrained minimization problem is solved by introducing a Lagrange multiplier to combine the last two equations into a new objective function

$$J' = [\mathbf{G}(\mathbf{y} - \mathbf{w})]^H [\mathbf{G}(\mathbf{y} - \mathbf{w})] + \mu(\mathbf{w}^H\mathbf{w} - P_W)\,. \tag{5.56}$$

The objective function J' is minimized by the following estimator of the noise sequence

$$\hat{\mathbf{w}} = (\mathbf{G}^H\mathbf{G} + \mu\mathbf{I})^{-1}\mathbf{G}^H\mathbf{G}\mathbf{y}\,. \tag{5.57}$$

The "clean signal", $\hat{\mathbf{s}} = \mathbf{y} - \hat{\mathbf{w}}$, can be estimated as follows:

$$\hat{\mathbf{s}} = [\mathbf{I} - (\mathbf{G}^H\mathbf{G} + \mu\mathbf{I})^{-1}\mathbf{G}^H\mathbf{G}]\mathbf{y}\,. \tag{5.58}$$

At this stage some comments are in order. First, we note that when $\mu = 0$, $\hat{\mathbf{w}} = \mathbf{y}$. In other words, we have annihilated the signal ($\hat{\mathbf{s}} = 0$). If μ is too large, $\hat{\mathbf{w}} = \mathbf{0}$. In this case there is no noise attenuation ($\hat{\mathbf{s}} = \mathbf{y}$). This is also the case of perfect predictability in the absence of noise where equation (5.53) becomes $\mathbf{G}\mathbf{s} = \mathbf{0}$.

In general, a line search procedure could be used to determine the value of μ that yields a noise sequence with a mean square error that agrees with the estimated noise power obtained after solving equation (5.51). In practical applications the parameter μ is determined by assessing the quality of the final results in the same way as it is done in seismic deconvolution.

5.3.3 ARMA and Projection Filters

Our algorithm is composed of two stages: 1) determination of the prediction error filter or eigenfilter and 2) noise estimation. Stage 2 is equivalent to the projection filtering technique proposed by Soubaras (1994, 1995). In Soubaras' technique the noise is estimated by minimizing an objective function that establishes a tradeoff between predictability (or pseudo-predictability) and random noise attenuation. Soubaras first assumes that the prediction error filter is known and then uses the objective function J' (equation (5.56)) to determine the noise sequence. The first term of J' is associated to signal predictability, the second term is associated to random noise attenuation. In the projection filter approach the prediction error filter is bootstrapped from the data in an iterative manner.

It is clear that the ARMA representation of the seismic signal in the $f-x$ domain leads not only to a technique to estimate the prediction error filter but also to the projection filter estimator of the noise sequence.

In Figure (5.9) we present a 1D synthetic example. The signal is composed of one real sinusoid immersed in white noise with standard error $\sigma = 0.5$. The number of signals is $p = 2$ (1 real sinusoid is represented by 2 complex harmonics). We have estimated the noise sequence using different tradeoff parameters (Figure (5.10)). In Figure (5.11) we portrayed the estimated signal versus the tradeoff μ. The optimum tradeoff parameter is $\mu = 0.01$. This value yields a noise sequence with variance $\hat{\sigma} = 0.53$. In Figure (5.12) we portray the transfer function of the projection filter utilized in the example. The wavenumber response of the projection filter is given by

$$A(k) = \frac{|g(k)|^2}{\mu + |g(k)|^2} , \qquad (5.59)$$

where $g(k)$ is the Fourier transform of the eigenfilter. In this example the eigenfilter has a root on the unit circle at $k_0 = 0.05$, therefore equation (5.59) represents a notch filter that eliminates that particular spectral component. When the proper value of μ is chosen, the transfer function $A(k)$ selectively attenuates the signal letting only the noise pass. It is clear that when μ is increased the transfer function will attenuate both signal and noise.

In Figures (5.13) and (5.14) we compare the performance of Canales' $f - x$ AR filtering using a 15 points prediction error filter and the $f - x$ ARMA approach using a 4 points eigenfilter ($p = 3$). These results look quite similar. However, one can observe that in the AR formulation a portion of the signal leaks into the noise space.

In Figure (5.15) we provide a field data example. We have used the ARMA approach and the AR $f - x$ filtering approach to attenuate random noise in a common offset gather section. The two methods produce similar results. In the ARMA method we have adopted a filter of length 3 ($p = 2$) and a tradeoff parameter $\mu = 0.1$. The AR filtered panel was obtained using a forward/backward prediction error filter of length $p = 15$. These results look quite similar, it is clear that in real data applications both models (AR and ARMA) do not properly represent the true underlying signal and this is the major source of errors in both techniques.

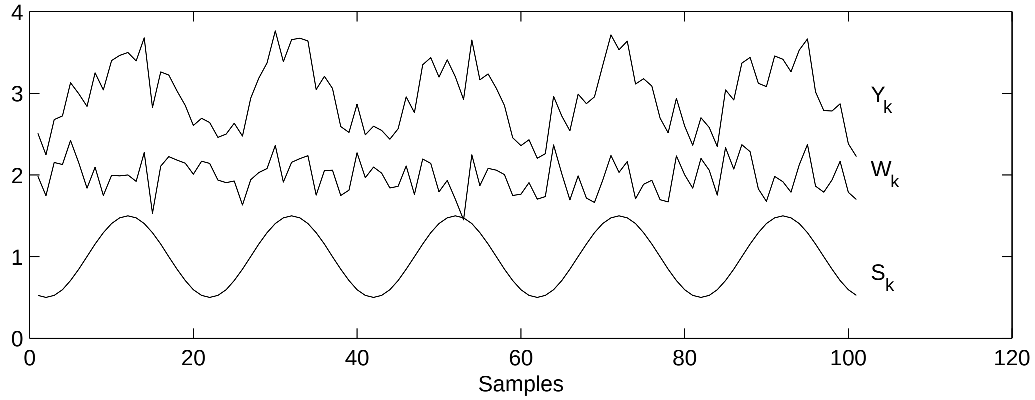


Figure 5.9: D synthetic example. A sinusoid of normalized wavenumber $k_0 = 0.05$ is used to test the ARMA filtering method described in the paper. The data are contaminated with white noise with standard error $\sigma = 0.5$.

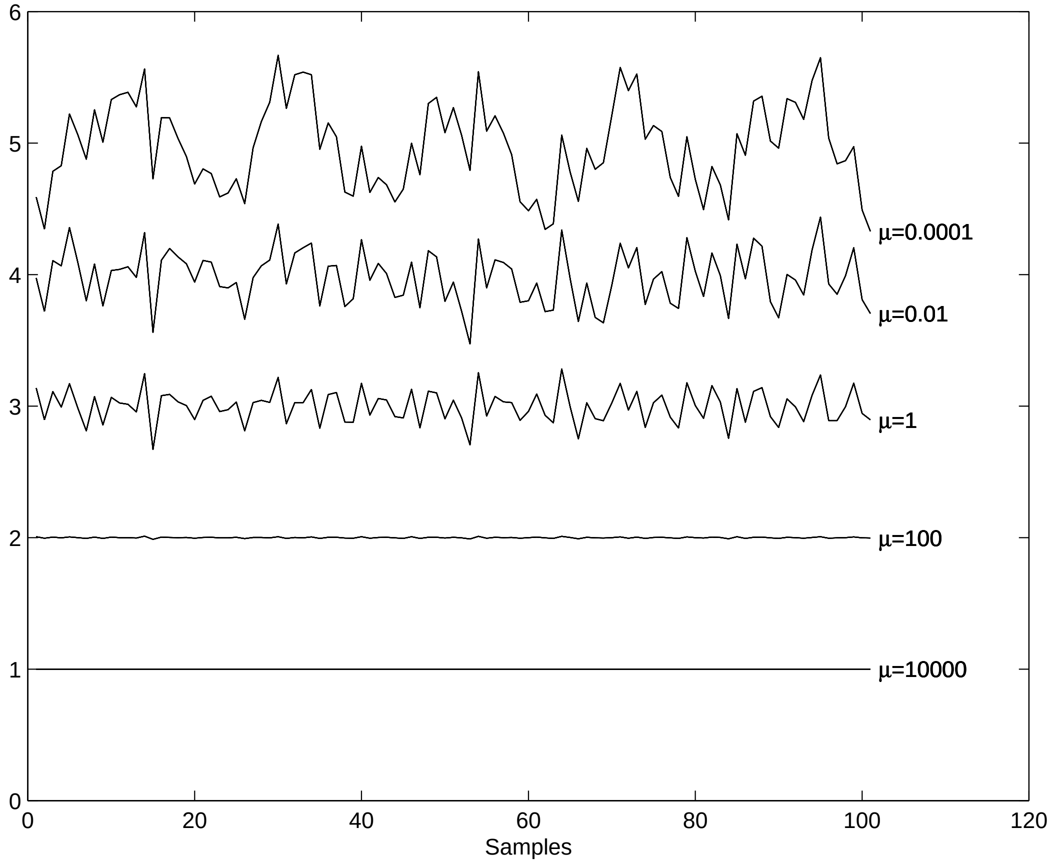


Figure 5.10: The eigenfilter estimated from the noisy data is used to estimate the noise. In this figure we portray the estimator of the noise sequence versus the tradeoff parameter μ. Large values of μ will completely annihilate the noise. Small values of μ will overestimate the noise (the signal leaks into the noise sequence.)

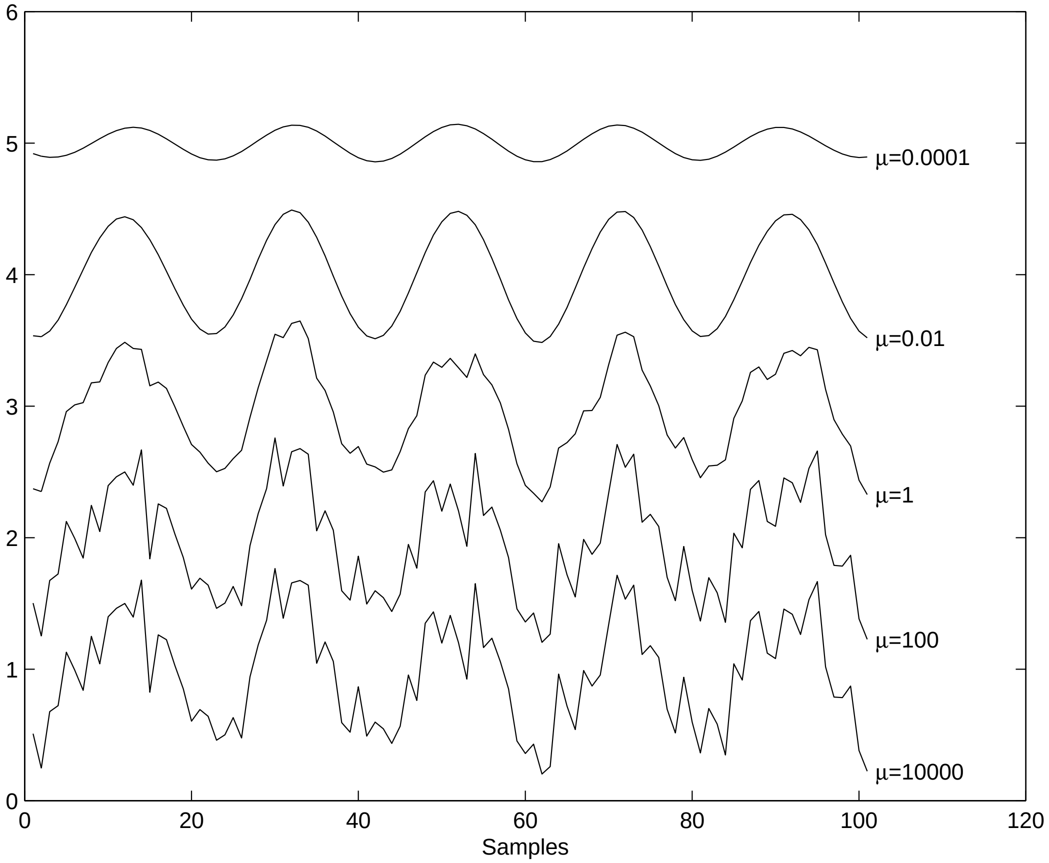


Figure 5.11: Estimator of the signal versus the tradeoff parameter μ. Note that for $\mu = 0.01$ the signal is recovered.

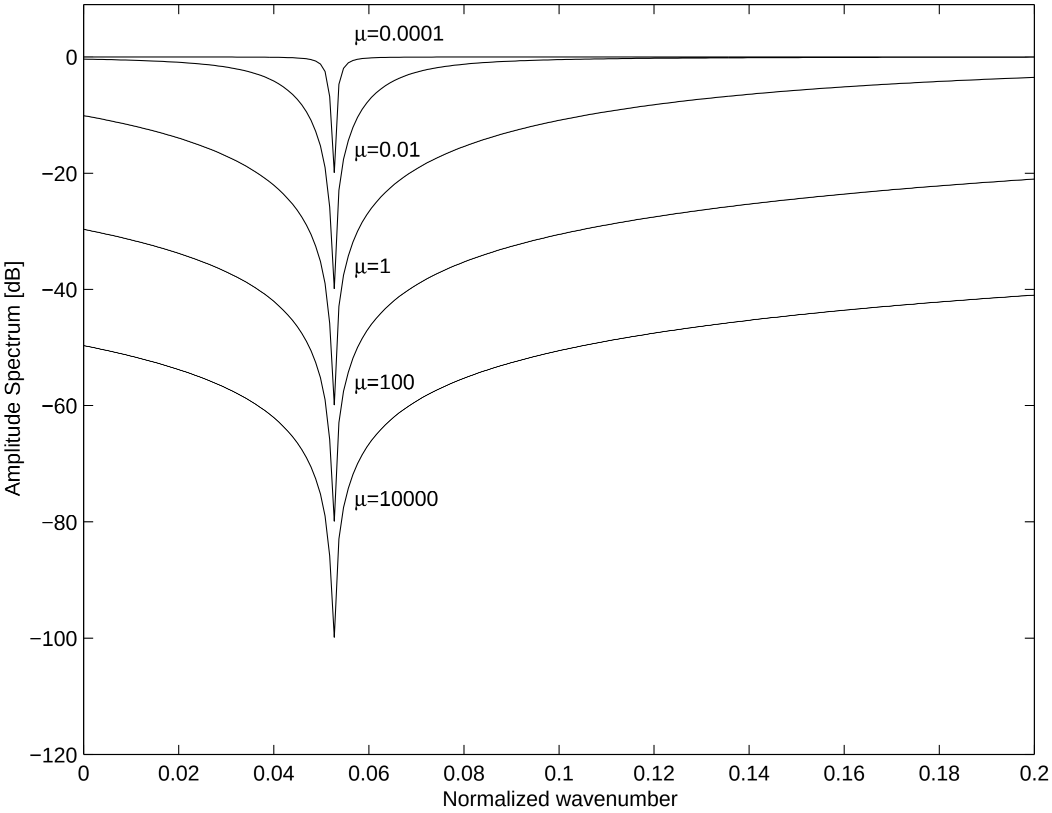


Figure 5.12: Amplitude response of the projection filter utilized to estimat e the noise in Figure 1b. Note that large values of μ attenuates both signal and noise. On the other h and, small value of μ does not properly attenuate the signal.

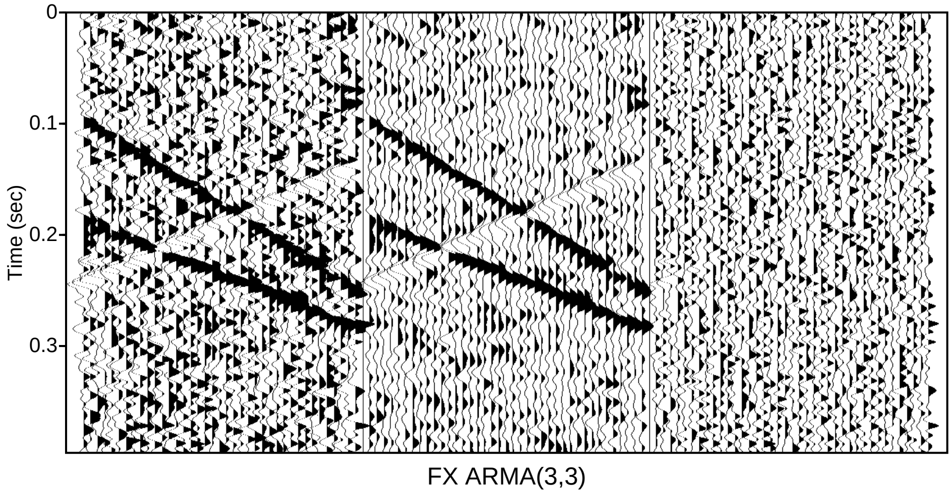


Figure 5.13: Left: The original data (3 waveforms immersed in spatially uncorrelated noise). Center: Filtered data using the ARMA representation. Right: Estimate of the noise.

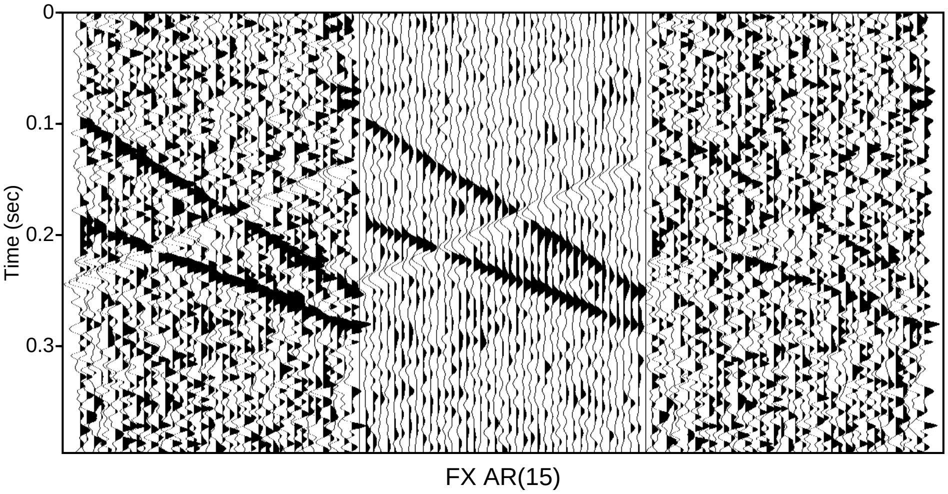


Figure 5.14: Left: The original data (3 waveforms immerse in spatially unco rrelated noise). Center: Filtered data using the AR representation (Conventional $f - x$ r andom noise attenuation). Right: Estimate of the noise.

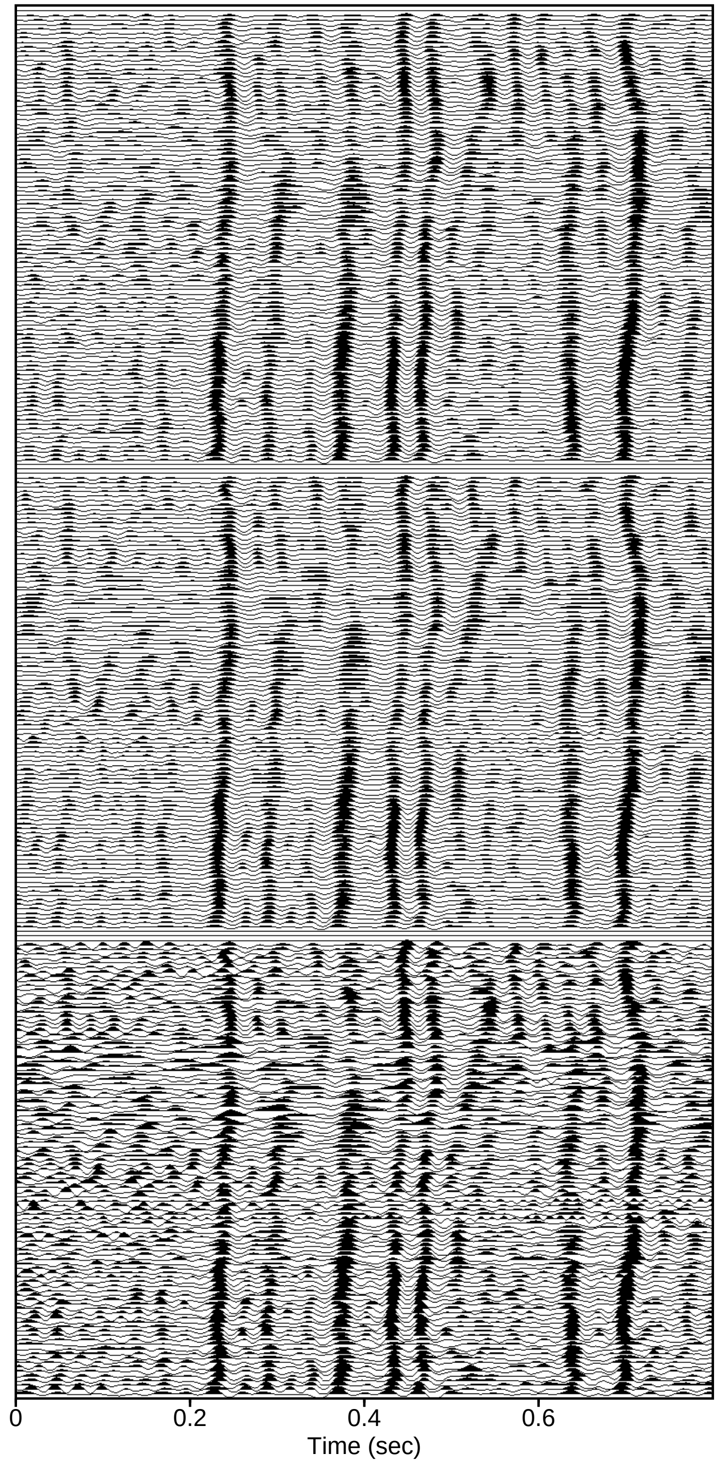


Figure 5.15: Left: A window of a common offset gather. Center: The filtered section using the ARMA representation with a 3 points eigenfilter ($p = 2$). Right: Conventional $f - x$ random noise attenuation ($p = 15$).

