Contact:

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

MSacchi@ualberta.ca
www-geo.phys.ualberta.ca/~sacchi/
Contents

1 Fourier Analysis 1

1.1 Introduction .................................................. 1

1.1.1 Orthogonal Functions ........................................ 1

1.1.2 Fourier Series ............................................... 3

1.2 The Fourier Transform .......................................... 5

1.2.1 Properties of the FT ......................................... 6

1.2.2 The FT of some signals ...................................... 8

1.2.3 Truncation in time ........................................... 13

1.3 Symmetries ..................................................... 14

1.4 Living in a discrete World ...................................... 17

1.5 References ..................................................... 19

2 Z-transform and Convolution 23

2.1 Linear Systems ................................................ 23

2.1.1 Discrete convolution ......................................... 28

2.1.2 An algorithm to compute the convolution sum .......... 29

2.2 The Z transform ............................................... 31

2.2.1 Convolution and the Z-transform ......................... 32

2.2.2 Deconvolution .............................................. 33
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.3</td>
<td>Elementary Signals: Dipoles</td>
<td>34</td>
</tr>
<tr>
<td>2.3.1</td>
<td>Minimum phase dipoles</td>
<td>34</td>
</tr>
<tr>
<td>2.3.2</td>
<td>Maximum phase dipoles</td>
<td>38</td>
</tr>
<tr>
<td>2.3.3</td>
<td>Autocorrelation function of dipoles</td>
<td>42</td>
</tr>
<tr>
<td>2.3.4</td>
<td>Least squares inversion of a minimum phase dipole</td>
<td>46</td>
</tr>
<tr>
<td>2.3.5</td>
<td>Inversion of Minimum Phase sequences</td>
<td>51</td>
</tr>
<tr>
<td>2.4</td>
<td>MATLAB codes used in Chapter 2</td>
<td>55</td>
</tr>
<tr>
<td>2.4.1</td>
<td>Inversion of dipoles</td>
<td>55</td>
</tr>
<tr>
<td>2.4.2</td>
<td>Amplitude and phase</td>
<td>55</td>
</tr>
<tr>
<td>2.4.3</td>
<td>Least squares inversion of a dipole</td>
<td>56</td>
</tr>
<tr>
<td>2.4.4</td>
<td>Eigenvalues of the Toeplitz matrix</td>
<td>57</td>
</tr>
<tr>
<td>2.4.5</td>
<td>Least square inverse filters</td>
<td>57</td>
</tr>
<tr>
<td>2.5</td>
<td>The autocorrelation function</td>
<td>59</td>
</tr>
<tr>
<td>2.5.1</td>
<td>The Toeplitz matrix and the autocorrelation coefficients</td>
<td>60</td>
</tr>
<tr>
<td>2.6</td>
<td>Inversion of non-minimum phase wavelets: optimum lag Spiking filters</td>
<td>63</td>
</tr>
<tr>
<td>3</td>
<td>Discrete Fourier Transform</td>
<td>65</td>
</tr>
<tr>
<td>3.1</td>
<td>The Z transform and the DFT</td>
<td>65</td>
</tr>
<tr>
<td>3.1.1</td>
<td>Inverse DFT</td>
<td>67</td>
</tr>
<tr>
<td>3.1.2</td>
<td>Zero padding</td>
<td>69</td>
</tr>
<tr>
<td>3.1.3</td>
<td>The Fast Fourier Transform (FFT)</td>
<td>72</td>
</tr>
<tr>
<td>3.1.4</td>
<td>Working with the DFT/FFT</td>
<td>74</td>
</tr>
<tr>
<td>3.2</td>
<td>The 2D DFT</td>
<td>76</td>
</tr>
<tr>
<td>3.3</td>
<td>On the Design of Finite Impulse Response filters</td>
<td>78</td>
</tr>
<tr>
<td>3.3.1</td>
<td>Low Pass FIR filters</td>
<td>78</td>
</tr>
</tbody>
</table>
CONTENTS

3.3.2 High Pass filters ................................................................. 82
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:

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

(1.1)

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

\[
\delta_{i,j} = 0 \quad \text{if} \quad i \neq j
\]

\[
\delta_{i,j} = 1 \quad \text{if} \quad i = j.
\]
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:

\[
 f(t) \approx \sum_{i=1}^{N} c_i \phi_i(t) \quad (1.2)
\]