5.4 FX Processing Codes

5.4.1 Prediction of harmonic models using AR filters

Program used to obtain Figures (5.1) and (5.2)

```
% A program for linear prediction of
% harmonic models, 1D example.

% First we prepara a synthetic

dt = 4./1000.; nt =290;
t = 0:dt:(nt-1)*dt;
f = 10;
lf =20;

% Prepare a noise signal
D_in = cos(2*pi*f*t)' + 0.2*randn(nt,1);
D_pred = zeros(nt,1);
C = convmtx(D_in,lf);
d = zeros(nt+lf-1,1); d(1:nt-1) =D_in(2:nt);

% Compute the pef
mu = .001;
f = inv(C'*C+mu*eye(lf))* C'*d;
aux = conv(f,D_in);

% Forward prediction
D_pred(2:nt,1) = aux(1:nt-1);

% Prediction error
E = D_in  - D_pred;
```

5.4.2 FX algorithm, Canales (1984)

Program used to prepare Figures (5.3) and (5.4)

```
% FX.m
% Canales method for SNR enhancement.
% needs function ar.m to compute the predicted
% at a given frequency

clear; clf

dt = 4./1000;                         % Make a synthetic t-x model
w = ricker(20.,dt);
nw=max(size(w));
nx = 32; nt = 128;
DATA = zeros(nx,nt);
for i=1:nx
 for j=1:nw
  DATA(i,20+j+i) = w(j);
 end
end

NOISE = 0.2 * randn(nx,nt);           % Add noise to data
DATA = DATA + NOISE;


p = 20                                % Length of the pef
DATA_FX = fft(DATA,[],2);             % Go to f-x

 for i=1:nt;                          % Do prediction at each freq
  aux_in  = DATA_FX(:,i);
   aux_out = ar(aux_in,p);
  DATA_FX(:,i) = aux_out;
 end
DATA = real(ifft(DATA_FX,[],2));      % Back to t-x
```

```
subplot(211);wigb(DATA');
title('Input, \sigma_n=0.2')

subplot(212);wigb(DATA');
title('Output after FX filtering, p=20')
```

5.4.3 Linear prediction using AR filters

Function to perform 1D prediction.

```
function [D_pred] = ar(D_in,lf);
% 1D prediction function used by Canales' method.
%
% D_in: Data in f-x (one column at freq. f)
% lf: length of the pef (AR)
% D_pred: predicted data (clean data)
%

n = max(size(D_in));
D_pred = zeros(n,1);
C = convmtx(D_in,lf);             % convolution matrix
d = zeros(n+lf-1,1);              % RHS vector (desired output)
d(1:n-1) =D_in(2:n);

mu = .001;
f = inv(C'*C+mu*eye(lf))* C'*d;    % Filter (AR)
aux = conv(f,D_in);               % apply filter to data

D_pred(2:n,1) = aux(1:n-1);       % Prediction
return
```

5.4.4 ARMA filtering

Program used to generate Figures (5.10), (5.11), and (5.12).

```
 function [s,w,g] = eigen_filtering(y,p,mu);
% Given a 1D noisy sequence y, the order p of
% the ARMA(p,p) model and the regularization parameter mu
% this function computes the clean signal s, an estimate of
% the noise sequence w, and the prediction error filter g.

 N = length(y);
 Y = convmtx(y,p+1);                 % data convolution matrix
 R = Y'*Y/N;                         % data correlation matrix
 [g,Pw] = eigs(R,1,'SM');            % compute 1 eigenvalue (the SMallest)
                                     % and the associated eigenvector
 g0 = g(1);
 g  = g/g0;
 e = Y*g;                            % non-white sequence e
 G = convmtx(g,N);                   % convolution  matrix of the filter
 D = eye(N)*mu;                      % regularization term
 w = inv(G'*G+D)*G'*e;               % estimate of the noise
 s = y-w;                            % estimate of the clean signal
 return
```

5.4.5 References

Akaike, H., 1974, A new look at statistical model information, IEEE Trans. Auto. Control, **AC-19**, 716-723.

Canales, L. , 1984, Randon noise reduction: 54th Annual SEG Meeting, p.525-527.

Fahlman, G. G., and Ulrych, T. J., 1982, A new method of estimating the power spectrum of gapped data: Mon. Not. Roy. Astr. Soc., v.199, p.53-65.

Harris, P. E., and White, R. E., 1997. Improving performance of $f - x$ prediction filtering at low signal-to-noise ratios: Geophysical Prospecting, **45**, 269-302.

Marple, S. L., 1987. Digital Spectral Analysis. Prentice-Hall, Inc.

Pisarenko, V.P, 1973. The retrieval of harmonics from a covariance function. Geophys. J. R. astr. Soc., **33**: 347-366.

Sacchi M.D., and Kuehl, H., 2001, ARMA formulation of fx Prediction error filters and projection filters: Journal of Seismic Exploration, **9**, 185-197.

Soubaras R., 1994, Signal preserving random noise attenuation by the f-x projection: 64 Annual SEG Mtg. 1576-1579

Soubaras R., 1995, Prestack random and impulsive noise attenuation bt f-x projection filtering: 65 Annual SEG Mtg. 711-714

Soubaras R., 1995, Deterministical and statistical noise attenuation: 57th EAEG Mtg.

Spitz, S., 1991, Seismic trace interpolation in the F-X domain: Geophysics, **56**, 785-794.

Ulrych, T. J., and Clayton, R. W., 1976, Time series modeling and maximum entropy: Phys. of the Earth and Plan. Int., **12**, 188-200.

Walker, C., and Ulrych, T. J., 1983, Autoregressive recovery of the acoustic impedance: Geophysics, **48**, 1338-1350.

Wang, Xi-Shuo, 1995, Random noise attenuation of pre-stack data by surface consistent prediction in the frequency domain: CSEG National Convention, Calgary, Expanded Abstracts, 133-134.

Wang, Y., 1999. Random noise attenuation using forward-backward linear prediction. Journal of Seismic Exploration, **8**: 133-142.

Wiggins, R. A, and Miller, S. P., 1972, New noise-reduction technique applied to long-period oscillations from the Alaskan earthquake: SSA Bull., **62**, 417-479.

Chapter 6

The KL transform and eigenimages

In this chapter we will discuss another technique to improve the information content of seismic data. The application of eigenimage analysis in seismology was proposed by Hemon and Mace (1978). In their approach they use a particular linear transformation called the Karhunen-Loeè (KL) transformation. The KL trnsormation is also knows as the principal component transformation, the eigenvector transfomation or the Hotelling transofmation. Of particular relevance to the ensuing discussion is the excellent paper by Ready and Vintz (1973) which deals with information extraction and SNR improvement in multispectral imagery.

In 1983, the work of Hemon and Mace was extended by a group of researchers at the University of British Columbia in Canada which culminated in the work of Jones and Levy (1987).

In 1988 Freire and Ulrych applied the KL transformation in a somewhat different manner to the processing of vertical seismic profiling data. The actual approach which was adopted in this work was by means of singular value decomposition (SVD), which is another way of viewing the KL transformation (the relationship between the KL and SVD transformations is discussed in this chapter).

A seismic section which consists of M traces with N points per trace may be viewed as a data matrix $\mathbf{X}$ where each element x_{ij} represents the i^{th} point of the j^{th} trace. A singular value decomposition (Lanczos, 1961), decomposes $\mathbf{X}$ into a weighted sum of orthogonal rank one matrices which have been designated by Andrews and Hunt (1977) as eigenimages of $\mathbf{X}$. A particularly useful aspect of the eigenimage decomposition is its application in the complex form. In this instance, if each trace is transformed into the analytic form, then the eigenimage processing of the complex data matrix allows both time and phase shifts to be considered which is of particular importance in the case of the correction of residual statics.

6.1 Mathematical framework

We consider the data matrix $\mathbf{X}$ to be composed of M traces with N data points per trace, the M traces forming the rows of $\mathbf{X}$. The SVD of $\mathbf{X}$ is given by, (Lanczos (1961)),

$$\mathbf{X} = \sum_{\mathbf{i=1}}^{\mathbf{r}} \sigma_\mathbf{i} \mathbf{u_i} \mathbf{v_i^T}. \tag{6.1}$$

where ^{T} indicates transpose, r is the rank of $\mathbf{X}$, $\mathbf{u}_i$ is the ith eigenvector of $\mathbf{XX^T}$, $\mathbf{v}_i$ is the ith eigenvector of $\mathbf{X^TX}$ and σ_i is the ith singular value of $\mathbf{X}$. The singular values σ_i can be shown to be the positive square roots of the eigenvalues of the matrices $\mathbf{XX^T}$ and $\mathbf{X^TX}$. These eigenvalues are always positive owing to the positive definite nature of the matrices $\mathbf{XX^T}$ and $\mathbf{X^TX}$. In matrix form equation (6.1) is written as

$$\mathbf{X} = \mathbf{U\Sigma V^T} \tag{6.2}$$

Andrews and Hunt (1977) designate the outer dot product $\mathbf{u}_i\mathbf{v}_i^T$ as the ith eigenimage of the matrix $\mathbf{X}$. Owing to the orthonormality of the eigenvectors, the eigenimages form an orthonormal basis which may be used to reconstruct $\mathbf{X}$ according to equation (6.1).

Suppose, for example, that $\mathbf{X}$ represents a seismic section and that all M traces are linearly independent. In this case $\mathbf{X}$ is of full rank M, all the σ_i are different from zero and a perfect reconstruction of $\mathbf{X}$ requires all eigenimages. On the other hand, in the case where all M traces are equal to within a scale factor, all traces are linearly dependent, $\mathbf{X}$ is of rank one and may be perfectly reconstructed by the first eigenimage $\sigma_1\mathbf{u_1}\mathbf{v_1^T}$. In the general case, depending on the linear dependence which exists among the traces, $\mathbf{X}$ may be reconstructed from only the first few eigenimages. In this case, the data may be considered to be composed of traces which show a high degree of trace-to-trace correlation. Indeed, $\mathbf{XX^T}$ is, of course, a weighted estimate of the zero lag covariance matrix of the data $\mathbf{X}$ and the structure of this covariance matrix, particularly the distribution of the magnitudes of the corresponding eigenvalues, indicates the parsimony or otherwise of the eigenimage decomposition. If only $p, p < r$, eigenimages are used to approximate $\mathbf{X}$, a reconstruction error ϵ is given by

$$\epsilon = \sum_{k=p+1}^{r} \sigma_k^2 \,. \tag{6.3}$$

Freire and Ulrych (1988) defined band-pass $\mathbf{X_{BP}}$, low-pass $\mathbf{X_{LP}}$ and high-pass $\mathbf{X_{HP}}$ eigenimages in terms of the ranges of singular values used. The band-pass image is reconstructed by rejecting highly correlated as well as highly uncorrelated traces and is given by

$$\mathbf{X_{BP}} = \sum_{\mathbf{i=p}}^{\mathbf{q}} \sigma_\mathbf{i}\mathbf{u_i}\mathbf{v_i^T}\,, \qquad \mathbf{1 < p \leq q < r}\,. \tag{6.4}$$

The summation for X_{LP} is from $i = 1$ to $p - 1$ and for $\mathbf{X_{HP}}$ from $i = q + 1$ to r. It may be simply shown that the percentage of the energy which is contained in a reconstructed image X_{BP} is given by E, where

$$E = \frac{\sum_{i=p}^{q} \sigma_i^2}{\sum_{i=1}^{r} \sigma_i^2}\,. \tag{6.5}$$

The choice of p and q depends on the relative magnitudes of the singular values, which are a function of the input data. These parameters may, in general, be estimated from a plot of the eigenavalues $\lambda_i = \sigma_i^2$ as a function of the index i. In certain cases, an abrupt change in the eigenvalues is easily recognised. In other cases, the change in eigenvalue magnitude is more

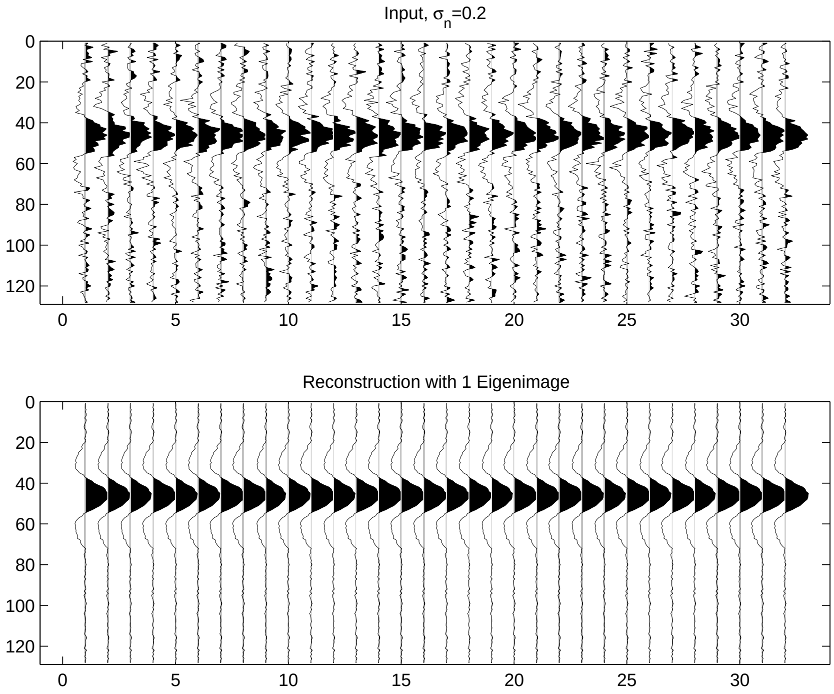


Figure 6.1: A flat event immersed in nose and the recostruction by means of the first eigenimage

gradual and care must be exercised in the choice of the appropriate index values.

In Figures 6.1 and 6.2 we illustrate the reconstruction fo a flat event immersed in noise using the first eigenimage of the data. In this example only the most energetic singular value was retained. When the data exhibit some type of moveout, one eigenimage is not sufficient to properly reconstruct the data. This can be observed in Figures 6.3 and 6.4.

As we have seen, decomposition of an image **X** into eigenimages is performed by means of the SVD of **X**. Many authors also refer to this decomposition as the Karhunen-Loève or KL transformation. We believe however, that the SVD and KL approaches are not equivalent theoretically for image processing and, in order to avoid confusion, we suggest the adoption of the term eigenimage processing. Some clarification is in order.

A wide sense stationary process $\xi(t)$ allows the expansion

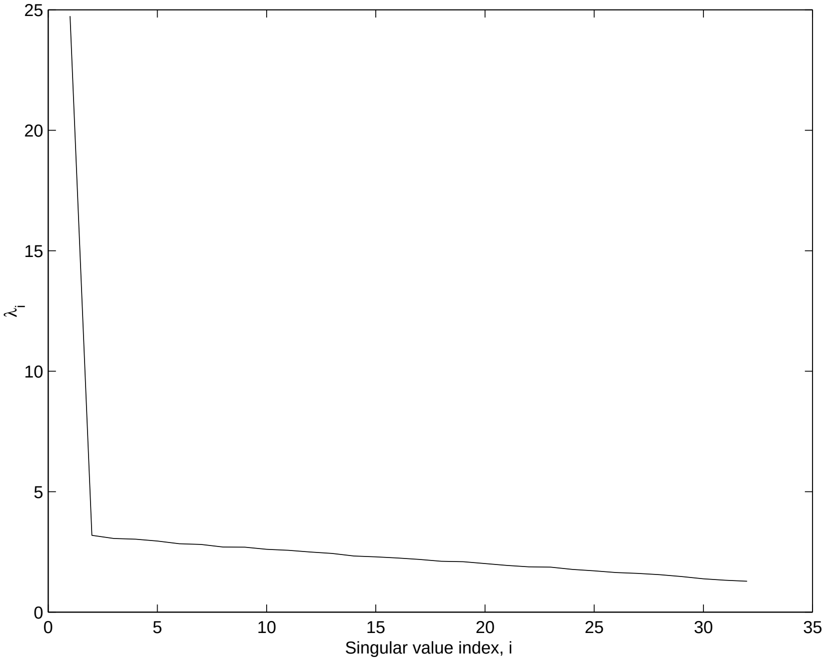


Figure 6.2: Spectrum of singular values for the data in Figure 6.1.

$$\widehat{\xi}(t) = \sum_{n=1}^{\infty} c_n \psi_n(t) \qquad 0 < t < T \tag{6.6}$$

where $\psi_n(t)$ is a set of orthonormal functions in the interval $(0, T)$ and the coefficients c_n are random variables. The Fourier series is a special case of the expansion given by equation (6.6) and it can be shown that, in this case, $\xi(t) = \widehat{\xi}(t)$ for every t and the coefficients c_n are uncorrelated only when $\xi(t)$ is mean squared periodic. Otherwise, $\xi(t) = \widehat{\xi}(t)$ only for $|t| < T/2$ and the coefficients c_n are no longer uncorrelated. In order to guarantee that the c_n are uncorrelated and that $\xi(t) = \widehat{\xi}(t)$ for every t without the requirement of mean squared periodicity, it turns out that the $\psi_n(t)$ must be determined from the solution of the integral equation

$$\int_0^T R(t_1, t_2)\psi(t_2)dt_2 = \lambda\psi(t_1) \qquad 0 < t_1 < T \tag{6.7}$$

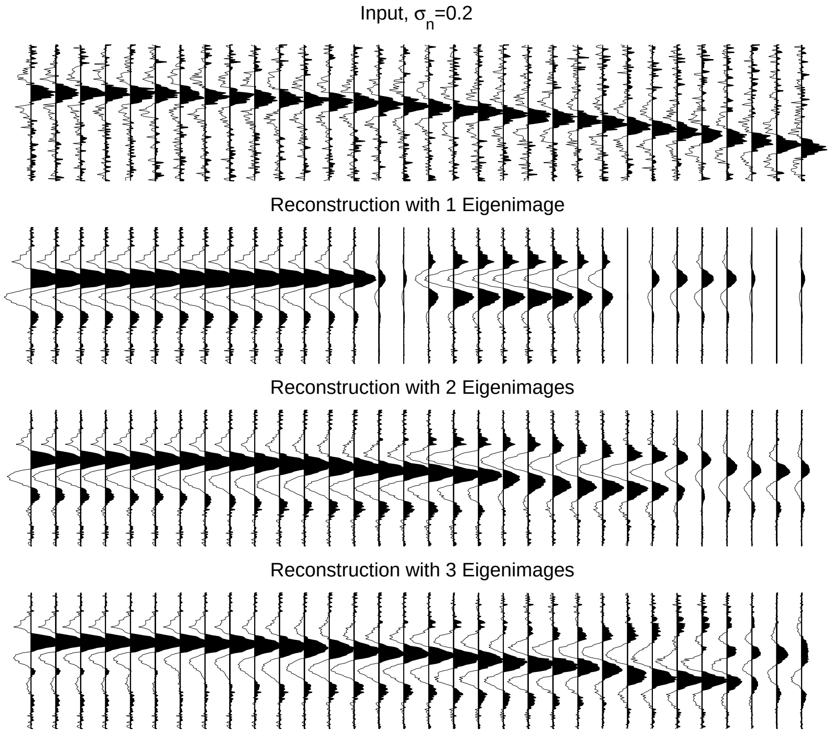


Figure 6.3: A Parabolic event immersed in nose and the reconstruction by means of the 1,2 and 3 eigenimages

where $R(t_1, t_2)$ is the autocovariance of the process $\xi(t)$.
Substituting the eigenvectors which are the solutions of equation (6.7) into equation (6.6) gives the KL expansion of $\xi(t)$. An infinite number of basis functions is required to form a complete set. For a $N \times 1$ random vector $\mathbf{x}$ we may write equation (6.6) in terms of a linear combination of orthonormal basis vectors $\mathbf{w_i} = (w_{i1}, wi2, \ldots, wiN)^T$ as

$$x_k = \sum_{i=1}^{N} y_i w_{ik} \qquad k = 1, 2, \ldots, N \tag{6.8}$$

which is equivalent to

$$\mathbf{x} = \mathbf{W}\mathbf{y} \tag{6.9}$$

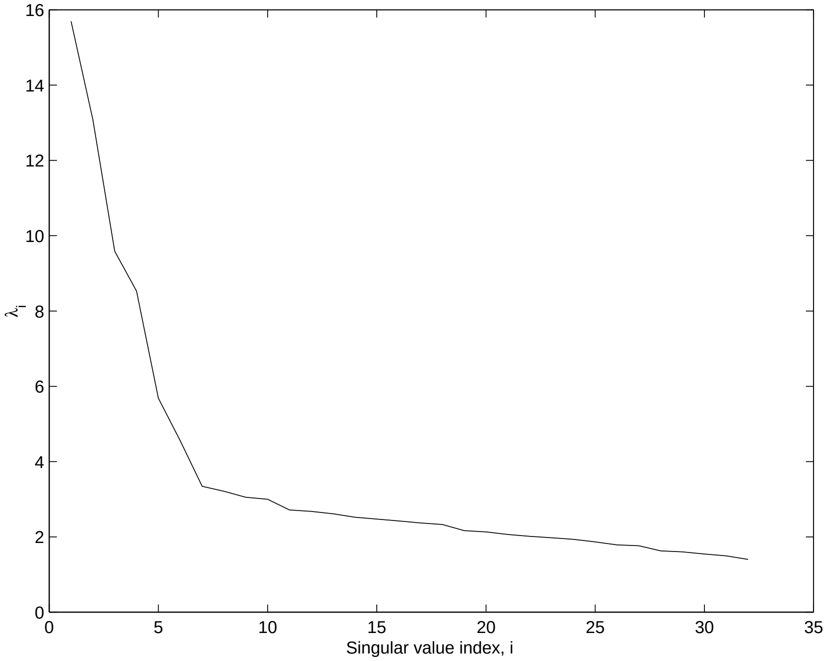


Figure 6.4: Spectrum of singular values for the data in Figure 6.3.

where $\mathbf{W} = (\mathbf{w_1}, \mathbf{w_2}, \ldots, \mathbf{w_N})$. Now only N basis vectors are required for completeness. The KL transformation or, as it is also often called, the KL transformation to principal components, is obtained as

$$\mathbf{y} = \mathbf{W^T x} \tag{6.10}$$

where $\mathbf{W}$ is determined from the covariance matrix $\mathbf{C_x}$ of the process

$$\mathbf{C_x} = \mathbf{W \Lambda W^T} \tag{6.11}$$

Let us now turn our attention to the problem of the KL transformation for multivariate statistical analysis. In this case we consider M vectors $\mathbf{x_i}, \mathbf{i} = \mathbf{1}, \mathbf{M}$ arranged in a $M \times N$ data matrix $\mathbf{X}$. The M rows of the data matrix are viewed as M realizations of the stochastic process $\mathbf{x}$ and consequently the assumption is that all rows have the same row covariance matrix $\mathbf{C_r}$. The KL transform now becomes

$$\mathbf{Y} = \mathbf{W^T X} \tag{6.12}$$

An unbiased estimate of the row covariance matrix is given by

$$\hat{\mathbf{C_r}} = \frac{1}{M-1}\sum_{i=1}^{M} \mathbf{x_i x_i^T} \tag{6.13}$$

assuming a zero mean process for convenience. Since the factor $M - 1$ does not influence the eigenvectors, we can see from equation (12) and the definition of $\mathbf{U}$ that $\mathbf{W} = \mathbf{U}$. Consequently, we can rewrite equation (11) as

$$\mathbf{Y} = \mathbf{U^T X} \tag{6.14}$$

Substituting equation (6.1) into equation (6.14), we obtain

$$\mathbf{Y} = \mathbf{U}^T\mathbf{U}\Sigma\mathbf{V}^T = \Sigma\mathbf{V}^T \tag{6.15}$$

The principal components contained in the matrix $\mathbf{Y}$ may be viewed as the inner product of the eigenvectors of $\mathbf{XX^T}$ with the data, or as the weighted eigenvectors of $\mathbf{X^T X}$.

Since $\mathbf{X}$ may be reconstructed from the principal component matrix $\mathbf{Y}$ by the inverse KL transformation

$$\mathbf{X} = \mathbf{UY} \tag{6.16}$$

we may combine last two equations to obtain

$$\mathbf{X} = \mathbf{U\Sigma Y^T} \tag{6.17}$$

Last equation is identical with equation (6.1), showing that, providing we are considering a multivariate stochastic process, the SVD and the KL transformation are computationally equivalent.

6.2 Eigenimage analysis of common offset sections

We investigate the application of eigenimage analysis to common offset sections. Our principal goal is to show that often, common offset sections can be efficiently compressed using eigenimages. A subsidiary goal is to improve the S/N ratio of pre-stack data by eigenimage filtering of common offset sections.

We consider the data matrix X to be composed of n_x traces with n_t data points per trace, the n_x traces forming the columns of X. The Singular Value Decomposition (SVD) of X (Lanczos, 1961), is given by:

$$X = \sum_{i=1}^{r} \lambda_i \, u_i \, v_i^T \,, \tag{6.18}$$

where r indicates the rank of the matrix X, u_i is the i-th eigenvector of $X\,X^T$, v_i is the $i-$th eigenvector of $X^T\,X$ and λ_i is the i-th singular values of X. Andrew and Hunt (1977) called the outer product $u_i\,v_i^T$ the i-th eigenimage of the matrix X.

Suppose that X represents a seismic section and that all the n_x traces are linearly independent. In this case the matrix X is of full rank and all the singular values are different from zero. A perfect reconstruction of X requires all eigenimages. On the other hand, in the case where all n_x traces are equal to within a scale factor, all traces are linearly dependent, X is of rank one and may be perfectly recovered by the first eigenimage, $\lambda_1 u_1\, v_1^T$.

The eigenimage decomposition can be used to optimally extract laterally coherent waveforms. In general, common offset sections exhibit a good lateral coherence. Our approach in this paper is to first decompose the pre-stack data cube into common offset sections and then apply eigenimage analysis to compress each common offset section and improve the S/N ratio.

Our strategy is summarized as follows:

1. The pre-stack data cube is decomposed into common offset sections, in our examples we construct 10 common offset sections containing traces with offsets indicated in Table 1.

2. Each common offset section is decomposed into eigenimages. 3- Only the eigenvectors that correspond to the first p singular values are kept.

3. Equation (6.18) is used to reconstruct the common offset section. If the misfit is acceptable, we save the vectors u_i, v_i, λ_i, $i = 1 \dots p$.

It is interesting to note that the amount of data compression that can be achieved using this procedure is remarkably high. Using the SVD we can represent each common offset section by n_2 floats:

$$n_2 = p \times n_t + p \times n_x + p \,.$$

We define the compression ratio as follows

$$C = (n_1 - n_2)/n_2,$$

where $n_1 = n_x \times n_t$ is the total number of floats required to represent the common offset section, X.

In Table 1 we summarize the compression ratio for the ten common offset sections in which we have decomposed the data cube. In this example p corresponds to the number of singular values that account for 30% of the total power encountered in the spectrum of singular values. In Figure 6.51 we portray the spectra of singular values. We note that the eigen-decomposition is in terms of a few energetic singular values that correspond to coherent events in the common offset domain.

COS#	Offset [m]	p	$C = (n_1 - n_2)\,/\,n_2$
1	0-221	9	13.7
2	221-427	6	20.0
3	427-633	4	18.7
4	633-839	5	17.8
5	839-1045	4	23.7
6	1045-1250	5	21.2
7	1250-1456	6	18.2
8	1456-1662	6	14.4
9	1662-1868	6	15.2
10	1868-2780	7	13.0

Table 6.1: Compression ratios for 10 common offset sections. The variable p indicates the number of singular values used in the eigen-decomposition.

In Figures 6.6 and 6.7 we display the common offset section #2 after and before eigenimage filtering. Since the evenst are fairly flat, we can always retain the information content of the section in a few eigenimages. compression and S/N ratio enhancement

In Figures 6.8 and 6.9 we display a CDP after and before performing the eigenimage analysis in common offset domian. It is clear that we cannot use eigenimages in the CDP domain, but after filtering in the common offset domain an sorting in CDPs we note that some high frequency noise at near offset traces was eliminated.

In summary, by sorting the data into common offset section we have been able to apply the eigenimage analysis on individual common offset traces. The pre-stack volume is reconstructing with a minimal distortion.

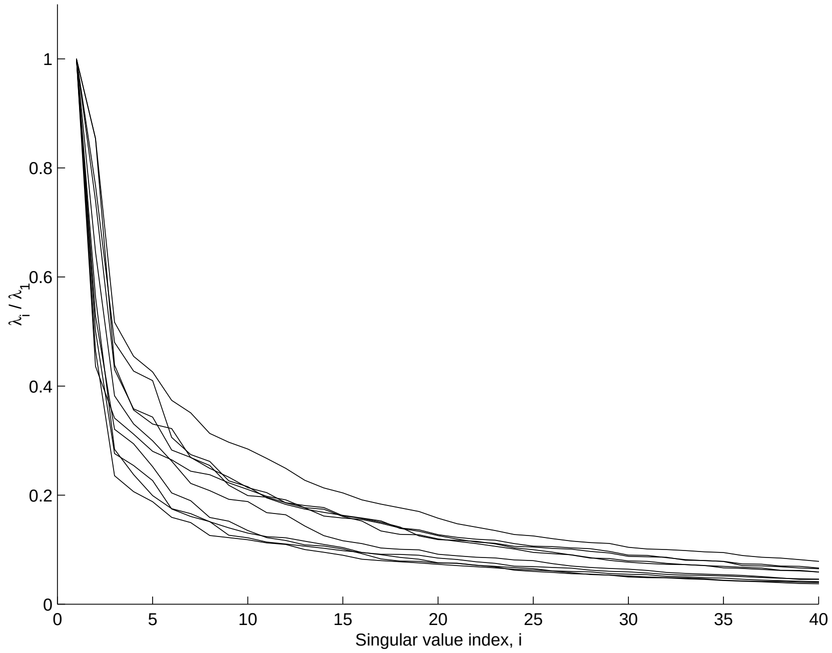


Figure 6.5: Spectra of singular values for the 10 common offset sections used to test the algorithm.

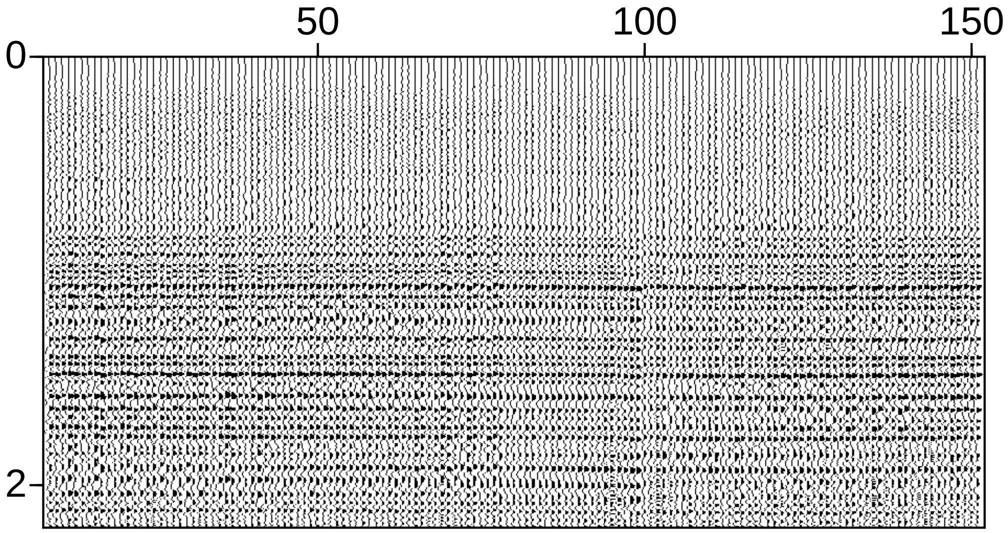


Figure 6.6: Common offset section #2.

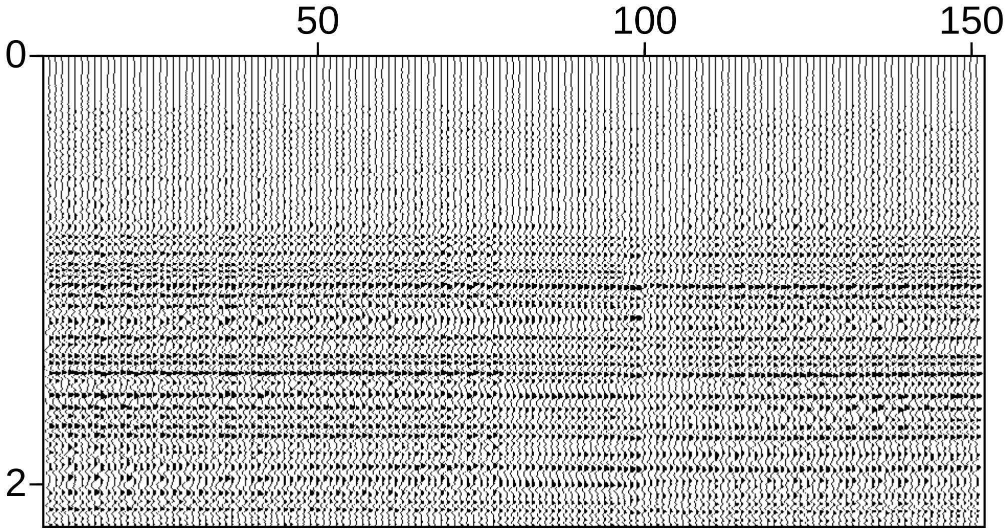


Figure 6.7: Common offset section #2 after eigenimage filtering

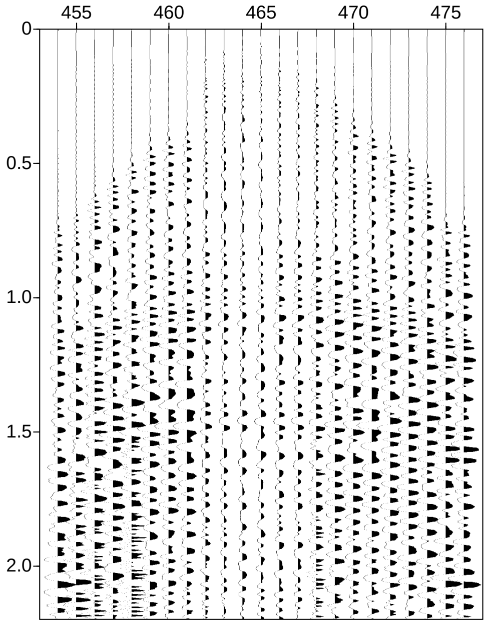


Figure 6.8: Original CDP.

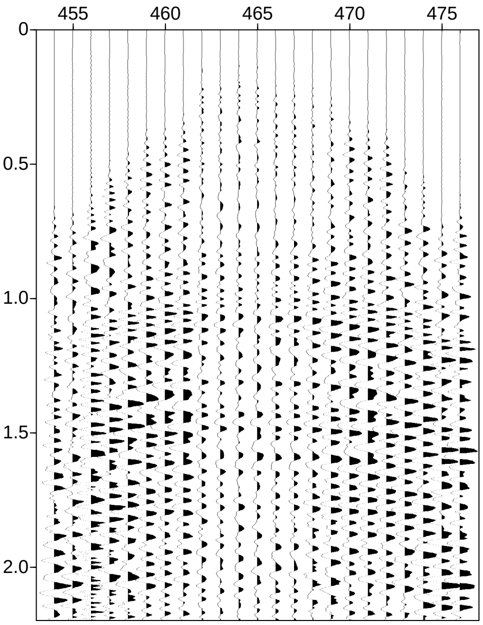


Figure 6.9: CDP after Eigenimage filtering in the common offset domain

6.2.1 Eigenimages and application to Velocity Analysis

Eigen-decomposition of seismic data (hyperbolic windows in CMP gathers) can be used to design coherence measures for high resolution velocity analysis. The idea is to replace the semblance measure by a norm that is a function of the eigenvalues of the covariance matrix of the gate of analysis. In this section we will derive a very simple algorithm that can be used to compute high resolution coherence measures for velocity analysis.
Techniques that exploit the eigen-structure of the covariance matrix have been borrowed from the field of array processing (Bienvenu and Kopp, 1983; Wax et al., 1984), and applied to velocity analysis by different researchers (Biondi and Kostov, 1989; Key and Smithson, 1990; Kirlin, 1992).
The seismic signal, in the presence of noise, at receiver i may be modeled using the following equation:

$$x_i(t) = s(t-\tau_i) + n_i(t) \qquad i = 1, N \,, \tag{6.19}$$

where $\tau_i = (t_0^2 + d_i^2/v^2)^{1/2} - t_0$ is the delay of the signal between the i-th receiver and a receiver having $d_0 = 0$. If a waveform is extracted along a hyperbolic path parametrized with velocity v, equation (6.19) may be rewritten as

$$x_i(t) = s(t) + n_i(t) \qquad i = 1, N \,, \tag{6.20}$$

noindent where, to avoid notational clutter, I used the same variable $x(t)$ to designate the delayed waveform (equation (6.19)) and the corrected waveform (equation (6.20)). The covariance matrix of the the signal is defined as:

$$R_{i,j}(t) = E[x_i(t)x_j(t)] \qquad i, j = 1, N \,, \tag{6.21}$$

where E denotes the expectation operator. If we assume the noise and signal to be uncorrelated the data covariance matrix becomes:

$$R_{i,j}(t) = R_{si,j}(t) + \sigma_n^2(t)\delta_{i,j} \,, \tag{6.22}$$

where $R_{si,j}(t)$ denotes the signal covariance matrix, and $\delta_{i,j} = 1$, if $i = j$ and $\delta_{i,j} = 0$, otherwise. Assuming a stationary source and a stationary noise process, we may drop the dependence on t. It is easy to verify that the eigenvalues of the covariance matrix become

$$\lambda_i = \lambda_{si} + \sigma_n^2 \qquad i = 1, 2, \ldots, N\,, \tag{6.23}$$

where λ_{si} are the eigenvalues of the signal covariance matrix. Assuming that the signal is invariant across each trace, the signal covariance matrix is rank 1, and we can write the following relationships:

$$\begin{aligned} \lambda_{s1} &= N.P_s \\ \lambda_{si} &= 0 \qquad i = 2, ..., N\,, \end{aligned} \tag{6.24}$$

where $P_s = E[s(t)^2]$ denotes the signal power. Using equation (6.23), the eigenvalues of the data covariance matrix become

$$\begin{aligned} \lambda_1 &= N.P_s + \sigma_n^2 \\ \lambda_i &= \sigma_n^2 \quad i = 2, ..., N\,. \end{aligned} \tag{6.25}$$

For uncorrelated noise, the minimal $N - 1$ eigenvalues of the data are equal to the variance of the noise. The largest eigenvalue is proportional to the power of energy of the coherent signal plus the variance of the noise.