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

\[
 MSE = \frac{1}{t_2 - t_1} \int_{t_1}^{t_2} \left( f(t) - \sum_{i=1}^{N} \phi_i(t) \right)^2 dt \quad (1.3)
\]

the last equation can be expanded as follows:

\[
 MSE = \frac{1}{t_2 - t_1} \int_{t_1}^{t_2} \left( f(t)^2 - \sum_{i=1}^{N} \phi_i(t)^2 \right) dt + \sum_{i=1}^{N} c_i^2 \int_{t_1}^{t_2} \phi_i(t)^2 dt - 2 \sum_{i=1}^{N} c_i \gamma_i(f(t)) dt \quad (1.4)
\]

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

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

where

\[
 \gamma_i = \int_{t_1}^{t_2} \phi_i(t)f(t) dt \quad . \quad (1.6)
\]

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

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

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

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

(1.7)

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

\[ c_i \sqrt{(k_i)} = \frac{\gamma_i}{\sqrt{k_i}} \]  

(1.8)

or, in other words, the coefficient \( c_i \) is given by

\[ c_i = \frac{\gamma_i}{k_i} = \frac{\int_{t_1}^{t_2} f(t) \phi_i(t) \, dt}{\int_{t_1}^{t_2} \phi(t)^2 \, dt}, \]  

(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:

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

(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”:

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

(1.11)

1.1.2 Fourier Series

Consider the orthogonal set given by
\[ e^{jn\omega_0 t}, \quad n = 0, \pm 1, \pm 2, \pm 3, \ldots \] (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

\[
Int = \int_{t_0}^{t_0 + \frac{2\pi}{\omega_0}} e^{jn\omega_0 t} e^{-jk\omega_0 t} dt
= \frac{1}{j\omega_0(n-k)} e^{j(n-k)}(e^{j2\pi(n-k)} - 1),
\] (1.13)

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

\[
Int = \begin{cases} 
0 & \text{if } n \neq k \\
1 & \text{if } n = k 
\end{cases}
\] (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:

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

where the coefficients of the expansion are given by

\[ F_n = \frac{2\pi}{\omega_0} \int_{t_0}^{t_0 + \frac{2\pi}{\omega_0}} f(t)e^{-jn\omega_0 t} dt \] (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:

\[
\omega_n = n\omega_0
\]

---

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

2we have already obtained this result for an arbitrary set \( \phi_i(t) \)
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 allows 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:

\[
f(t) = \sum_{n=\infty}^{\infty} F_n e^{j\omega_0 t} dt, \quad \omega_0 = \frac{2\pi}{T}, \quad (1.17)
\]

where the coefficients are given by

\[
F_n = \frac{1}{T} \int_{-T/2}^{T/2} f(t) e^{-j\omega_0 t} dt \quad (1.18)
\]

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

\[
f(t) = \sum_{n=\infty}^{\infty} \frac{1}{T} \int_{-T/2}^{T/2} f(t) e^{-j\omega_0 t} dt e^{j\omega_0 t} \quad (1.19)
\]

Now suppose that we make \( T \to \infty \), 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

\[\text{3We want to extend out periodic signal into a non-periodic one}\]
continuous one \( \omega \), and finally, since now we have a summation on a continuous variable \( \omega \) we will convert the summation into an integral

\[
f(t) = \int_{-\infty}^{\infty} \frac{d\omega}{2\pi} \left( \int_{-\infty}^{\infty} f(t)e^{-j\omega t} dt \right) e^{j\omega t} d\omega.
\] (1.20)

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

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

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

\[
f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega)e^{j\omega t} d\omega.
\] (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:

\[
F(\omega) = |F(\omega)|e^{j\theta(\omega)}
\] (1.23)

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.

\(^4\)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.
1.2. THE FOURIER TRANSFORM

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

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

**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 $\leftrightarrow$ frequency multiplication.*

---

5This 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.
CHAPTER 1. FOURIER ANALYSIS

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

\[ f_1(t) \cdot f_2(t) \leftrightarrow \frac{1}{2\pi} \int_{-\infty}^{\infty} F_1(v) F_2(\omega - v) \, dv \]

or in a few words:\textit{time multiplication} \Rightarrow \textit{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

A Boxcar

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

\(^6\)We will use this property to estimate the FT of signal that has been recorded in a finite temporal window. See (1.2.3)
1.2. THE FOURIER TRANSFORM

\[ f(t) = \begin{cases} 
1 & |t| < T/2 \\
0 & \text{otherwise}
\end{cases} \quad (1.24) \]