In real situations, the eigen-spectrum is retrieved from an estimate of the data covariance matrix. If the stationary random processes $x_i(t)$ and $x_j(t)$ are ergodic the ensemble averages defined in equation (6.21) can be replaced by time averages (see for instance, Bendat and Piersol, 1971). The estimator of the covariance matrix becomes:

$$\hat{R}_{i,j} = \frac{1}{2M+1} \sum_{k=-M}^{M} x_i(k\Delta t) x_j(k\Delta t)\,. \tag{6.26}$$

Using the results given in equations (6.24) and (6.25) it is evident that an estimator of the noise variance is

$$\hat{\sigma}_n^2 = \frac{1}{N-1} \sum_{i=2}^{N} \hat{\lambda}_i\,. \tag{6.27}$$

Similarly, an estimator of the signal energy is given by

$$\hat{P}_s = \frac{\hat{\lambda}_1 - \hat{\sigma}_n^2}{N} \,, \tag{6.28}$$

and equations (6.27) and (6.28) can be combined into a single measure, the signal-to-noise-ratio:

$$\hat{C} = \frac{1}{N} \frac{\hat{\lambda}_1 - \sum_{i=2}^{N} \hat{\lambda}_i/(N-1)}{\sum_{i=2}^{N} \hat{\lambda}_i/(N-1)} \,. \tag{6.29}$$

The coherence measure, $\hat{C}$, was devised assuming the presence of a signal and that the proper velocity is used to extract the waveform. In general the coherence ,$\hat{C}$, is computed for different gates and different trial velocities. It is convenient to explicitly emphasize the dependence of the coherence on these parameters by denoting $\hat{C}(t_0, v)$. When the gate of analysis contains only noise, the measure $\hat{C}(t_0, v)$ tends towards zero. When the trial velocity does not match the velocity of the reflection, it is not possible to decompose the eigen-structure of the data into signal and noise contributions. In this case, the covariance matrix has a complete set of eigenvalues different from zero; therefore it is not possible to recognize which part of the eigen-spectrum belongs to the noise and which belongs to the signal process. Key and Smithson (1990) proposed another coherence measure based on a log-generalized likelihood ratio which tests the hypothesis of equality of eigenvalues,

$$\hat{W}_{ml} = M \log^N \left[\frac{(\sum_{i=1}^{N} \hat{\lambda}_i/N)^N}{\prod_{i=1}^{N} \hat{\lambda}_i} \right] \,. \tag{6.30}$$

In the absence of signal, $\lambda_i = \sigma_n^2$, $i = 1, N$ and hence $W_{ml} = 0$. In the presence of a single reflected signal, $\lambda_1 \neq 0$, $\lambda_i = 0$, $i = 2, N$ and $W_{ml} \to \infty$. Therefore, W_{ml} provides a strong discrimination between signal and noise. Key and Smithson (1990) combined equation (6.29) and (6.30) into a single measure, K_{ml}, given by the product:

$$\hat{K}_{ml} = \hat{W}_{ml} \, \hat{C} \,. \tag{6.31}$$

It is important to point out that only one eigenvalue, λ_1, is required to estimate the coherence measure, $\hat{C}$. Since

$$Trace(\hat{\mathbf{R}}) = \hat{\lambda}_1 + \hat{\lambda}_2 + \ldots + \hat{\lambda}_N \tag{6.32}$$

where

$$Trace(\hat{\mathbf{R}}) = \sum_{i=1}^{N} \hat{R}_{i\,i}\,. \tag{6.33}$$

It is easy to see from equations (6.32) and (6.33) that only $\hat{\lambda}_1$ is needed to compute the coherence measure, $\hat{C}$.

It is also important to mention that the velocity panel obtained via the SNR coherence measure can be further improved by adopting a bootstrap procedure (Sacchi, 1998). In this case, the seismic traces are randomly sampled to produce individual estimates of the coherence measure. From this information one can obtained an average coherence measure and a histogram (in fact a density kernel estimator) of the position of the peak that optimizes the coherence. The improve SNR coherence obtained with this techniques is portrayed in Figure (6.11).

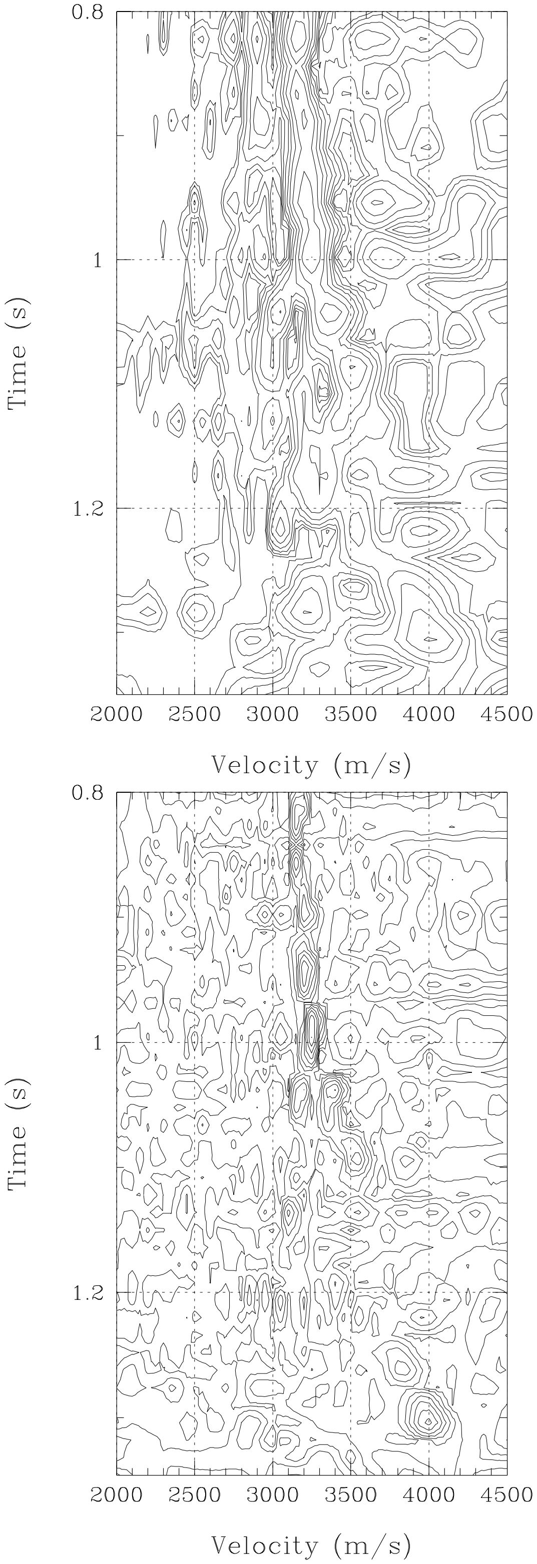


Figure 6.10: Left: Semblance of a CMP gather. Right: High resolution coherence analysis (SNR measure).

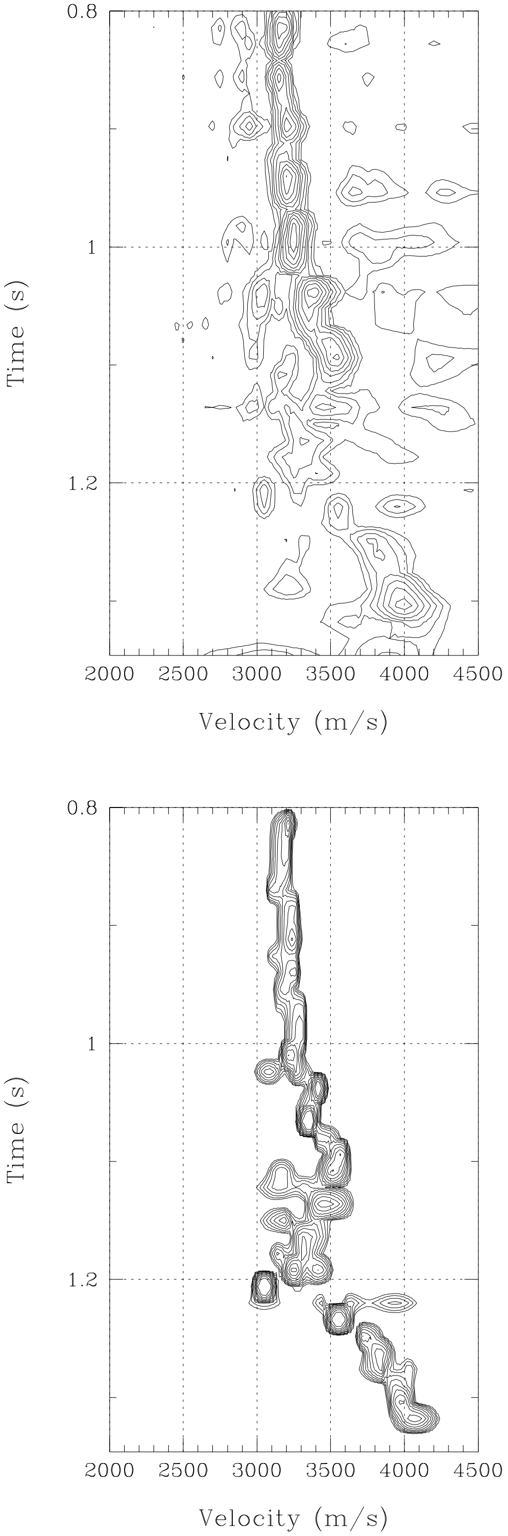


Figure 6.11: Left: Average SNR measure obtained via bootstrapping individual realizations. Right: Frequency distribution of the peak that maximizes the coherence after 50 bootstrap realizations.

6.3 A Matlab Code for Eigenimage Analysis

```
% A Code to filter data using the Eigenimage approach

% Generation of the model. Single
% event with parabolic moveout.

dt = 4./1000; w = ricker(20.,dt); nw=max(size(w));
nx = 32; nt = 128;
DATA = zeros(nx,nt);

for i=1:nx
 for j=1:nw
   c= fix(0.05*i*i);
  DATA(i,20+j+c) = w(j);
 end
end

% Add noise to the model
NOISE = 0.2 * randn(nx,nt);
DATA = DATA + NOISE;


[U S V] = svd(DATA);

% Reconstruction with 3 eigenimages

 p = 4;    % Keep 1,2,3.
 q = min(size(S));
 for i = p:q;
 S(i,i) = 0;
 end
```

```
% Filtered image
 DATA = U*S*V';
```

6.3.1 References

Andrews, H. C., and Hunt, B. R., 1977, *Digital image restoration*, Prentice-Hall, Signal Processing Series.

Bienvenu, G., and Kopp, L., 1983, Optimality of high resolution array processing using the eigensystem approach: IEEE, Trans. Acoust., Speech, Signal Processing., **ASSP-31**, 1235-1248.

Biondi, B. L., and Kostov, C., 1989, High-resolution velocity spectra using eigenstructure methods: Geophysics, **54**, 832-842.

Key, S. C., and Smithson, S. B., 1990, New approach to seismic-event detection and velocity determination: Geophysics, **55**, 1057-1069.

Kirlin, R. L., 1992, The relationship between the semblance and the eigenstructure velocity estimators, Geophysics: **57**, 1027-1033.

Freire, S.L.M and Ulrych, T.J., 1988, Application of singular value decomposition to vertical seismic profiling, Geophysics, **53**, 778-785.

Hemon, C.H. and Mace, D., 1978, Essai d'une application de la transformation de Karhunen-Loève au traitement sismique, Geophysical Prospecting, **26**, 600-626.

Jones, I.F. and Levy, S., 1987, Signal-to-noise ratio enhancement in multichannel seismic data via the Karhunen-Loève transform, Geophysical Prospecting, **35**, 12-32.

Lanczos, C., 1961, *Linear Differential operators*, D. Van Nostrand Co.

Sacchi, M.D., 1998, A bootstrap procedure for high-resolution velocity analysis: Geophysics, **65**, 1716-1725.

Chapter 7

Radon Transforms

In this chapter we deal with the numerical implementation of the Radon transform. We will analyze the problem using the inverse problem formalism and study the problem of designing a high resolution Parabolic Radon transform for multiple attenuation.

7.1 Slant Stacks

Different techniques have been devised to identify and/or filter linear events. Generally, they have the following common framework. First, they assume that a set of linear events are recorded on an array with discrete and limited coverage. Secondly, they assume that the noise is uncorrelated with the signals. In geophysics, linear event identification has been an active field of research. Two classic examples are vertical seismic profiles (VSP) and slowness vector estimation in seismographic arrays for earthquake detection and location. In VSP processing, linear event detection-estimation is used to identify and separate the principal components of the VSP data: the up-going and the down-going waves.

A general strategy for event identification-estimation involves the following approach. First, the data are transformed to a new domain where each component may be isolated. Then, after masking the undesired components, the data are mapped back to the original domain retaining only the desired information.

In seismic processing, the Radon transform is commonly known as the $\tau - p$

(τ denotes time and p ray parameter) or slant stack transform. The original idea developed by Radon in 1917 (Deans, 1983), has provided a basic framework for many problems of image reconstruction in physics, astronomy, medicine, optics, non-destructive testing, and geophysics. In image processing, it is also called the Hough transform (Pratt, 1991), which may be regarded as a transformation of a line in Cartesian coordinate spac e to a point in polar coordinate space.

In geophysics, the properties of the Radon transform were examined by Phinney et al. (1981), Durrani and Bisset (1984) and Tatham (1984). Chapman (1981) developed exact formulas for a point source in Cartesian or spherical coordinates, and for a line source in cylindrical coordinates. The relationship between the Radon transform and the plane wave decomposition is also well established (Stoffa et al., 1981; Treitel et al., 1982). Least squares procedures to compute the Radon transform were investigated by Thorson and Claerbout (1985), Beylkin (1987) and Kostov (1990). These authors showed how to mitigate the smearing caused by the finite aperture. Recently, Zhou and Greenhalgh (1994) linked the least squares solution to p-dependent Wiener filters. These researchers derived the slant stack formulas in the continuous domain, but the resulting algorithms are identical to those obtained by other researchers (Beylkin, 1987; Kostov, 1990).

In order to avoid the inversion of prohibitively large matrices the problem may be posed in the frequency-space domain ($f-x$). This technique was adopted by Beylkin (1987), Kostov (1990), Foster and Mosher (1992), and recently by Zhou and Greenhalgh (1994). This allows us to solve several small problems in the band that comprises the signal. Some stability concerns arise when the problem is tackled in this manner. Particularly, a least squares solution can be extremely unstable at low frequencies. In addition, it is interesting that slant stacks can be also computed in the time-space domain. Thorson and Claerbout (1985) and, recently Yilmaz and Tanner (1994), have presented high resolution least squares slant stack operators designed in time-space domain. Their procedures use an iterative inversion scheme especially devised to solve large linear sparse operators. Thorson and Claerbout (1985) have also shown how to update in each iteration the variances of the model to drive the solution to minimum entropy. Yilmaz and Taner (1994) have also developed an interesting scheme based on fuzzy logic to mitigate the alias.

7.1.1 The slant stack operator (conventional definition)

Let $u(h,t)$ represent a seismic signal. Throughout this chapter the variable t designates the time and h the offset or range. For a continuous array we define the slant stack by means of the following transformation

$$v(p,\tau) = (\mathcal{L}u)(p,\tau) = \int_{-\infty}^{\infty} u(h, t = \tau + h\,p)dh\,. \tag{7.1}$$

Where p and τ denote the slope or ray parameter and the intercept time, respectively. $v(p,\tau)$ is used to designate the signal in the $\tau - p$ domain. The adjoint transform $\mathcal{L}^*$ is given by

$$\tilde{u}(h,t) = (\mathcal{L}^* v)(p,\tau) = \int_{-\infty}^{\infty} v(p, t = \tau - hp)dp\,. \tag{7.2}$$

In the frequency domain, the pair of transformations are given by,

$$V(p,\omega) = \int_{-\infty}^{\infty} U(h,\omega) e^{i\omega ph} dh, \tag{7.3}$$

$$\tilde{U}(h,\omega) = \int_{-\infty}^{\infty} V(p,\omega) e^{-i\omega ph} dp, \tag{7.4}$$

substituting, (2.3) into (2.4) yields

$$\tilde{U}(h,\omega) = \int_{-\infty}^{\infty} U(h',\omega) \int_{-\infty}^{\infty} e^{-i\omega p(h-h')} dp\, dh' \tag{7.5}$$

which may be written as follows

$$\tilde{U}(h,\omega) = U(h,\omega) * \rho(h,\omega)\,, \tag{7.6}$$

where $*$ denotes convolution and the function ρ is given by

$$\rho(h,\omega) = \int_{-\infty}^{\infty} e^{-i\omega ph} dp\,, \tag{7.7}$$

making the substitution $z = -\omega p$ equation (7.7) becomes

$$\rho(h,\omega) = \int_{-\infty}^{\infty} \frac{1}{|\omega|} e^{ihz} dz = \frac{2\pi}{|\omega|}\delta(h)\,. \tag{7.8}$$

The convolution operator is a delta function with respect to the variable h. Using the property of the δ function,

$$\begin{aligned} \tilde{U}(h,\omega) = &\ \tfrac{2\pi}{|\omega|} U(h,\omega) * \delta(h) \\ = &\ \tfrac{2\pi}{|\omega|} U(h,\omega) \end{aligned} \tag{7.9}$$

the inversion formula becomes,

$$U(h,\omega) = \frac{|\omega|}{2\pi}\tilde{U}(h,\omega)\,. \tag{7.10}$$

The inverse is computed in two steps. First, the adjoint is used to evaluate $\tilde{U}(h,\omega)$. Then, $\tilde{U}(h,\omega)$ is multiplied by the frequency response of the ρ filter. The conventional slant stack pair in the frequency domain results in,

$$\begin{aligned} V(p,\omega) &= \int_{-\infty}^{\infty} U(h,\omega) e^{i\omega ph} dh, \\ U(h,\omega) &= \frac{|\omega|}{2\pi} \int_{-\infty}^{\infty} V(p,\omega) e^{-i\omega ph} dp\,. \end{aligned} \tag{7.11}$$

Now, consider that the range of p is a finite interval, $p \in [-P, P]$. This case leads to the following ρ filter,

$$\rho(h,\omega) = \int_{-P}^{P} e^{-i\omega ph} dp = 2P\frac{\sin(\omega Ph)}{\omega Ph}\,. \tag{7.12}$$

Substituting (7.12) in (7.6),

$$\tilde{U}(h,w) = 2P\int_{-\infty}^{\infty} U(h',\omega)\frac{\sin(\omega P(h-h'))}{\omega P(h-h')} dh'. \tag{7.13}$$

It is evident that the data may be recovered after solving a deconvolution problem. Spatial deconvolution is required since the infinite range of the variable p is truncated to a finite range. The wavenumber response of the ρ filter has the following expression:

$$\begin{aligned}
\rho(k,\omega) &= \int_{-\infty}^{\infty} \rho(h,\omega) e^{ikh} dh \\
&= \int_{-\infty}^{\infty} \int_{-P}^{P} e^{-i(\omega p - k)h} dh dp \\
&= \int_{-P}^{P} \delta(\omega p - k) dp \\
&= \frac{1}{\omega} \int_{-\omega P}^{\omega P} \delta(k' - k) dk' \qquad (7.14) \\
&= \left\{ \begin{array}{lcl} \frac{1}{\omega} & & k \leq |\omega P| \\ 0 & , & \text{otherwise} \end{array} \right.
\end{aligned}$$

According to the last equation, the spatial deconvolution will be unstable if the wavenumbers in the data lie outside the range $[-\omega P, \omega P]$. Equation (2.14) also shows that the deconvolution is unstable at low frequencies.

7.1.2 The inverse slant stack operator

The definition of the forward slant stack operator and its adjoint may be changed to construct another slant stack pair,

$$u(h,t) = (\mathcal{L}^* v)(p,\tau) = \int_{-\infty}^{\infty} v(p, t = \tau - hp) dp \qquad (7.15)$$

$$\tilde{v}(p,\tau) = (\mathcal{L} u)(h,t) = \int_{-\infty}^{\infty} u(h, t = \tau + hp) dh, \qquad (7.16)$$

the pair of transformations can be posed in the frequency-offset domain,

$$U(h,\omega) = \int_{-\infty}^{\infty} V(p,\omega) e^{-i\omega ph} dp, \qquad (7.17)$$

$$\tilde{V}(p,\omega) = \int_{-\infty}^{\infty} U(h,\omega) e^{i\omega ph} dh\,. \tag{7.18}$$

Substituting, (7.17) into (7.18) yields,

$$\tilde{V}(p,\omega) = \int_{-\infty}^{\infty} V(p',\omega) \int_{-\infty}^{\infty} e^{-i\omega h(p-p')} dh dp'\,, \tag{7.19}$$

where now, the convolution is with respect to the variable p, and the convolutional operator is given by

$$\gamma(p,\omega) = \int_{-\infty}^{\infty} \frac{1}{|\omega|} e^{ihp} dh = \frac{2\pi}{|\omega|}\delta(p)\,. \tag{7.20}$$

The γ filter is a delta function with respect to the variable p. Therefore, equation (7.19) becomes,

$$\tilde{V}(p,\omega) = \frac{2\pi}{|\omega|} V(p,\omega) \tag{7.21}$$

or equivalently

$$V(p,\omega) = \frac{|\omega|}{2\pi}\tilde{V}(p,\omega)\,. \tag{7.22}$$

From the above derivation, it is clear that the ρ and the γ filters have the same frequency response. Finally, the slant stack pair becomes,

$$\begin{aligned} V(p,\omega) &= \frac{|\omega|}{2\pi}\int_{-\infty}^{\infty} U(h,\omega) e^{i\omega ph} dh, \\ U(h,\omega) &= \int_{-\infty}^{\infty} V(p,\omega) e^{-i\omega p} dp\,. \end{aligned} \tag{7.23}$$

Assuming that $h \in [-H, H]$ (finite aperture), the γ filter has the following structure

$$\gamma(p,\omega) = \int_{-H}^{H} e^{i\omega ph} dh = 2H\frac{\sin(\omega H p)}{\omega H p}\,. \tag{7.24}$$

Hence, $V(p,\omega)$ may be calculated by solving the following integral equation,

$$\tilde{V}(p,\omega) = 2H\int_{-\infty}^{\infty} V(p',\omega)\frac{\sin(\omega H(p-p'))}{\omega H(p-p')}dp'\,. \tag{7.25}$$

After a comparison of the slant stacks pairs, equations (7.11) and (7.23), it is clear that a deconvolution procedure is required in both cases. In the conventional slant stack transform, the deconvolution is necessary to recover the data from the $\tau - p$ space. In the inverse slant stack operator the deconvolution process is required to estimate the $\tau - p$ space. The truncation effect of the variable p may be alleviated by choosing the proper region of support of the transform. The truncation of the variable h is associated with the resolution of the transform and cannot be alleviated by simple means. Generally, both the variables h and p are truncated. Thus, deconvolution should be carried out in both the forward and inverse transform (Zhou and Greenhalgh, 1994). However, the range of p may be chosen in such a way that most of the energy in the signal lies within this range.

7.1.3 The sampling theorem for slant stacks

Assuming that the wavefield is evenly sampled according to $U(n\Delta h,\omega), n = 0, \pm 1, \pm 2, \ldots$, the relationship between the $\tau - p$ and the $h - t$ spaces is given by

$$U(n\Delta h,\omega) = \frac{|\omega|}{2\pi}\int_{-\infty}^{\infty} V(p,\omega)e^{-i\omega p n\Delta h}dp\,, \tag{7.26}$$

where $V(p,\omega)$ denotes the slant stack corresponding to a continuous wavefield $U(h,\omega)$. The integration domain can be decomposed into small subdomains as follows,

$$\begin{aligned} U(n\Delta h,\omega) &= \frac{|\omega|}{2\pi}\sum_{k=-\infty}^{\infty}\int_{-(2k-1)\frac{\pi}{\omega\Delta h}}^{(2k+1)\frac{\pi}{\omega\Delta h}} V(p,\omega)e^{-i\omega p n\Delta h}dp \\ &= \frac{|\omega|}{2\pi}\sum_{k=-\infty}^{\infty}\int_{-\frac{\pi}{\omega\Delta h}}^{\frac{\pi}{\omega\Delta h}} V(p+2k\frac{\pi}{\omega\Delta h},\omega)e^{-i\omega(p+2k\frac{\pi}{\omega\Delta h})n\Delta h}dp \end{aligned} \tag{7.27}$$

since $e^{-i2\pi nk} = 1 \; \forall n \, k$, the last equation may be written in the following form

$$U(n\Delta h, \omega) = \frac{|\omega|}{2\pi} \int_{-\frac{\pi}{\omega\Delta h}}^{\frac{\pi}{\omega\Delta h}} V_d(p, \omega) e^{-i\omega pn\Delta h} dp, \tag{7.28}$$

where the relationship between the slant stack of the continuous signal and the one corresponding to the sampled wavefield, $V_d(p, \omega)$, is given by

$$V_d(p, \omega) = \sum_{k=-\infty}^{\infty} V(p + 2k\frac{\pi}{\omega\Delta h}, \omega). \tag{7.29}$$

Thus, the discrete signal has an $\omega - p$ representation with support in the range $p \in [-\frac{\pi}{\omega\Delta h}, \frac{\pi}{\omega\Delta h}]$. The components with slope $p - 2\frac{\pi}{\omega\Delta h}, p + 2\frac{\pi}{\omega\Delta h}, p - 4\frac{\pi}{\omega\Delta h}, p + 4\frac{\pi}{\omega\Delta h}, \ldots$ will appear to have slope p and every slope outside the range $(-\frac{\pi}{\omega\Delta h}, \frac{\pi}{\omega\Delta h})$ will have an alias inside this range. If the continuous signal has all the components inside that range, the aliased components do not exist and therefore we can write $V_d(p, \omega) = V(p, \omega)$. It is clear from the above discussion that spatial sampling must be chosen so as to avoid the aliasing effect. If $P = P_{max} = -P_{min}$, the following relationship guarantees the absence of alias,

$$\Delta h \leq \frac{1}{2Pf_{max}}, \tag{7.30}$$

where $f_{max} = \omega_{max}/2\pi$ is the maximum temporal frequency of the seismic signal. The product $P\,f_{max}$ is also the maximum wavenumber. Similarly, if Δh is given, the maximum ray parameter that can be retrieved ed without alias is given by

$$P_{max} = \frac{1}{2\Delta h f_{max}}. \tag{7.31}$$

For a non-symmetric slant stack, $P_{max} \neq -P_{min}$, equation (7.30) is modified as follows (Turner, 1990),

$$\Delta h \leq \frac{1}{P' f_{max}}, \tag{7.32}$$

where $P' = |P_{max} - P_{min}|$.

7.2 Discrete slant stacks

Discrete versions of equations the continuous Radon pair are obtained by replacing integrals by summations and imposing finite limits. First, assume that the seismogram contains $N = L_f - L_n$ traces, where the indices L_f and L_n denote far and near offset traces respectively.

$$v(p, \tau) = (\mathcal{L}u)(p, \tau) = \sum_{l=L_n}^{L_f} u(h_l, \tau + h_l p)\Delta h_l, \tag{7.33}$$

where $\Delta h_l = (h_{l+1} - h_l)$ for $l = L_n, \dots, L_f - 1$. Similarly, we approximate the continuous Radon transform by the following expression

$$\tilde{u}(h, t) = (\mathcal{L}^* v)(\tau, p) = \sum_{j=J_{min}}^{J_{max}} v(h, t - hp)\Delta p_j \tag{7.34}$$

where $\Delta p_j = (p_{j+1} - p_j)$ for $j = J_{min}, \dots, J_{max} - 1$. Taking the Fourier transform of the above equations yields

$$V(p, f) = \sum_{l=L_n}^{L_f} U(h_l, f) e^{2\pi i f h_l p} \Delta h_l \tag{7.35}$$

$$\tilde{U}(h, f) = \sum_{j=J_{min}}^{J_{max}} V(p, f) e^{-2\pi i f h p_j} \Delta p_j \,. \tag{7.36}$$

Using matrix notation it is possible to rewrite the slant stack and its adjoint as follows (f is omitted to avoid notational clutter),

$$\mathbf{m} = \mathbf{L}^H \mathbf{d} \tag{7.37}$$

$$\tilde{\mathbf{d}} = \mathbf{L}\mathbf{m} \tag{7.38}$$

The operators $\mathbf{L}$ and $\mathbf{L^H}$ form an adjoint pair. The matrix $\mathbf{L}$ is the forward operator and $\mathbf{L}^*$ denotes the adjoint operator. The vector $\mathbf{m}$ indicates the Radon space $V(p, f)$ at discrete values of p and fix frequency f, whereas the vector $\mathbf{d}$ indicates the data $U(h, f)$ at discrete values of h and fix frequency f.

7.2.1 The discrete slant stack operator (conventional definition)

The slant stack operator, equation (7.37), maps the $t - x$ space into the $\tau - p$ domain; the adjoint, equation (7.38), maps the $\tau - p$ domain into the $t - x$ domain. It is clear that since $\mathbf{L}$ is non-orthogonal $\mathbf{L}$ and $\mathbf{L}^H$ do not constitute an inverse pair. Given $\mathbf{m} = \mathbf{L}\mathbf{d}$, the problem is how to recover $\mathbf{d}$. A relationship between $\mathbf{d}$ and $\tilde{\mathbf{d}}$ is obtained after substituting (7.37) into (7.38)

$$\tilde{\mathbf{d}} = \mathbf{L}\,\mathbf{L^H}\,\mathbf{d}\,. \tag{7.39}$$

Equation (2.41) is uniquely invertible in $f \in B$ provided that $\det(\mathbf{L}\,\mathbf{L^H}) \neq \mathbf{0}$ in the band B,

$$\begin{aligned} \mathbf{d} = &\ (\mathbf{L}\,\mathbf{L}^H)^{-1}\tilde{\mathbf{d}} \\ = &\ \mathbf{G}^{-1}\tilde{\mathbf{d}}\,. \end{aligned} \tag{7.40}$$

The $N \times N$ matrix $\mathbf{G} = \mathbf{L}\,\mathbf{L}^H$ represents a discrete version of the ρ filter. The pair of transformations which map a signal from $f - h$ to $f - p$ and vice-versa is given by

$$\begin{aligned} \mathbf{m} = &\ \mathbf{L}\mathbf{d} \\ \mathbf{d} = &\ \mathbf{G}^{-1}\mathbf{L}\mathbf{m}\,. \end{aligned} \tag{7.41}$$

The vector $\mathbf{m}$ always exists since it is obtained by means of a simple mapping. Both expressions constitute an inverse pair when the inverse of $\mathbf{G}$

exits. The forward and inverse pair does not permit to adequately model the signal when additive noise is present. If the data are contaminated with noise, the noise is mapped to the Radon domain.

7.2.2 The least squares solution

Assume that the data is the result of applying a Radon operator (slant stack) to a **m**.

$$\mathbf{d} = \mathbf{L}\mathbf{m} \tag{7.42}$$

The idea is to find **m** such that the following objective function is minimized (Yilmaz, 1994),

$$J = ||\mathbf{d} - \mathbf{L}\mathbf{m}||^2 \tag{7.43}$$

The solution to this problem is the least squares solution

$$\mathbf{m} = (\mathbf{L}^H\mathbf{L})^{-1}\mathbf{L}^H\mathbf{d} \tag{7.44}$$

In general the inverse needs to be stabilized using a damping parameter.

$$\mathbf{m} = (\mathbf{L}^H\mathbf{L} + \mu\mathbf{I})^{-1}\mathbf{L}^H\mathbf{d} \tag{7.45}$$

In general, this is the approach that it is used to compute slant stacks and parabolic Radon transfomrm. Other techniques to improve the resoltion of these operators where proposed by Sacchi and Ulrych (1995).

7.2.3 Example

In Figure (7.2) we display 3 panels. The first panel is the ideal $\tau - p$ signal; the second panel is the $\tau - p$ signal transformed to $offset - time$. These are two linear events with positive and negative slope. The third panel (left) is the inverted $\tau - p$ signal using least-squares. It is clear that artifacts have been created in the inverted $\tau - p$. These artifacts are generated by alias. As we have already seen f_{max}, the maximum ray parameter and the spatial sampling must satisfied a Nyquist condition (equation (7.30)). This condition is not satisfied and, therefore, the $\tau - p$ domain exhibits alias.

In Figure (7.2) I muted the the $\tau - p$ domain eliminating all the contribution where $p > 0$. The muted $\tau - p$ domain is used to reconstruct the data, this is displayed in the Figure (**??**) [Left]. This procedure can be used to discriminate down-going and up-going wavefields in Vertical Seismic Profiles (VSP).

Figures (7.3) and (7.4) displayed a simulation similar to the one described above but now the original signal in $t - offset$ has spectral components that are contained in the $5 - 35$Hz band. In other words I have eliminated the alias artifacts.

7.3 Parabolic Radon Transform (Hampson, 1986)

This is a simple modificarion to the slant stack, instead of integrating along curves of the form

$$t = \tau + ph$$

we use curves of the type

$$t = \tau + qh^2$$

.

This is a good approximation to process data containing hyperbolic events after NMO correction Parabolic Radon Transform are utilized to removed multiple reflections. After NMO correction the moveout of the primaries is

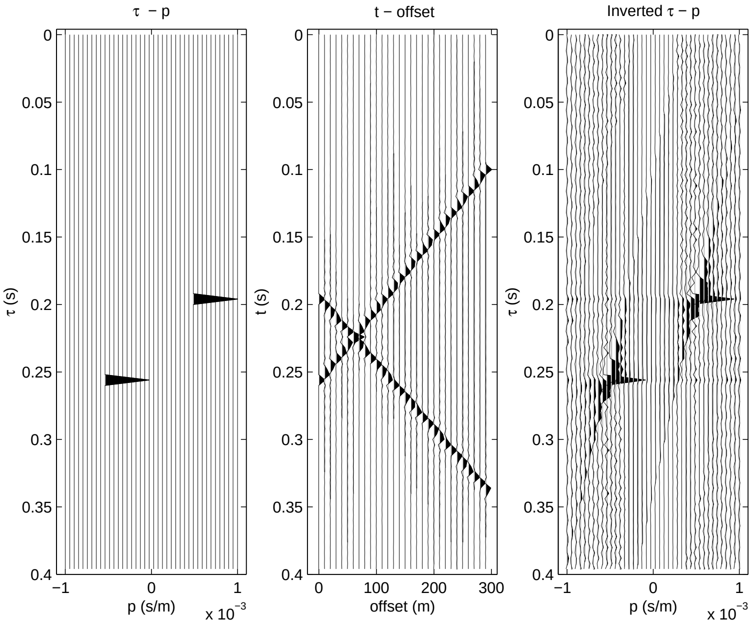


Figure 7.1: Left: Ideal $\tau - p$ panel. Center: Data generated by forward transforming the ideal $\tau - p$ panel. Right: Inverted $\tau - p$ panel.

zero. The residual move out of the multiples follows in a first order approximation a parabolic moveout. The transform is used to isolated multiples from primaries in order to mute (filter) them out.

Assume that in a CMP (common mid point gather) you have two events: a primary and a multiple. Let us assume that the intercept time of these events T_0 is the same.

The travel-time curve for the primary is given by:

$$T_p = \sqrt{(T_0^2 + h^2/v_p^2)} \tag{7.46}$$

,

and the travel-time for the multiple is given by:

$$T_m = \sqrt{(T_0^2 + h^2/v_m^2)}\,. \tag{7.47}$$

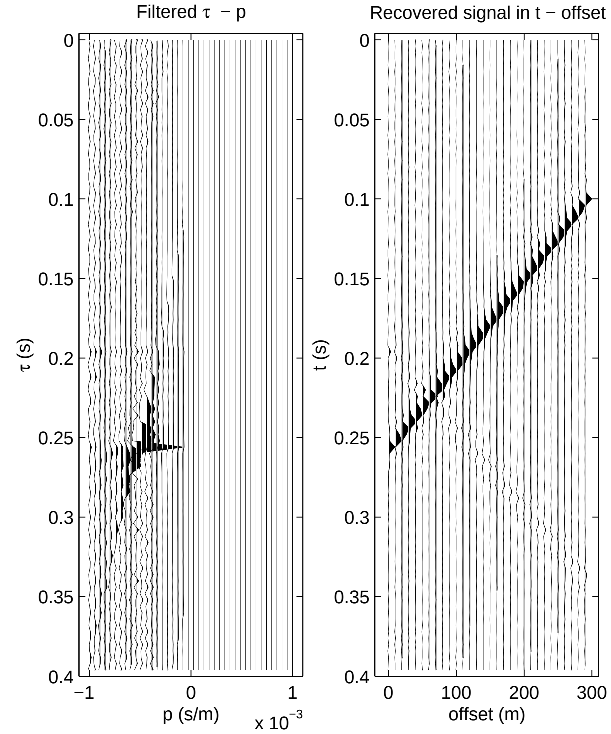


Figure 7.2: Left: Inverted $\tau - p$ panel after muting. Right: Data reconstructed by forward modeling the inverted/muted $\tau - p$ panel.

If the multiple is generated by a shallow layer or by the water column we can consider $v_p > v_m$.
Now suppose that we apply NMO correction to the complete data set with the NMO law that uses the velocity of the primary. The NMO correction entails applying the following time shift to the data

$$\Delta T_{NMO} = T_0 - \sqrt{(T_0^2 + h^2/v_{NMO}^2)}\,, \tag{7.48}$$

therefore, the time of the primary after NMO is

$$T_p(After) = T_p + \Delta T_{NMO} \tag{7.49}$$

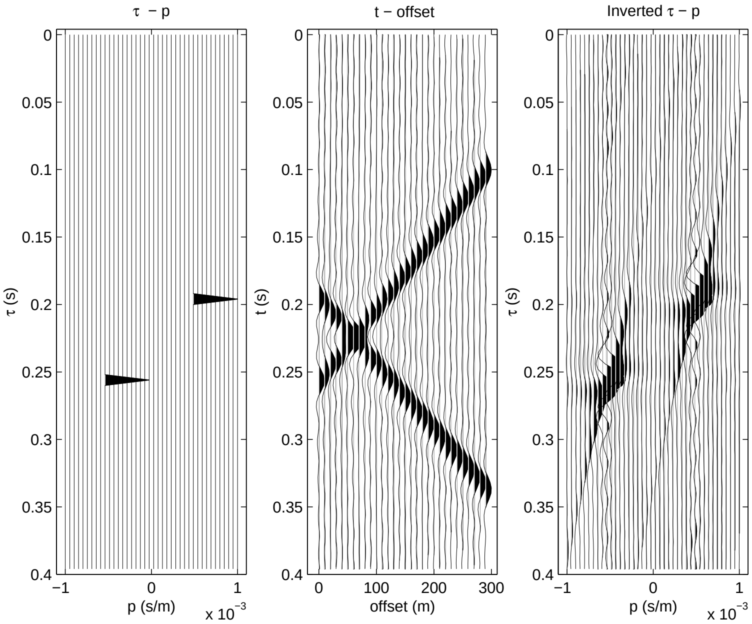


Figure 7.3: Left: Ideal $\tau - p$ panel. Center: Data generated by forward transforming and band-limiting the ideal $\tau - p$ panel. Band-limiting is needed to eliminate alias. Right: Inverted $\tau - p$ panel.

It is clear that is the NMO velocity is the velocity of the primary, the time of the primary becomes

$$T_p(After) = T_0 \,. \tag{7.50}$$

In other words, the primary has the same time for all offsets (a flat event). What is the time of the multiple after NMO?. Let us try to compute it,

$$T_m(After) = T_m + T_0 - \sqrt{(T_0^2 + h^2/v_{NMO}^2)} \,, \tag{7.51}$$

or after replacing T_m

$$T_m(After) = T_0 + \sqrt{(T_0^2 + h^2/v_m^2)} - \sqrt{(T_0^2 + h^2/v_{NMO}^2)} \tag{7.52}$$

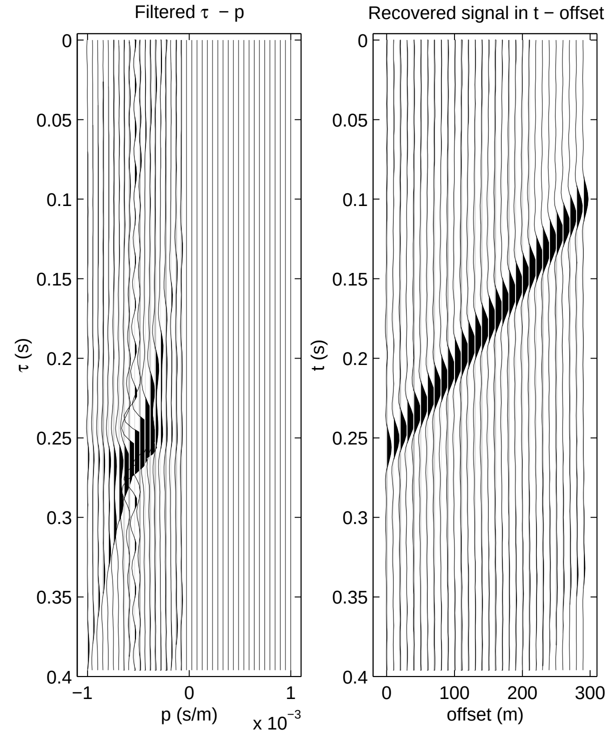


Figure 7.4: Left: Inverted $\tau - p$ panel after muting. Right: Data reconstructed by forward modeling the inverted/muted $\tau - p$ panel. Note that the alias artifacts have disappeared.