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

\[
F(\omega) = \int_{-T/2}^{T/2} 1.e^{-j\omega t} dt \\
= -\frac{j}{\omega}(e^{-j\omega T/2} - e^{j\omega T/2}) \\
= \frac{T \text{sinc}(\omega T/2)}{(1.25)}
\]

where in last equation \( \text{sinc}(x) = \sin(x)/x \). The FT of the boxcar function is a \( \text{sinc} \) 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 (1.2.2) 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 $T = 10$ secs
1.2. THE FOURIER TRANSFORM

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

\[ f(t) = \delta(t) \]

the \( \delta \) 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 \( \delta \) 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
1.2. THE FOURIER TRANSFORM

\[ F(t) \leftrightarrow 2\pi f(-\omega) \]

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 \( \omega_0 \) is a delta at the corresponding frequency \( \omega = \omega_0 \).

1.2.3 Truncation in time

Given \( f(t) \in (-\infty, \infty) \), 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 \( B_T(\omega) = T \text{sinc}(\omega T/2) \). 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 \( F(\omega) \).

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

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:

\[
F_T(\omega) = \frac{1}{2\pi} T \int_{-\infty}^{\infty} 2\pi \delta(\omega - \omega_0) \text{sinc}((\omega - u)T/2) du .
\]

\[ F_T(\omega) = T \text{sinc}((\omega - \omega_0)T/2) . \]

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

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

The FT of such a signal (if measured in an infinity interval) is given by two delta functions at frequencies \( \omega_1 \) and \( \omega_2 \). 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 \( \omega_1 = 0.5 \text{ rad/sec} \) and \( \omega_2 = 1. \text{ rad/sec} \).

### 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,

\[
F(\omega) = \int f(t)e^{-j\omega t} dt
\]

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

\[
F(\omega) = R(\omega) + iG(\omega)
\]

where
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 ($|F(\omega)|$).
\[ R(\omega) = \int f(t) \cos(\omega t) dt \] (1.33)

and

\[ G(\omega) = -\int f(t) \sin(\omega t) dt \] (1.34)

Since the \( \cos \) is an even function and the \( \sin \) an odd function:

\[ R(\omega) = R(-\omega) \] (1.35)

\[ G(\omega) = -G(-\omega) \] (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:

\[ F(\omega) = R(\omega) + iG(\omega) \] (1.37)

\[ F(-\omega) = R(-\omega) + iG(-\omega) \] (1.38)

by combining last equation with equation (1.36) we obtain

\[ F(-\omega) = R(\omega) - iG(\omega). \] (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 easy to prove that the amplitude is an even function:

\[ |F(\omega)| = |F(-\omega)| \quad (1.40) \]

and that the phase is an odd function

\[ \theta(\omega) = -\theta(-\omega). \quad (1.41) \]

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 \( \Delta t \) seconds

\[ f_s(t) = f(t) \sum_{k=-\infty}^{\infty} \delta(t - k\Delta t). \quad (1.42) \]

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

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

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
One can observe that the FT of the sampled signal is a periodic function with period $\omega_0$.

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 $[-\omega_{\text{max}}, \omega_{\text{max}}]$ are zero. If the following condition is satisfied

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

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

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

which reduces to

$$\Delta T \leq \frac{1}{2f_{\text{max}}}.$$
1.5. REFERENCES

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 observe 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 is 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 undersampled 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 $\omega_{\text{max}} = 10$ and $\omega_0 = 20$. The Fourier transform of the continuous signal is perfectly represented in the interval $[-w_{\text{max}}, w_{\text{max}}]$.

Figure 1.8: The Fourier transform the continuous signal after being discretized, in this case $\omega_{\text{max}} = 10$ 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 $y(t)$

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

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

P1:

$$ax(t) \rightarrow ay(t).$$

P2: If

$$x_1(t) \rightarrow y_1(t)$$

and
\[ x_2(t) \rightarrow y_2(t) \]

then

\[ x_1(t) + x_2(t) \rightarrow y_1(t) + y_2(t) \]

\[ x(t - T) \rightarrow y(t - T) \]

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

\[ \alpha x_1(t) + \beta x_2(t) \rightarrow \alpha y_1(t) + \beta y_2(t) . \]

**P3:** Properties \( \textbf{P1} \) and \( \textbf{P2} \) 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 represent our linear system as follows:

\[ \mathcal{H}[x(t)] = y(t) . \quad (2.1) \]