The two square roots in the above equation can be expanded in Taylor series (Keeping only up to the second order term) we have

$$T_m(After) \approx T_0 + \frac{1}{2T_0\, v_m^2} h^2 - \frac{1}{2T_0\, v_{NMO}^2} h^2 \tag{7.53}$$

which can be re-written as

$$T_m(After) \approx T_0 + qh^2 \tag{7.54}$$

where

$$q = \frac{1}{2\,T_0}(\frac{1}{v^2} - \frac{1}{v_{NMO}^2})\,. \qquad (7.55)$$

It is clear that a transform with parabolic integration path can be constructed by simple interchanging in the original slant stack h by h^2. Some people prefer to parameterize the parabola in terms of the residual moveout time at far offset,

$$t = \tau + q\frac{h^2}{h_{max}^2}\,,$$

then it is clear that the parameter q is nothing else that the moveout in seconds at the far offset trace.

In Figures (7.56) and (7.57) we portrayed a primary and a multiple before and after parabolic Radon transform filtering. In this example q is residual moveout at far offset.

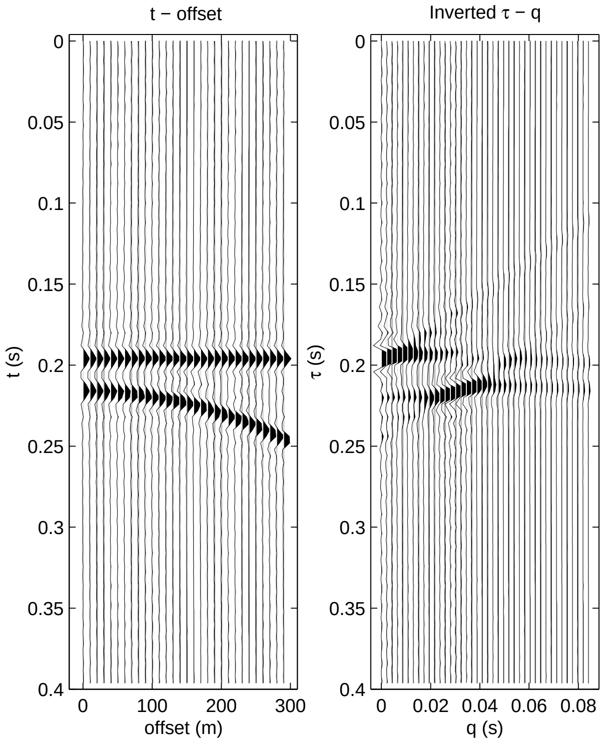


Figure 7.5: Left:A Primary and a multiple after NMO correction. Inverted $\tau - q$ panel.

7.4 High resolution Parabolic Radon Transform

The high resolution Parabolic Radon transform proposed by Sacchi and Ulrych (1995) entails the utilization of a regularization technique that leads to an operator that does not exhibit a Toeplitz structure. In the original formulation of the high resolution Radon transform the operator is inverted using Cholesky decomposition. This is quite expensive compared to the classical least squares Radon transform that uses the Levinson recursion to invert a Toeplitz form.

We propose a method to achieve high resolution at a computational cost of the order of the conventional parabolic least squares Radon transform. This feature makes our new algorithm quite attractive to process large data sets. The Parabolic Radon transform is a widely accepted technique for multiple

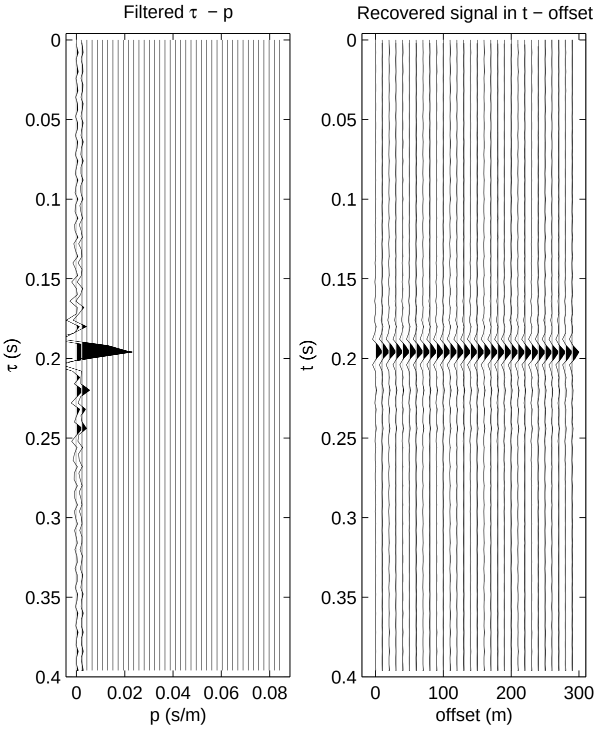


Figure 7.6: Left: Inverted $\tau - q$ panel after muting. Right: Data reconstructed by forward modeling the inverted/muted $\tau - q$ panel. In this example the multiple has been eliminated by muting in the $\tau - q$ domain.

removal (Hampson, 1986). The technique can be implemented in the frequency domain via a fast algorithm that exploits the Toeplitz structure of the least squares Radon operator (Kostov, 1990; Darche, 1990). Recently, Sacchi and Ulrych (1995) proposed a high resolution algorithm to increment the ability of the transform to distinguish events with similar moveout curves. This algorithm is based on a procedure that attempts to find a sparse representation of the reflections in the parabolic Radon domain. A similar algorithm has been proposed by Cary (1998). In this case the Radon panel is constrained to be sparse in both the Radon parameter and the intercept time. The high resolution parabolic Radon transform can be used to isolate multiples interferences with a few milliseconds of residual moveout at far offset. This is a problem frequently encountered when dealing with short period multiple reflections generated by carbonate targets in the Western Canadian Basin (Hunt et al., 1996).

One of the advantages of the high resolution parabolic Radon transform is that the focusing power of the transform is considerably increased with respect to the classical least squares parabolic Radon transform. Unfortunately, the high resolution parabolic Radon transform leads to the inversion of an operator that is Hermitian but does not exhibit a Toeplitz structure. The resulting Hermitian operator is inverted using Cholesky decomposition. The Cholesky method for solving Hermitian linear systems of equations requires a number of operations that is proportional to M^3, where M is the dimension of the Hermitian operator.

7.4.1 Least squares Parabolic Radon transform

Common mid point (CMP) gathers after normal moveout (NMO) correction can be modeled as a superposition of events with parabolic moveout:

$$d(x_j,t) = \sum_{k=1}^{M} m(q_k, \tau = t - q_k\, x_j^2)\,, j = 1, N\,, \tag{7.56}$$

where $d(x_j,t)$ denotes the CMP gather, x_j the offset, $m(q_k,\tau)$ is the Radon panel, q_k the discrete Radon parameter and τ the intercept time. The data con sist of N seismic traces which do not need to be regularly sampled. The Radon parameter is uniformly discretized according to $q_k = q_0 + \Delta q\,(k - 1)$, $k = 1, \dots, M$.

Equation (7.56) is essentially a decomposition of the CMP gather in terms of parabolic events distributed in the plane τ, q. It is computational more convenient to rewrite the last equation in the frequency-offset domain. Taking Fourier transform with respect to the temporal variable t we arrive to the following expression

$$d(x_j,f) = \sum_{k=1}^{M} m(q_k,f)\, e^{i2\pi f q_k x_j^2}\,, j = 1, \dots, N\,. \tag{7.57}$$

The calculations can be carried out independently for each frequency f. Equation (7.57) can be written in matrix form as follows:

$$\mathbf{d}(f) = \mathbf{L}(f)\,\mathbf{m}(f)\,. \tag{7.58}$$

To avoid notational clutter we will drop the frequency dependency in equation (7.58) and write $\mathbf{d} = \mathbf{L}\,\mathbf{m}$.
The least squares Radon operator is estimated by minimizing the following cost function.

$$J = ||\mathbf{d} - \mathbf{L}\,\mathbf{m}\,||^2 + \mu||\mathbf{m}||^2\,. \tag{7.59}$$

The regularization term $\mu||\mathbf{m}||^2$ is used to control the roughness of the solution. It can be shown that this term is one of the major sources of amplitude smearing in the Radon panel (Sacchi and Ulrych, 1995).
Taking derivatives of J with respect to $\mathbf{m}$ and equating them to zero yields

$$\begin{aligned}(\mathbf{L}^H\,\mathbf{L} + \mu\mathbf{I})\mathbf{m} &= \mathbf{L}^H\mathbf{d} \\ &= \mathbf{m}_{adj}\,.\end{aligned} \tag{7.60}$$

In the last equation $\mathbf{m}_{adj}$ denotes the low resolution Radon transform obtained using the adjoint or transpose operator $\mathbf{L}^H$. The least squares solution becomes

$$\begin{aligned}\mathbf{m} &= (\mathbf{L}^H\,\mathbf{L} + \mu\mathbf{I})^{-1}\mathbf{m}_{adj} \\ &= (\mathbf{R} + \mu\mathbf{I})^{-1}\mathbf{m}_{adj}\,.\end{aligned} \tag{7.61}$$

At this point some observations are in order. First it is clear that $\mathbf{R} = \mathbf{L}^H\,\mathbf{L} + \mu\mathbf{I}$ is a Toeplitz form (Kostov, 1990), with elements given by

$$\{\mathbf{R} + \mu\mathbf{I}\}_{l,m} = \sum_{k=1}^{N} e^{-i2\pi f\Delta q(l-m)x_k^2} + \mu\delta_{l,m}\,. \tag{7.62}$$

Solving this equation using the Levinson recursion requires approximately $4M^2 + 7M$ operations, and storage of only the first row of the Toeplitz matrix (Marple, 1987). This feature yields to a very efficient algorithm to compute the parabolic Radon transform.

7.4.2 High resolution parabolic Radon transform

In the high resolution parabolic Radon transform the vector $\mathbf{m}$ is retrieved by solving the following equation:

$$(\mathbf{R} + \mathbf{W}^H\mathbf{W})\mathbf{m} = \mathbf{m}_{adj} \,. \tag{7.63}$$

The matrix $\mathbf{W}$ is a diagonal matrix with elements that depend on $\mathbf{m}$ (Sacchi and Ulrych, 1995). This leads to an iterative algorithm where $\mathbf{W}$ is bootstrapped from the result of a previous iteration. In general, the iterative procedure is not required if we are able to design $\mathbf{W}$ from a priori information. The matrix of weights $\mathbf{W}$ is a diagonal matrix with elements given by

$$\{\mathbf{W}\}_{l,m} = w_l \, \delta_{l,m}, \;\; l, m = 1, \dots, M \,. \tag{7.64}$$

The elements of the diagonal form $\mathbf{R} + \mathbf{W}^H\,\mathbf{W}$ become:

$$\{\mathbf{R} + \mathbf{W}^H\,\mathbf{W}\}_{l,m} = \sum_{k=1}^{N} e^{-i2\pi f \Delta q (l-m) x_k^2} \; + \; w_l^2 \delta_{l,m} \,. \tag{7.65}$$

It is clear that the addition of a diagonal matrix with non-constant e elements has destroyed the Toeplitz structure of the operator. The above matrix can be inverted by the Cholesky method in a number of operations proportional to M^3. From the computational point of view it is more convenient to compute the Radon transform using a constant diagonal regularization (equation (7.60)). However, if we want to estimate a high resolution Radon operator, the regularization term must be a diagonal form with non-constant elements. The elements of $\mathbf{W}$ are used to emphasize the Radon parameters q_k that need to be constrained to be zero. In general, the matrix $\mathbf{W}$ is bootstrapped from the data in an iterative manner. The aforementioned procedure is described in Sacchi and Ulrych (1995).

In our synthetic example, the elements of the diagonal matrix $\mathbf{W}^H\mathbf{W}$ are given by

$$w_k^2 = \begin{cases} 100. & if \;\; q_k \notin Q \\ 0.0001 & if \;\; q_k \in Q \,, \end{cases} \tag{7.66}$$

where Q indicates the set of parameters q_k where the reflections are localized. These weights can be interpreted as the inverse of a variance in model space. If w_l^2 is large, $1/w_l^2$ is small and therefore, the algorithm will constraint the areas of no reflections in the τ, q space to be zero. It is clear that the resolution is enhanced by inhibiting the creation of smearing in the Radon panel.

7.4.3 Conjugate gradients and circulant matrices

To solve equation (7.63) we adopt the method of conjugate gradients, which is summarized below.
We want to solve $(\mathbf{R}+\mathbf{D})\mathbf{m} = \mathbf{m}_{adj}$, where $\mathbf{D} = \mathbf{W}^H\mathbf{W}$.
Start with an initial solution $\mathbf{m_0}$, set $\mathbf{p}_0 = \mathbf{r}_0 = \mathbf{m}_{adj} - (\mathbf{R}+\mathbf{D})\mathbf{m}_0$,
$\alpha_{i+1} = (\mathbf{r}_i, \mathbf{r}_i)/(\mathbf{p}_i, (\mathbf{R}+\mathbf{D})\mathbf{p}_i)$
$\mathbf{m}_{i+1} = \mathbf{m}_i + \alpha_{i+1}\mathbf{p}_i\ \mathbf{r}_{i+1} = \mathbf{r}_i - \alpha_{i+1}(\mathbf{R}+\mathbf{D})\mathbf{p}_i$
$\beta_{i+1} = (\mathbf{r}_{i+1}, \mathbf{r}_{i+1})/(\mathbf{r}_i, \mathbf{r}_i)$
$\mathbf{p}_{i+1} = \mathbf{r}_{i+1} + \beta_{i+1}\mathbf{p}_i$
where $i = 0, 1, 2, \ldots K$ denotes the iteration number.
The cost of the conjugate gradients algorithm is dominated by the cost of multiplying a matrix by a vector. In general, matrix times vector multiplication is an $O(M^2)$ process. In our problem we will use the Toeplitz structure of $\mathbf{R}$ to find a fast manner to compute the aforementioned operation.
The product $(\mathbf{R}+\mathbf{D})\mathbf{x}$ can be decomposed into two products: $\mathbf{Rx}+\mathbf{Dx}$. The first product can be efficiently computed using the Fast Fourier Transform (FFT), the second product involves only $2M$ operations (M products plus M additions) and does not substantially increase the computational cost of the inversion.
The first product, $\mathbf{y} = \mathbf{Rx}$, is evaluated by augmenting the system as follows:

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{y}' \end{bmatrix} = \mathbf{R}_{aug} \begin{bmatrix} \mathbf{x} \\ \mathbf{0} \end{bmatrix}, \tag{7.68}$$

where $\mathbf{R}_{aug}$ is the original Toeplitz matrix after being properly folded to become a circulant matrix (Strang, 1986; Schonewille and Duijndam, 1998). The right hand side can be computed by multiplying the Fourier transform of the first row of $\mathbf{R}_{aug}$ by the Fourier transform of vector $[\mathbf{x}\,, \mathbf{0}]^T$, and taking the inverse Fourier transform of this product. Now our matrix times vector operation takes $O(M' \log M')$ operations where M' is the size of augmented matrix ($M' = 2M$). We have found that the conjugate gradients algorithm convergences after a few iterations ($K \approx M/5$). Therefore, the inversion becomes an $O(K\,M'\,Log(M'))$ process. This is more efficient than the direct inversion of equation (7.63) by the Cholesky method.

7.4.4 Example

In Table 7.1 we present a comparison of CPU times in seconds for 3 different algorithms. The times in Table 1 correspond to the total computational cost for 512 frequencies. These simulations were performed on a SGI Origin 2000. In both cases we have 4 parabolic events which were mapped to the Radon domain u sing the following algorithms:

1. **Lev**: Classical least squares parabolic Radon transform implemented via the Levinson recursion (valid for a constant damping).

2. **Chol**: High resolution Radon transform implemented via the Cholesky decomposition.

3. **CG+FFT**: High resolution parabolic Radon transform implemented via conjugate gradients plus matrix times vector multiplication using the FFT.

Is is clear that the new algorithm can achieved high resolution at a computational cost comparable to the one of the classical least squares Radon transform computed with the Levinson recursive solution.
In Figure (7.1) we portray the results obtained for the 256 × 256 simulation. Note that the differences between the high resolution Radon transform computed with the Cholesky decomposition and the proposed algorithm are minimal.

7.5 Programs for Slant Stack and Parabolic Radon Transforms

The following two programs are a MATLAB implementation of the Radon transform in $f-x$ and $f-p$. The inverse transform is solved using Least

$N \times M$	**Lev**	**Chol**	**CG+FFT**
128×128	2	6	3
256×256	8	42	12

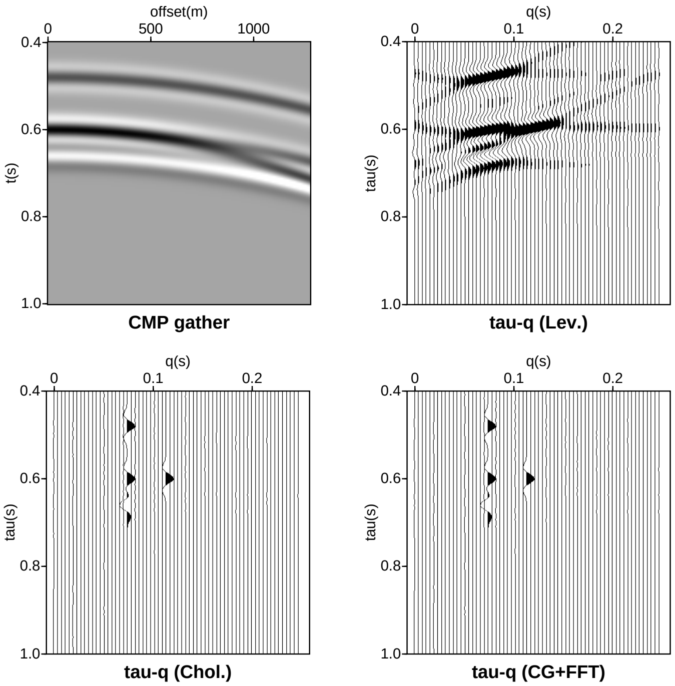
Table 7.1: CPU times in seconds for the 3 algorithms tested in this study. N denotes the number of traces and M the number of q parameters.

squares. The high resolution implementation using circulant matrices is a little bit more tricky and requires more than a few lines of Matlab.

Forward Transform

Operator to compute the forward linear and parabolic Radon transform.

```
function [d]=for_taup(m,dt,h,q,N,flow,fhigh);
%INV_TAUP    An inverse Radon transforms. Given the seismic data, this
%            function computes the Radon panel by inverting the Radon operator
%
%  [d] = for_taup(m,dt,h,q,N);
%
%
%  IN   m: the Radon panel (d(nt,nq)
%       dt: sampling in sec
%       h(nh) offset or position of traces in mts
%       q(nq) ray parameters to retrieve or curvature
%             of the parabola if N=2
%       N:1 Linear tau-p
%        :2 Parabolic tau-p
%       flow, fhig: min and max freq. in Hz
%
%  OUT  d: the data
%
%
%  SeismicLab
%  Version 1
```



CMP gather

tau-q (Lev.)

tau-q (Chol.)

tau-q (CG+FFT)

Figure 7.7: A synthetic CMP gather composed of 4 parabolic events is used to test 3 different algorithms to compute the Radon transform. **Lev.** indicates the classical solution using least squares with a constant damping term; the Levinson algorithm is used to invert the resulting Toeplitz form. **Chol.** indicates the high resolution solution using non-constant damping (8)), this solution is computed by means of the Cholesky decomposition. **CG+FFT** indicates the proposed fast algorithm to compute the high resolution Radon transform. In this example the size of the Radon operator is 256×256. CPU times in seconds are given in Table 7.1

```
%
% written by M.D.Sacchi, last modified December 10,  1998.
% sacchi@phys.ualberta.ca
%
% Copyright (C) 1998 Signal Analysis and Imaging Group
%                    Department of Physics
%                      The University of Alberta
%

nt= max(size(m));
nh = max(size(h));



M = fft(m,[],1);
D = zeros(nt,nh);
i = sqrt(-1);

ilow  = floor(flow*dt*nt)+1; if ilow<1; ilow=1;end;
ihigh = floor(fhigh*dt*nt)+1;
if ihigh>floor(nt/2)+1; ihigh=floor(nt/2)+1;end

for if=ilow:ihigh
f = 2.*pi*(if-1)/nt/dt;
L = exp(i*f*(h.^N)’*q);
x = M(if,:)’;
y = L * x;
D(if,:) = y’;
D(nt+2-if,:) = conj(y)’;
end
D(nt/2+1,:) = zeros(1,nh);
d = real(ifft(D,[],1));

return;
```

Inverse transform

Operator to compute the LS inverse Radon transform. Notice that this is an “academic” implementation. A fast implementation involves replacing inv by a fast solver (i.e., Levinson’s recursion).

```
function [m] = inv_taup(d,dt,h,q,N,flow,fhigh,mu);
%INV_TAUP   An inverse Radon transform. Given the seismic data,
%           this subroutine computes
%           the Radon panel by inverting the Radon operator
%
%  [m] = inv_taup(d,dt,h,q,N,flow,fhigh,mu)
%
%  IN   d: seismic traces  (d(nt,nh)
%       dt: sampling in sec
%       h(nh) offset or position of traces in mts
%       q(nq) ray parameters to retrieve or curvature
%             of the parabola if N=2
%       N:1 Linear tau-p
%        :2 Parabolic tau-p
%       flow:  freq.  where the inversion starts in HZ (> 0Hz)
%       fhigh: freq. where the inversion ends  in HZ   (> Nyquist)
%       mu: regularization parameter
%
%  OUT  m: the linear or parabolic tau-p panel
%
%
%  SeismicLab
%  Version 1
%
%  written by M.D.Sacchi, last modified December 10,  1998.
%  sacchi@phys.ualberta.ca
%
%  Copyright (C) 1998 Signal Analysis and Imaging Group
%                     Department of Physics
%                     The University of Alberta
%

nt= max(size(d));
nq = max(size(q));
nh = max(size(h));
```

```
D = fft(d,[],1);
M = zeros(nt,nq);
i = sqrt(-1);

ilow  = floor(flow*dt*nt)+1; if ilow<1; ilow=1;end;
ihigh = floor(fhigh*dt*nt)+1;
if ihigh>floor(nt/2)+1; ihigh=floor(nt/2)+1;end

for if=ilow:ihigh
f = 2.*pi*(if-1)/nt/dt;
L = exp(i*f*(h.^N)'*q);
y = D(if,:)';
x = L'*y;

MATRIX = L'*L;
tr=real(trace(MATRIX));
   Q =mu*tr*eye(nq);
   x = inv(MATRIX+Q) *L'* y;
M(if,:) = x';
M(nt+2-if,:) = conj(x)';
end
M(nt/2+1,:) = zeros(1,nq);
m = real(ifft(M,[],1));
return
```

7.6 Time variant velocity stacks

We will discuss in this section the computation of time variant operators that can be used as an alternative to the parabolic Radon transform.

The parabolic Radon transform is a time invariant operator, therefore it can be implemented in the frequency domain. This trick permits one to solve several small problems, one at each frequency, instead of a large problem involving all the time-offset-velocity samples at the same time.

It is clear that in the case of the parabolic Radon transform time invariance is achieved by means of an approximation. It might happen that the parabolic approximation is not properly satisfied and consequently, traveltimes (especially at far offsets) are not properly modeled.

In this part of the course, we will focus our attention of the computational aspects of Hyperbolic Radon operators.

We have already mentioned that the data in the CDP domain can be modeled as a superposition of hyperbolas. User this assumption a hyperbolic stack operator can be used to map hyperbolas (reflections) into $time - velocity$ pairs. In other words, our operator is used to map data from $offset - time$ to $velocity - time$ space. In the new space we can identify multiple reflection and filter them out. We can also use this type of operators to reduce random noise and to enhance the overall aspect of the seismic reflection which might be hidden by strong ground-roll (in a CSG) or any other type of deterministic noise.

The time-variant velocity-stack operator is defined in terms of summation along Dix hyperbolas, $m(\tau, v)$ is used to designate the velocity-stack and $d(t, h)$ the CMP gather:

$$d(t,h) = \int m(\tau = \sqrt{t^2 - h^2/v^2}, v)\, dv\,, \tag{7.69}$$

where h is source-receiver offset, t is two-way travel-time, v is the rms velocity, and τ is two-way vertical travel-time. After discretization and lexicographic arrangement, equation (7.69) can be written as

$$d = Lm\,. \tag{7.70}$$

The vectors d and m have $nt \times nh$ and $n\tau \times nv$ elements, respectively. The dimension of the operator L is $(nt \times nh) \times (nt \times n\tau)$. The forward or modeling operator, L, picks a wavelet in velocity space and produces a hyperbola in data space. The transpose operator L^T is a simple NMO followed by stacking operator.

To find the inverse operator we consider the problem

$$Minimize \; \{\phi = ||Lm - d||_2^2\}$$

Differentiating ϕ with respect to m yield the least-squares solution

$$\hat{m} = (L^T L)^{-1} L^T d = (L^T L)^{-1} m_0 \, , \tag{7.71}$$

where for simplicity we have assumed that L is full rank. In equation (7.71) m_0 is the low resolution velocity-stack computed by means of the adjoint or transpose operator (Sacchi and Ulrych, 1995). The velocity stack computed after inversion, $\hat{m}$, possesses more resolution that m_0. Unfortunately, the computation of $\hat{m}$ involves the inversion of $L^T L$. If we assume a typical CMP gather of 48 channels and 1000 samples per trace. In addition, suppose that 48 traces and 1000 samples were used to discretize the velocity, v, and the intercept time, τ, respectively. In this case $L^T L$ has dimension 24000×24000. It is evident that direct methods cannot be applied in this type of problems.

7.6.1 The conjugate gradients algorithm

The trick here is to use a semi-iterative technique to find an approximate solution to our problem. The advantage of the CG algorithm is the the matrix L does not need to be stored. In fact, L is not even a matrix but an operation perform on a vector. To apply the CG algorithm we need first to define the operations L and L^T.

It is clear that L is an operator that picks a wavelet in the $\tau - v$ and produces a hyperbola in $t - h$. The operator L^T (the adjoint or transpose operator) does the opposite, it gathers information in $t - h$ along a hyperbolic path and collapses this information into a point in $\tau - v$.

Let us assume that we have a code capable of performing the following operations (as I have already mentioned L and L' do not need to be matrices)

$$y = Lx \ \ x' = L^T y \, .$$

To solve the problem $||Lx - y||^2$ with an initial solution x_0, we use the following Conjugate Gradients (CG) algorithm:

```
Set initial values: $r = y - L\, x_0 \, , \, g = L^T\, r \, , \, s = g$
for i = 1:$ITERMAX$ {
  $ss = Ls \, , \ \delta = ||ss||^2$
  $\alpha = \gamma / \Delta$
  $x = x + \alpha\, s$
  $r = r - \alpha\, ss$
  $g = L^T\, r$
  $s = g + \beta\, s$
• }
```

The CG algorithm will find the least squares solution for the over-determined problem in N iterations where N is the total number of observations. In the under-determined problem, the CG converges to the minimum norm solution. The technique gives the exact answer for exact arithmetics, but of course round-off errors will affect the convergence of the algorithm. This is why the CG is often referred as a semi-iterative technique.

In the computation of the velocity stacks, we will use on a few iterations. How many iterations?. We can say, that we will use enough iterations to properly model the hyperbolic events. In fact, the CG method allows us to explore efficiently our solution by stopping the algorithm at any number of iteration and then, if the solution is not optimal, we can re-start the algorithm until a satisfactory misfit is obtained.

7.6.2 Example

We will analyze the performance of the CG with synthetic and real data examples. In Figure (7.8a) we portray a synthetic CMP gather. The model is composed of 2 primaries reflections of 1500m/s (water column) and a primary of 1700 m/s at 0.65 s. In Figure (7.8b) we portray the velocity gather obtained using the adjoint operator. It is clear that this gather does not offer enough resolution to properly identify and separete the multiple event at 0.65 s from the primary. In Figure (7.8c) we portray the velocity gather obtained after inverting the data using the CH algorithm. Figure (7.8d) is the primary obtained after muting the velocy gather.

In Figure (7.9) I displayed the velocty gather obtained via the CG algorithm after amplitude clipping. In this panel we also portray the artifacts that arise from finite aperture and sampling (alias).

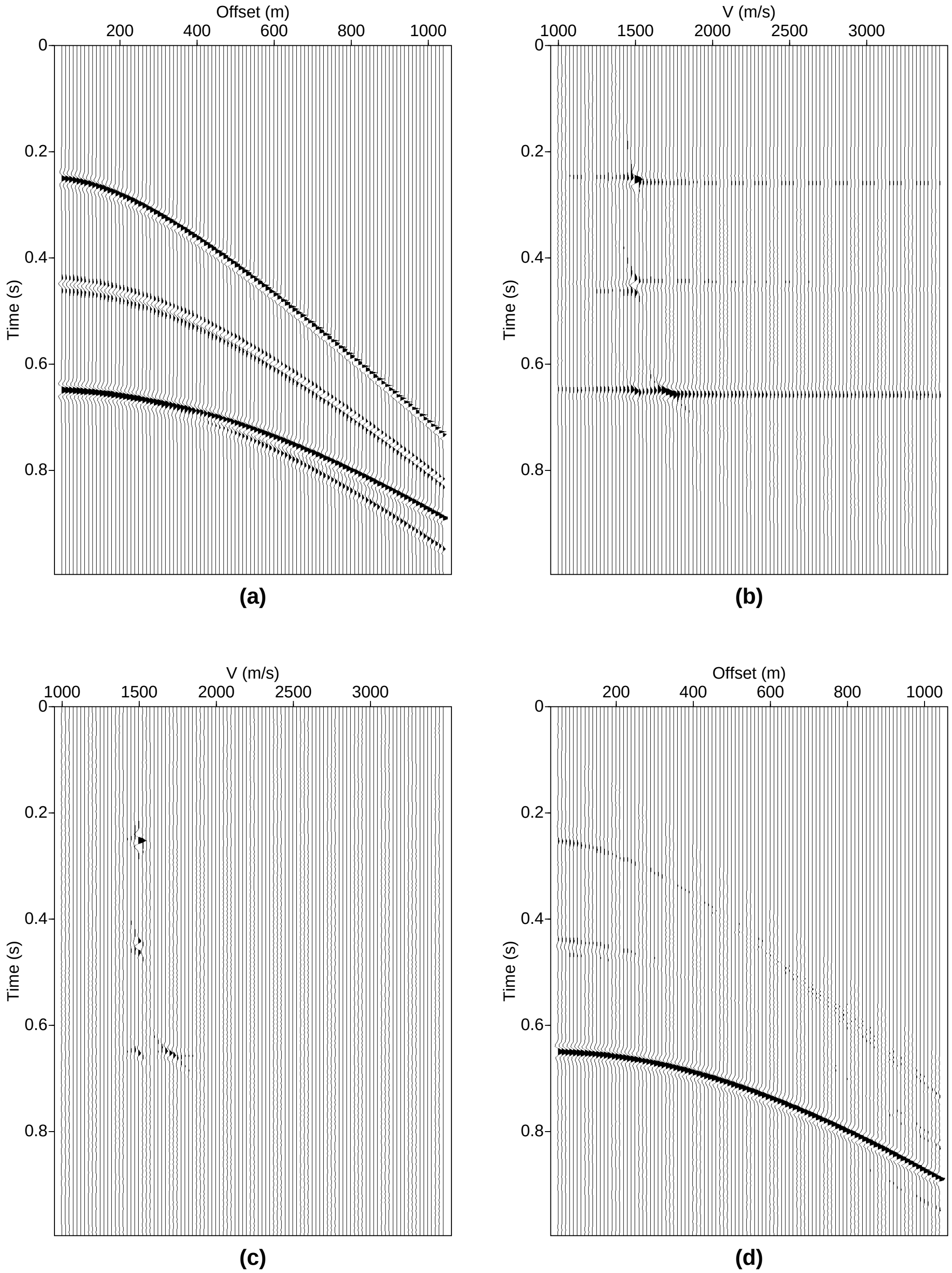
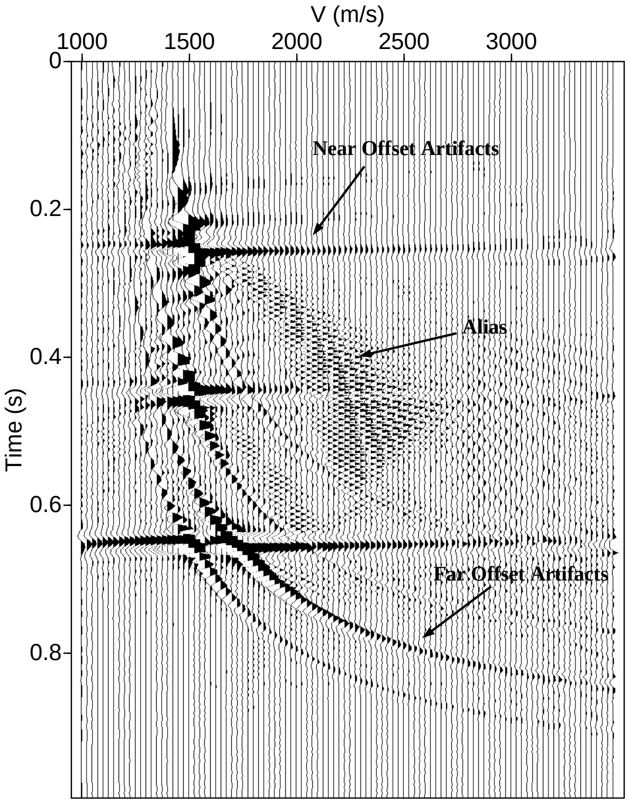


Figure 7.8: (a) Synthetic data. (b) Velocity gather obtained using the adjoint operator. (c) Velocity gather computed using the least-squares inversion. (d) Recovered data (primary) obtained after the de-multiple process.



(c)

Figure 7.9: Clipped version of Figure (7.8)b showing finite aperture and sampling (alias) artifacts.

The following subroutine permits to compute forward and adjoint Hyperbolic Radon operators. I have included a linear interpolation step. See how do I define the adjoint of interpolation. It is important to stress that the forward and adjoint pairs must pass the dot product test. Otherwise, the CG inversion will not work. Once you have your forward and adjoint operators encapsulated in a subroutine it's quite simple to put together a CG inversion code.

```
      subroutine Hyperbolic_Radon(dt,nt,v,nv,h,nh,m,d,c)
c
c  Compute velocity panels  when c = 'a' (Adjoint Hyper. Radon)
c  Compute CMP gathers when       c = 'f' (Forward Hyper. Radon)
c                                 c is character * 1
c INPUT
c  dt : sampling in sec
c  nt : number of time samples (also number of tau samples)
c  v(nv) : axis of the Radon panel (velocity in m/s). It can be
c          changed by 1/vel^2 or moveout at far offset
c  h(nh) : offset in meters
c          h(1) is offset of trace 1, h(2) is offset of trace 2....
c
c INPUT/OUTPUT
c  d(nh,nt) : cmp or super-cmp  input if c = 'a'
c  m(nv,nt) : Radon panel       output if c = 'a'
c
c  d(nh,nt) : cmp or super-cmp  output if c = 'f'
c  m(nv,nt) : Radon panel       input if c = 'f'
c

      real        d(300,2000), m(300,2000),h(300)
      real        v(300)

      character * 1 c
```

```
if(c.eq.'a') call clean(m,nv,nt)  ! initialize m with zeros
if(c.eq.'f') call clean(d,nh,nt)  ! initialize d with zeros

do ih =1,nh
 do iv=1,nv
  do itau=1,nt

   ttt=(itau-1)*dt
   time=sqrt(ttt**2+(h(ih)/v(iv))**2)
   it1 = int(time/dt)
   a = time/dt - float(it1)       !Coeff. of the linear interp.
   it2 = it1 + 1

if(it1.lt.nt.and.it1.ge.1) then
 if(c.eq.'a') m(iv,itau) = m(iv,itau)+(1.-a)*d(ih,it1)+a*d(ih,it2)
 if(c.eq.'f') d(ih,it1)  = d(ih,it1)+(1.-a)*m(iv,itau)
 if(c.eq.'f') d(ih,it2)  = d(ih,it2)+    a *m(iv,itau)
endif

  enddo    ! end offset loop
 enddo    ! end velocity loop
enddo    ! end tau loop

return
end
```

7.7 High Resolution Radon Transform

We can contruct a solution m that consists on a few isolated spikes in velocity space. This is what we often call a sparse solution. We have already outlined a procedure to compute sparse solutions using the Parabolic Radon transform. In that case the spareness constraint was used to invert the Radon operator in the frequency domain. When using hyperbolic Radon transforms, the sparseness constraint has to be imposed in the $\tau - v$ domain. In general one can use any measure of sparseness (we have seen various norms that can be used to retrieve sparse models when dealing with impedance inversion in Chapter 4). Let's assume that we use a Cauchy-like norm (Sacchi and Ulrych, 1995). In this case we minimize

$$J = ||Lm - d||_2^2 + \mu \sum_k \ln(1 + m_k^2/b) \qquad (7.72)$$

where m_k indicates an element of $m(\tau, v)$ after lexicographic arrangement (transformation of a matrix into a vector). The parameters μ and b are the hyper-parameters of the problem.
Taking derivatives of J with respect to m_k and equation them to zero leads to the following system

$$L^T L\, m - L^T d + Q\, m = 0 \qquad (7.73)$$

where Q is a diagonal matrix with elements given by

$$Q_i = \frac{2\,\mu}{b + m_i^2}$$

It is clear that the system needs to be solved in an iterative manner (Q depends on the unknown model m). We can rewrite our solution as follows:

$$m^k = (L^T L + Q^{k-1})^{-1} L^T d\,. \qquad (7.74)$$

where k indicates the iteration. The matrix of weights Q is computed from the result of the previous iteration. In general, one solve the problem for a given matrix of weights Q using CG, then after enough iteration to reach convergence, Q is updated and a CG is run again to solve the linear problem. The procedure is continues until we find the minimum of the cost function

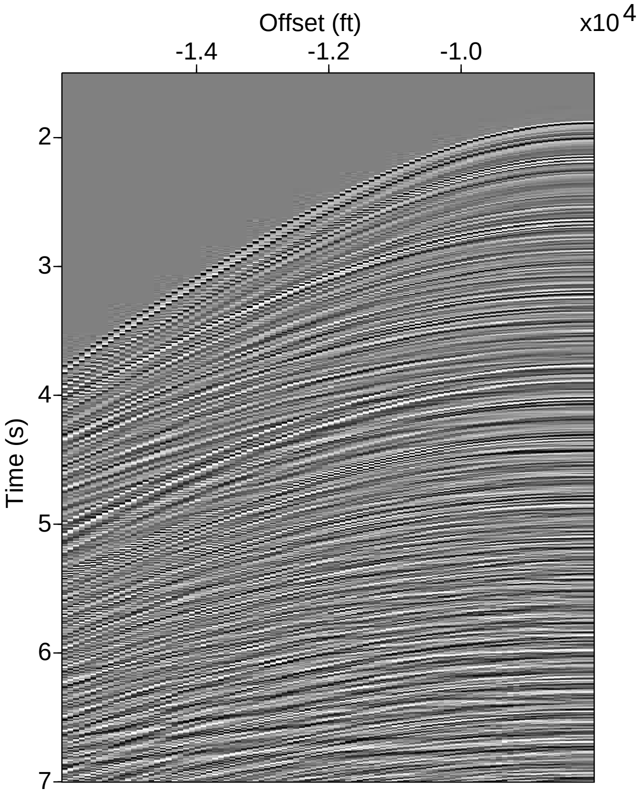


Figure 7.10: CMP gather # 1000 from a data set from the Gulf of Mexico.

J. This algorithm can be very expensive, and in general a good felling for the μ and b is required to reach a sparse solution. When working with real data the parameters needed for the inversion (μ, b, number of iterations) are estimated by trial an error from a single CMP gather, the same parameters are used to invert the rest of the CMPs in the seismic volume.
In Figures (7.10), (7.11), (7.12), and (7.13) we test the high resolution hyperbolic Radon transform with a data set from the Gulf of Mexico (data provided by Western Geophysical to test multiple attenuation codes).