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

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

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:

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

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:

\[ h(t, \tau) = h(t - \tau) . \quad (2.4) \]
2.1. LINEAR SYSTEMS

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 \( \rightarrow \) multiplication in the frequency domain”, we can rewrite the convolution integral as follows:

\[
Y(\omega) = H(\omega) \cdot X(\omega).
\]

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 \( \delta \) 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.
2.1. LINEAR SYSTEMS

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 \( \delta_n \) 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:

\[ y_k = \sum_{n=\infty}^{\infty} h_n x_{k-n} \]  

(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., \( x_n, n = 0 : NX - 1 \),

\[ y_k = \sum_{n=p1}^{p2} h_{k-n} x_n, \]  

(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,

\[ \begin{align*}
y_0 &= x_0 h_0 \\
y_1 &= x_1 h_0 + x_0 h_1 \\
y_2 &= x_2 h_0 + x_1 h_1 + x_0 h_2 \\
y_3 &= x_3 h_0 + x_2 h_1 + x_1 h_2 \\
y_4 &= x_4 h_0 + x_3 h_1 + x_2 h_2 \\
y_5 &= x_4 h_1 + x_3 h_2 \\
y_6 &= x_4 h_2 \\
\end{align*} \]  

(2.11)

The output time series is given by \( y = [y_0, y_1, y_2, \ldots, y_7] \). \(^1\)

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 \( NY = NX + NH - 1 \).
2.1. LINEAR SYSTEMS

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

(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)

\[ y_{j+n} = \sum_{j=0}^{NH-1} h_j x_n, \quad j = 0 : NX - 1 \ldots \] (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:

\[ x(1) \ x(2) \ x(3) \ x(4) \ldots \ x(NX) \]

Where \( x_0 = x(1) \ldots \ x_{NX-1} = x(NX) \). 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 \texttt{conv} to perform the same task \(^2\).

\(^2\) Use \texttt{" help conv "}.

Convolution in MATLAB

```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:

```fortran
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

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:

\[ x_0, x_1, x_2, x_3 \] (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 \( \Delta t \) seconds.

The \( z \) transform of \( x_k, k = 1, 2, \ldots \) is defined as

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

For a finite length time series \( x_k, k = 0, \ldots, N-1 \) we write

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

A simple example is a time series composed of 4 samples

\[ x = 4, 12, -1, -3, \] (2.17)

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

\[ X(z) = 4 + 12z - 1z^2 + 3z^3. \] (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.
In this case the z transform is given by

\[ X(z) = -z^{-3} + 3z^{-2} + 4z^{-1} + 3 + 5z + 6z^2 - 10z^3. \] (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:

\[ X(z).H(z) = x_0 h_0 + \]

\[ (x_1 h_0 + x_0 h_1) z + \]

\[ (x_2 h_0 + x_1 h_1 + x_0 h_2) z^2 + \]

\[ (x_3 h_0 + x_2 h_1 + x_1 h_2) z^3 + \]

\[ (x_4 h_0 + x_3 h_1 + x_2 h_2) z^4 + \]

\[ (x_4 h_1 + x_3 h_2) z^5 + \]

\[ (x_5 h_2) z^6 \] (2.22)

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

\[ Y(z) = X(z).H(z). \] (2.23)
2.2. THE Z TRANSFORM

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:

\[ y_k = h_k \ast x_k \quad \rightarrow \quad Y(z) = H(z) \cdot X(z) \]

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:

\[ X(z) = \frac{Y(z)}{H(z)}. \quad (2.24) \]

The inverse operator is defined as:

\[ F(z) = \frac{1}{H(z)}, \quad (2.25) \]

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

\[ X(z) = F(z) \cdot Y(z) \quad (2.26) \]

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:

\[ x_k = f_k \ast y_k. \quad (2.27) \]

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

$$s_k = w_k * q_k.$$  \hspace{1cm} (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

$$1 + az$$  \hspace{1cm} (2.29)
2.3. **ELEMENTARY SIGNALS: DIPOLES**

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

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

We have already seen that 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:

\[
F(z) = 1 - az + (az)^2 - (az)^3 + (az)^4 \ldots \tag{2.35}
\]

⁴A geometric series.
Since $|a| < 1$ the above series is a convergent series. $F(z)$ is the $z$ transform of the time series $f_k, k = 0, \ldots, \infty$:

$$1, -a, a^2, -a^3, a^4, \ldots$$  \hspace{1cm} (2.36)

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

$$(1, a) * (1, -a, a^2, -a^3, a^4, \ldots) = 1, 0, 0, 0, 0, \ldots$$  \hspace{1cm} (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 $1, -a, a^2, -a^3, a^4, \ldots$ 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 $a = 0.9$:

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

In the second case, we have:

$$F(z) = \frac{1}{1 + 0.1z} = 1 - 0.1z + 0.01z^2 - 0.001z^3 + 0.0001z^4 \ldots.$$  \hspace{1cm} (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:

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

$$(1, 0.1) * (1, -0.1, 0.01, -0.001, 0.0001) = (1, 0.0, 0.0, 0.0, 0.0).$$  \hspace{1cm} (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 filter 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, \( \xi \), 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:

\[
X(z)F(z) = 1 \tag{2.45}
\]

\[
(1 + a_1 z)(1 + a_2 z)(1 + a_3 z) \ldots F(z) = 1.
\]

From the above equation where we can write
Chapter 2. Z-Transform and Convolution

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:

\[
F(z) = \frac{1}{1 + a_1 z^{-1}} \frac{1}{1 + a_2 z^{-1}} \frac{1}{1 + a_3 z^{-1}} \ldots
\]

(2.46)

\[
= \left[1 - a_1 z + (a_1 z)^2 - (a_1 z)^3 \ldots\right]\left[1 - a_2 z + (a_2 z)^2 - (a_2 z)^3 \ldots\right] \ldots
\]

(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. ELEMENTARY SIGNALS: DIPOLES

Figure 2.5: Inversion of minimum phase dipoles

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.
\[ D(z) = 1 + bz \Rightarrow D(\xi) = 1 + b\xi = 0 \Rightarrow \xi = -1/b. \tag{2.48} \]

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

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

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

\[ \frac{1}{1 + bz} = 1 - bz + (bz)^2 - (bz)^3 \ldots \tag{2.50} \]

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 \textit{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

\[ F(z) = (bz)^{-1}(1 - (bz)^{-1} + (bz)^{-2} - (bz)^{-3} \ldots). \tag{2.52} \]

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

\[ f = \ldots, -b^{-3}, b^{-2}, -b^{-1}, 0 \tag{2.53} \]

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):
2.3. **ELEMENTARY SIGNALS: DIPOLES**

Dipole, $d = (1,2)$  \hspace{1cm}  Truncated inverse filter (non-casual), $f$  \hspace{1cm}  Output, $d \otimes f$

![Diagram showing dipole, inverse filter, and output](image)

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

$$f = (-0.0156, 0.0312, -0.0625, 0.125, -0.25, 0.5, 0)$$  \hspace{1cm}  (2.54)

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

$$d \ast f = (-0.0156, 0.0312, -0.0625, 0.125, -0.25, 0.5, 0) \ast (1, 2)$$ \hspace{1cm}  (2.55)

$$= (-0.0156, 0, 0, 0, 0, 1, 0)$$
2.3.3 Autocorrelation function of dipoles

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

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

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

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

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

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

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

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

We have arrived to a very important conclusion

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

\(^5\text{Note that for real dipoles, } a^* = a\)
or in other words, two different sequences can have the same autocorrelation function. The autocorrelation sequence in both cases is the following time series

\[ a^*, (1 + a^2), a \]

or

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

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

\[
R(z) = r_1z^{-1} + r_0 + r_1z^{-1} = a^*z^{-1} + (1 + a^2) + az^{-1}
\]  

(2.64)

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

\[
r_k = \sum_n x_{n+k}^* x_n, \]

(2.65)

\[
R(z) = X(z)X^*(z^{-1}),
\]

(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 autocorrelation 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.

\[
R(\omega) = R(z)|_{z = e^{-i\omega}}
\]

(2.67)
or

\[ R(\omega) = [X(z) \cdot X^*(z^{-1})]_{z=e^{-i\omega}} \]  

(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:

\[ D_{\text{min}}(z) = 1 + a z \Rightarrow z = e^{-i\omega} \Rightarrow D_{\text{min}}(\omega) = 1 + ae^{-i\omega} . \]  

(2.69)

Whereas for the maximum phase dipole we have

\[ D_{\text{max}}(z) = a + z \Rightarrow z = e^{-i\omega} \Rightarrow D_{\text{max}}(\omega) = a + 1e^{-i\omega} \]  