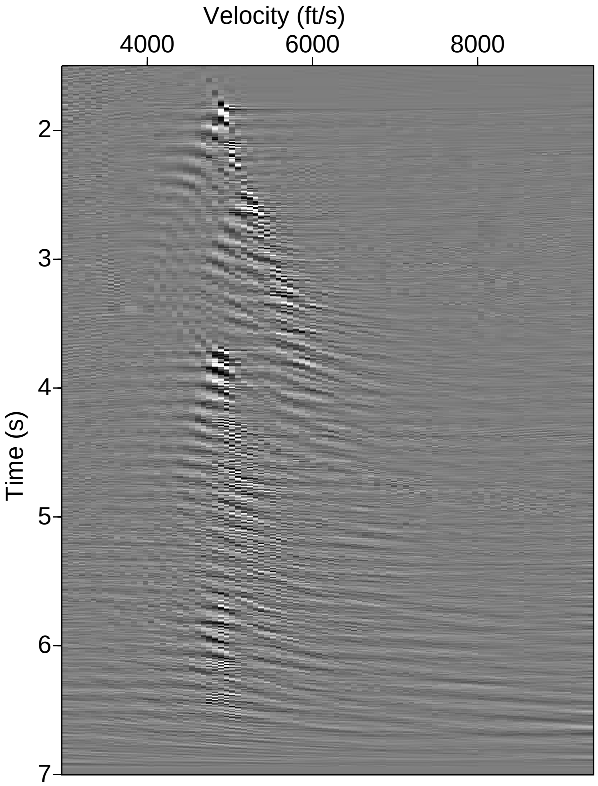


Figure 7.11: Velocity panel obtained by inversion of the Hyperbolic Radon transform using least-squares. CMP gather # 1000 from a data set from the Gulf of Mexico.

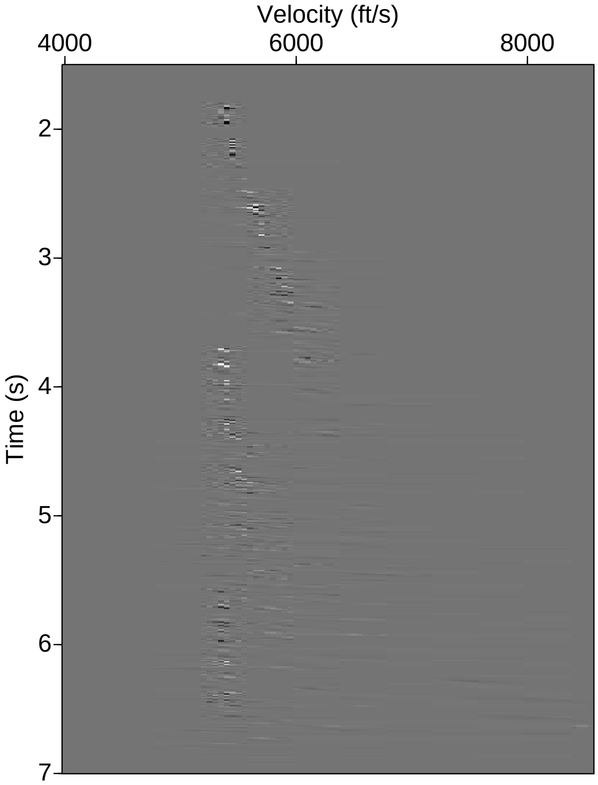


Figure 7.12: Velocity panel obtained by inversion of the Hyperbolic Radon

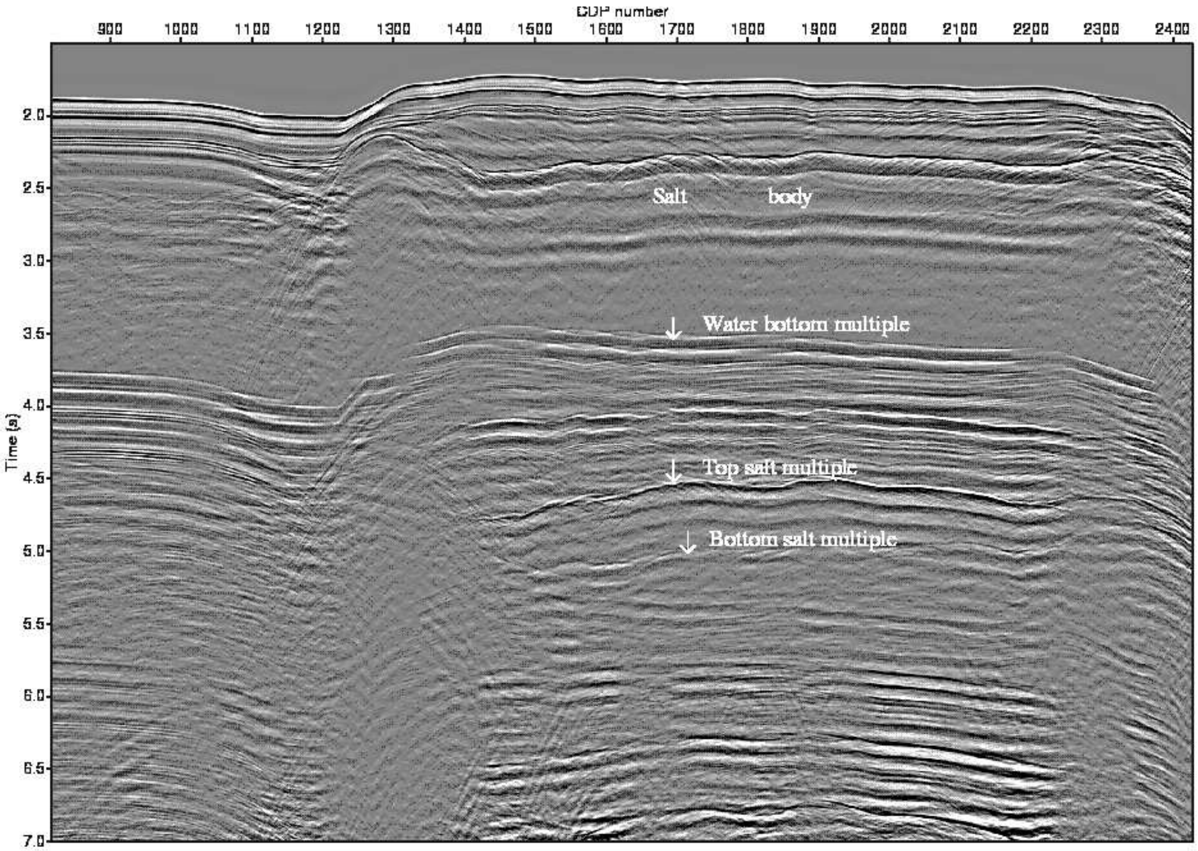


Figure 7.13: Stack section of the Gulf of Mexico data set before multiple removal.

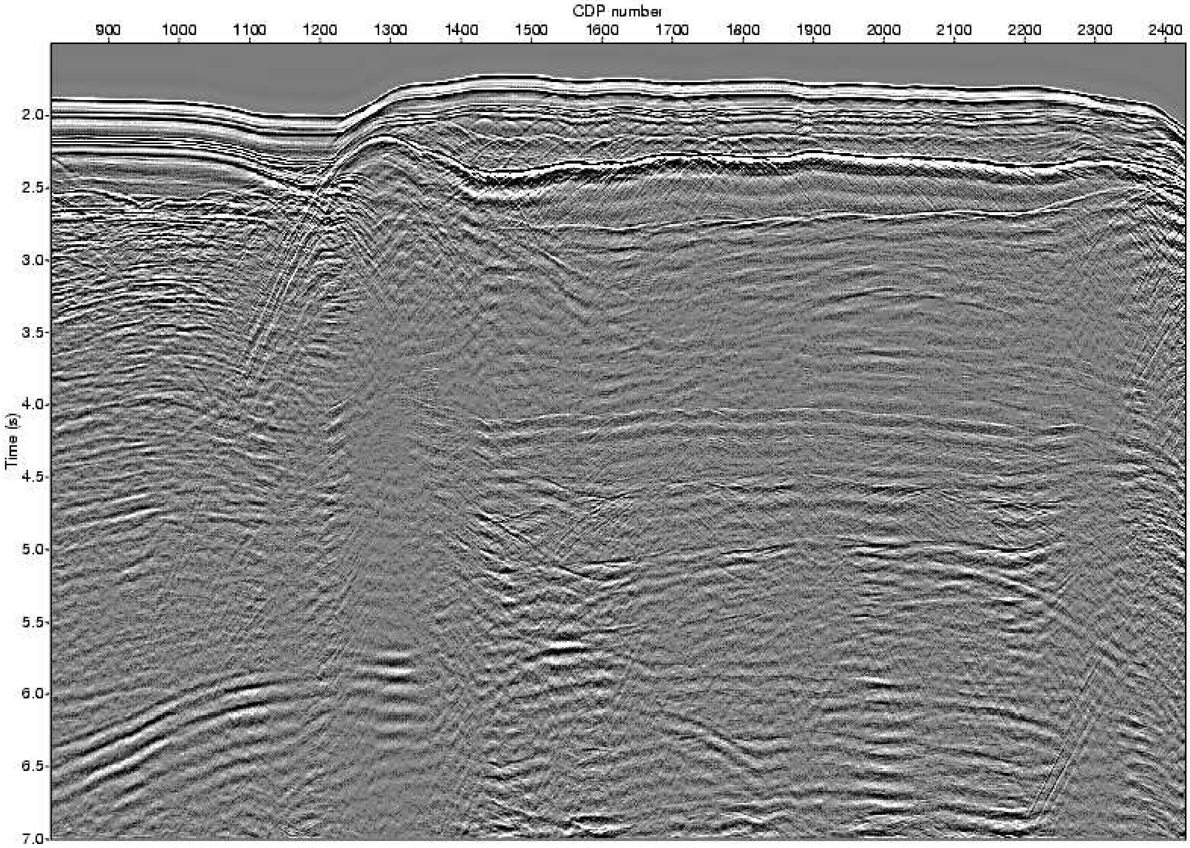


Figure 7.14: Stack section after multiple removal.

7.8 Interpolation problems

The parabolic and hyperbolic Radon transform can be used to interpolate CMP gathers. This is quite simple and, basically, entails mapping back the velocity stack to a data space using a new geometry. To interpolate pre-stack data in receiver-source space (or midpoint-offset) a more sophysicated approach is required. Various research group in the area of signal analysis have proposed algorithms to interpolate 1D data. These algorithms assume that the data are band-limited. One can extend these ideas to the problem of reconstructing pre-stack data. In geophysics Duijndam et. al (1999) and Hindriks et. al (1997) have introduced a least-squares algorithm to invert the fourier transform of the data. We will review some basic features of these algorithms and introduce a regularization term that enables us to recover large gaps in our pre-stack data set.
We define the discrete 2-D inverse Fourier transformation in source and receiver coordinates as

$$u(x_s, x_r, \omega) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} U(k_s(m), k_r(n), \omega) e^{jk_s(m)x_s} e^{jk_r(n)x_r}, \qquad (7.75)$$

where x_s and x_r are the spatial variables along source and receiver coordinates, k_s and k_r are the corresponding wave-numbers and ω is the temporal frequency. Equation (7.75) gives rise to a linear system equations

$$u = AU \qquad (7.76)$$

where

$$A_{mn} = \frac{1}{MN} e^{jk_s(m)x_s} e^{jk_r(n)x_r}, \qquad (7.77)$$

u and U denote the known data and unknown coefficients of the DFT, respectively.

Therefore, the interpolation problem can be posed as finding, from the incomplete data, the 2D-DFT (U) by solving

$$u = AU + n \qquad (7.78)$$

where n denotes the noise in the data. A unique solution may be obtained by minimizing the following expression

$$J = \| AU - u \|_2^2 + \epsilon \| U \|_2^2 \tag{7.79}$$

and the solution can be shown to take the form:

$$\hat{U} = (A^T A + \epsilon I)^{-1} A^T u, \tag{7.80}$$

where T denotes the transpose of a matrix.

Next, we derive a similar result but using a weighted DFT-domain norm introduced in the previous section. In this case the function to be minimized is

$$J = \| AU - u \|_2^2 + \epsilon \| U \|_P^2 \tag{7.81}$$

The solution takes the form:

$$\hat{U} = (A^T A + DI)^{-1} A^T u, \tag{7.82}$$

where D is a diagonal matrix with diagonal elements corresponding to $\frac{\epsilon}{|P(k_s,k_r)|^2}$ and $|P(k_s, k_r)|^2$ is a vector that contains the amplitude spectrum of U in lexicographic form . Ideally, one should know the amplitude spectrum of the data. Unfortunately, U is the unknown of our problem. The latter can be overcome by defining an initial D in terms of the DFT of the irregularly sampled data $A^T u$ and smoothing the result to attenuate the artifacts introduced by the irregularity of u (Ning and Nikias, 1990).

The scheme can be summarized as follows:

- Start with an initial $\hat{U}$.
- Compute $D = S(\hat{U}^* \hat{U})$, where S is a smoothing filter.
- Solve $\hat{U} = (A^T A + DI)^{-1} A^T u$ using Conjugate Gradients.
- Iterate until convergence.

An example of reconstruction is demonstrated on 15 synthetic shot gathers. Figure 7.15 shows six of the shot gathers with the shots #3 and #7 removed. The reconstruction is performed using the minimum weighted norm method with adaptive weights. Figure 7.16 shows the reconstructed shot gathers (only six shots are shown). The missing shots have been completely reconstructed.

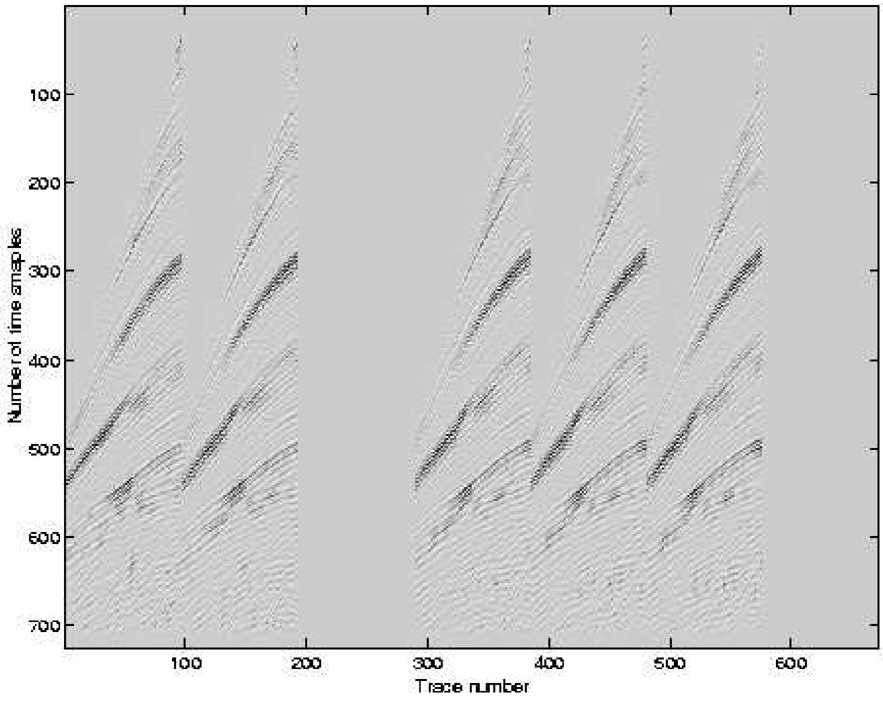


Figure 7.15: Six shot gathers with shots #3 and #7 removed.

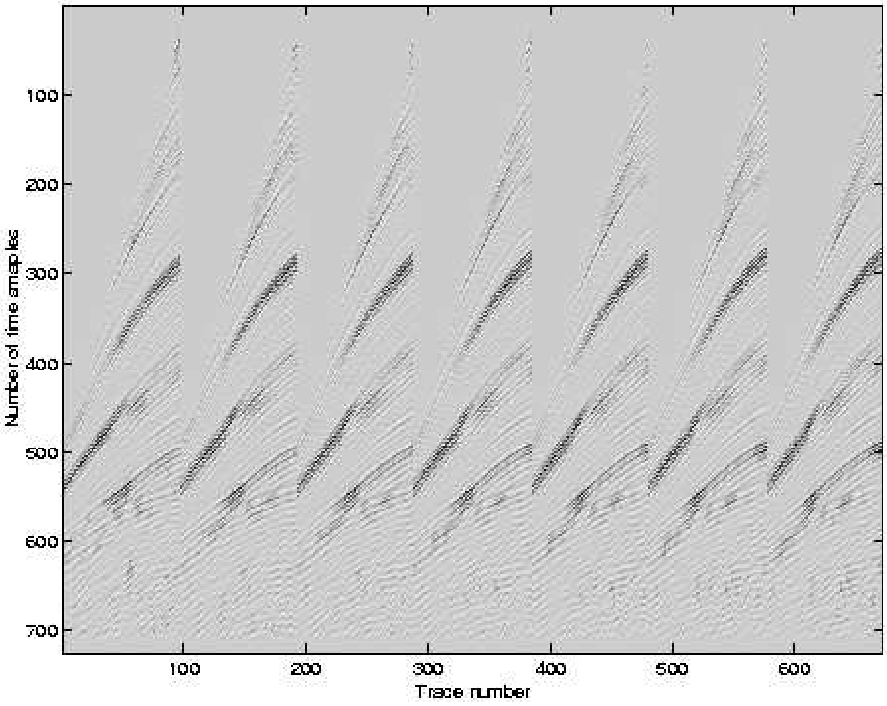


Figure 7.16: Reconstructed shot gathers using the minimum weighted norm algorithm.

7.9 References

Radon

Beylkin, G., 1987, Discrete radon transform: IEEE Trans. Acoust., Speech, Signal Processing., **ASSP-35**, 162-172.

Cary, P., W., 1998, The simplest discrete Radon transform, 68th Annual Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 1999-20 02.

Chapman, C. H., 1981, Generalized Radon transforms and slant stacks: Geophys. J. Roy. Astr. Soc., **54**, 481-518.

Deans, S. R., 1983, The Radon transform and some of its applications: J. Wil ey and Sons, Inc.

Durrani, T. S., and Bisset, D., 1984, The Radon transform and its properties : Geophysics, **49**, 1180-1187.

Foster, J. D., and Mosher, C. C., 1992, Suppression of multiples reflection using the Radon transform: Geophysics, **57**, 386-395.

Hampson, D., 1986, Inverse velocity stacking for multiple elimination: J. Can. Soc. Expl. Geophys., **22**, 44-55.

Hunt, L., Cary, P., and Upham, W., 1996, An improved Radon Transform for short period multiple attenuation: CSEG 23rd Annual Mtg., Expanded Abstracts, 58-59.

Kostov, C., 1990, Toeplitz structure in slant-stack inversion: 60th Annual Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 1618-1621.

Phinney, R. A., Chowdhury, K. R., and Frazer, L. N., 1981 Transformation and analysis of record sections: J. Geophys. Res., **86**, 359-377.

Pratt, W. K., 1991, Digital image processing: John Wiley and Sons, Inc

Sacchi, M.. D. and Ulrych, T., J., 1995, High-resolution velocity gathers and offset space reconstruction: Geophysics, **60**, 4, 1169-1177.

Thorson, J. R., and Claerbout, J. F., 1985, Velocity-stack and slant stack stochastic inversion: Geophysics, **50**, 2727-2741.

Yilmaz, Ö., 1989, Velocity-stack processing: Geophys. Prosp., **37**, 357-382.

Yilmaz, O., and Taner, M. T., 1994, Discrete plane-wave decomposition by least-mean-square-error method: Geophysics, **59**, 973-982.

Zhou, B., and Greenhalgh, S. A., 1994, Linear and parabolic $\tau - p$ revisited: Geophysics, **59**, 1133-1149.

FT Interpolation

Duijndam, A.J.W., Schonewille, M., and Hindriks, K., 1999, Reconstruction of seismic signals, irregularly sampled along on spatial coordinate: Geophysics, 64, 524-538.

Hindriks, K. O. H., Duijndam, A. J. W. and Schonewille, M. A., 1997, Reconstruction of two-dimensional irregularly sampled wavefields: Annual Meeting Abstracts, Society Of Exploration Geophysicists, 1163-1166.

Ning, T. and Nikias, C.L., 1990, Power Spectrum Estimation with Randomly Spaced Correlation Samples: IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 38, no. 6, 991-997.

Sacchi, M.D. and Ulrych, T.J., 1998, Interpolation and extrapolation using a high-resolution discrete Fourier transform: IEEE Trans. Signal Processing, vol. 46, no. 1, 31-38.