(2.70)

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

\[ R_{D_{\text{min}}}(\omega) = \sqrt{1 + 2a \cos(\omega) + a^2} \]  

(2.71)

\[ \theta_{\text{min}}(\omega) = \arctan \left( \frac{a \sin(\omega)}{1 + a \cos(\omega)} \right) \]  

(2.72)

For the maximum phase signal we have

\[ R_{D_{\text{max}}}(\omega) = \sqrt{1 + 2a \cos(\omega) + a^2} \]  

(2.73)

\[ \theta_{\text{max}}(\omega) = \arctan \left( \frac{\sin(\omega)}{a + \cos(\omega)} \right) \]  

(2.74)
2.3. ELEMENTARY SIGNALS: DIPOLES

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

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 \(N = 3\)),

\[
(1, a) \ast (f_0, f_1, f_2) \approx (1, 0, 0, 0). \tag{2.77}
\]

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

\[ C f \approx b, \quad (2.79) \]

where \( C \) 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 \( f \) and the desired output by \( d \). It is clear that the solution vector is the one that minimized the mean squared error

\[ \epsilon = ||C f - b||^2. \quad (2.80) \]

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

\[ C^T C f = C^T b. \quad (2.81) \]

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

\[ f = R^{-1} C^T b, \quad (2.82) \]

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

\[ R = \begin{pmatrix} 1 + a^2 & a & 0 \\ a & 1 + a^2 & a \\ 0 & a & 1 + a^2 \end{pmatrix}. \quad (2.83) \]

One can see that each row of the matrix \( R \) is composed by elements of the autocorrelation sequence given by equation (2.63).
CHAPTER 2. Z-TRANSFORM AND CONVOLUTION

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 \( \mathbf{R} \).

It is interesting to note that the condition number of the Toeplitz matrix\(^7\) 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

\[ \mathbf{R} = \begin{pmatrix} r_0 & r_1 & 0 \\ r_1 & r_0 & r_1 \\ 0 & r_1 & r_0 \end{pmatrix}. \quad (2.84) \]

---

\(^7\)The condition number of the matrix \( \mathbf{R} \) is the ratio \( \lambda_{\max}/\lambda_{\min} \) where \( \lambda_{\max} \) and \( \lambda_{\min} \) are the largest and smallest eigenvalues of \( \mathbf{R} \), respectively. A large condition number indicates that the problem is ill-conditioned (numerical problems will arise at the time of inverting the matrix.)
Figure 2.11: Condition number of the Toeplitz matrix versus $NF$ (filter length). The dipole is the minimum phase sequence $(1, 0.5)$.

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.
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. ELEMENTARY SIGNALS: DIPOLES

2.3.5 Inversion of Minimum Phase sequences

So far we have discussed 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 is a signal that can be decomposed through 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 $w_k, k = 1, \ldots, NW$, the filter $f_k, k = 1, \ldots, NF$ needs to be designed to satisfy the following equation:

$$ (w_0, w_1, \ldots w_{NW-1}) \ast (f_0, f_1, \ldots, f_{NF-1}) \approx (1, 0, \ldots, 0) \quad (2.85) $$

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

Again, this system is written in matrix form as $C f \approx d$. We will compute the inverse filter by minimizing the error function (mean squared error) $\epsilon$:

$$ \epsilon = ||e||^2 = ||C f - b||^2; \quad (2.87) $$

The Euclidean norm of the error vector $e = C f - b$ can be written down as

\(^8\)In other words, all the zeros of the $z$ transform of the wavelet lie outside the unit circle.
\[ \epsilon = e^T e = (C f - b)^T (C f - b) . \] (2.88)

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

\[ \frac{d\epsilon}{df} = 0 , \] (2.89)

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

\[ C^T C f = C^T b . \] (2.90)

It is clear that the inverse filter is solved by inverting the Toeplitz form \( R = C^T C \), 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 unstabilities at the time of inversion. This shortcoming can be avoided by using a regularization strategy. Instead of minimizing the misfit function \( \epsilon \) we will minimize the following penalized objective function

\[ J = \epsilon + \mu ||f||^2 , \] (2.91)

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

\[ \frac{dJ}{df} = 0 , \] (2.92)

leads to the following solution

\[ f = (R + \mu I)^{-1} C^T d . \] (2.93)
2.3. ELEMENTARY SIGNALS: DIPOLES

It is clear that the parameter $\mu$ 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 $\epsilon$, 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 $\mu$ is large the energy of the filter will be small and the misfit function $\epsilon$ will be large. In this case we have a matching filter of the form

$$f = \mu^{-1} C^T d.$$  \hspace{1cm} (2.94)

Last equation was obtained by doing the following replacement $(R + \mu I) \approx \mu I$, which is valid only when $\mu$ 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 $\mu$ is small the output sequence is a spike, when we increase $\mu$ 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, $\epsilon$, versus the norm of the filter $||f||^2$ for various values of the tradeoff parameter $\mu$. 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 ($\mu$).

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. MATLAB CODES USED IN CHAPTER 2

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
CHAPTER 2. Z-TRANSFORM AND CONVOLUTION

\[
\begin{align*}
C &= \text{convmtx}(w,NF); & \text{\textcopyright Convolution matrix} \\
R &= C' \ast C; & \text{\textcopyright Toeplitz Matrix} \\
rhs &= C' \ast b; & \text{\textcopyright Right hand side vector} \\
I &= \text{eye}(R) \ast \mu; \\
f &= \text{inv}(R+I) \ast \text{rhs}; & \text{\textcopyright Filter} \\
o &= \text{conv}(f,w); & \text{\textcopyright Actual output} \\
\text{return} & 
\end{align*}
\]
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)

\[ R(z) = X(z) X^*(z^{-1}) \quad (2.95) \]

\[ R(z) = x_0 x_2^* z^{-2} + (x_0 x_1^* + x_1 x_2^*) z^{-1} + (x_0 x_0^* + x_1 x_1^* + x_2 x_2^*) + (x_1 x_0^* + x_2 x_1^*) z + x_2 x_0^* z^{-2}. \quad (2.96) \]

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

\[ R(z) = \sum_{k=-\infty}^{\infty} r_k z^k \quad (2.97) \]

where

\[
\begin{align*}
  r_{-2} &= x_0 x_2^* \\
  r_{-1} &= x_0 x_1^* + x_1 x_2^* \\
  r_0 &= x_0 x_0^* + x_1 x_1^* + x_2 x_2^* \\
  r_1 &= x_1 x_0^* + x_2 x_1^* \\
  r_2 &= x_2 x_2^* \\
  r_k &= 0 \text{ otherwise.}
\end{align*}
\]

(2.98)

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

\[ x_0, x_1, x_2, x_3, \ldots, x_{NX-1} \]

the autocorrelation coefficient can be computed using the following formulas:
\[ r_{-k} = \sum_{i=0}^{NX-1} x_i x_{i+k}^* \quad k = 1, 2, 3, \ldots, NX - 1 \]
\[ r_0 = \sum_{i=0}^{NX-1} x_i x_i^* \]
\[ r_k = \sum_{i=0}^{NX-1} x_{i+k} x_i^* \quad k = 1, 2, 3, \ldots, NX - 1 \quad \text{[Note]}^9 \]

Properties of the autocorrelation sequence:

1. Hermitian Symmetry: \[ r_k = r_{-k}^* \quad k = \pm 1, \pm 2, \ldots \]

2. \[ r_0 > |r_k| \quad k = \pm 1, \pm 2, \ldots \]

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 \( x_0, x_1, \ldots, x_{NX-1} \) 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

\[ C^T C f = C b \quad (2.100) \]

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
2.5. **THE AUTOCORRELATION FUNCTION**

\[
C = \begin{pmatrix}
  w_0 & 0 & 0 & 0 \\
  w_1 & w_0 & 0 & 0 \\
  w_2 & w_1 & w_0 & 0 \\
  w_3 & w_2 & w_1 & w_0 \\
  w_4 & w_3 & w_2 & w_1 \\
  w_5 & w_4 & w_3 & w_2 \\
  w_6 & w_5 & w_4 & w_3 \\
  0 & w_6 & w_5 & w_4 \\
  0 & 0 & w_6 & w_5 \\
  0 & 0 & 0 & w_6 \\
\end{pmatrix}
\]  

(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 \( R = C^T C \) is given by

\[
R = \begin{pmatrix}
  r_0 & r_1 & r_2 & r_3 \\
  r_1 & r_0 & r_1 & r_2 \\
  r_2 & r_1 & r_0 & r_1 \\
  r_3 & r_2 & r_1 & r_0 \\
\end{pmatrix}
\]  

(2.102)

where the elements of \( R \) are given by:

\[
r_k = \sum_{i=0}^{NW-1-k} w_{i+k} w_i \quad k = 0, 1, 2, 3, \ldots, NF - 1
\]

(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:

\[
r_0 = \sum_{i=0}^{NW-1} w_i^2
\]

(2.104)

It is important to stress that at the time of computing the Toeplitz matrix we do not need to compute the product \( C^T C \); 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: optimum 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 least-squares approach is that the filter is also minimum phase.

If the wavelet is not minimum phase, the actual output (the convolution of the filter with the wavelet) does not resemble the desire output. To alleviate this problem, an optimum lag Wiener filter is defined. This is an inverse filter where the desired output is the following sequence:

\[(0, 0, 0, 0, \ldots, 1, 0, 0, 0, \ldots)\]  \hspace{1cm} (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

\[ P = 1 - E \]  \hspace{1cm} (2.106)

\[ E = \frac{1}{r_0}||C\hat{f} - b||^2 \]  \hspace{1cm} (2.107)

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

\[ 0 \leq E \leq 1 \]

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:

\[ X(z) = \sum_{n=0}^{N-1} x_n z^n. \] (3.1)

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

\[ z = e^{-i\omega}. \] (3.2)

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

\[ X(\omega) = \sum_{n=0}^{N-1} x_n e^{-i\omega n}. \] (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 \( \omega \)). The phase \( \omega \) is also the frequency variable in units of radians. It is easy to make an analogy with the Fourier transform

\[
F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-i\omega t} dt
\]  

(3.4)

In the last equation the frequency is given in radians/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, \( \omega \) has units of radians.

The DFT maps a discrete signal into the frequency domain. So far, the frequency \( \omega \) is a continuous variable, but let us assume that one wants to discretized \( \omega \) in the same way we have discretized the temporal variable \( t \). The limits of \( \omega \) are given by \([0, 2\pi)\), remember that \( \omega \) is an angular frequency.

If the time series is a \( N \) points time series, we can discretized the frequency axis as follows:

\[
\omega_k = k \frac{2\pi}{N}, \quad k = 0, 1, \ldots, N - 1
\]  

(3.5)

Now we can define the DFT as follows:

\[
X(\omega_k) = \sum_{n=0}^{N-1} x(n) e^{-i\omega_k n}, \quad k = 0, 1, \ldots, N - 1.
\]  

(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

\[
\begin{pmatrix}
  X_0 \\
  X_1 \\
  X_2 \\
  \vdots \\
  X_{N-1}
\end{pmatrix} =
\begin{pmatrix}
  1 & e^{-i2\pi/N} & e^{-i2\pi2/N} & \cdots & e^{-i2\pi(N-1)/N} \\
  1 & e^{-i2\pi2/N} & e^{-i2\pi4/N} & \cdots & e^{-i2\pi2(N-1)/N} \\
  1 & e^{-i2\pi4/N} & e^{-i2\pi4/N} & \cdots & e^{-i2\pi2(N-1)/N} \\
  \vdots & \vdots & \vdots & \ddots & \vdots \\
  1 & e^{-i2\pi(N-1)/N} & e^{-i2\pi2(N-1)/N} & \cdots & e^{-i2\pi(N-1)(N-1)/N}
\end{pmatrix}
\begin{pmatrix}
  x_0 \\
  x_1 \\
  x_2 \\
  \vdots \\
  x_{N-1}
\end{pmatrix}
\]

(3.7)

The last equation can be written in compact form as follows:
3.1. THE Z TRANSFORM AND THE DFT

\[ X = F \cdot x \]  

(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 invertibility of the DFT. We need a transform to come back from the frequency domain to the time domain, in other words we need \( F^{-1} \).

3.1.1 Inverse DFT

We propose the following inverse

\[ x_n = \sum_{l=0}^{N-1} a_l e^{i2\pi ln/N} . \]  

(3.9)

where the coefficients \( a \) 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 \( a_k \) are our unknowns. In order to find our unknowns we proceed as follows, first we replace the last equation into equation (3.6),

\[ X_k = \sum_{n=0}^{N-1} \sum_{l=0}^{N-1} a_l e^{i2\pi n(l-k)/N} . \]  

(3.10)

The last equation can be rewritten as

\[ X_k = \sum_{l=0}^{N-1} a_l \sum_{n=0}^{N-1} e^{i2\pi n(l-k)/N} = \sum_{l=0}^{N-1} a_l s_{l-k} , \]  

(3.11)

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

\[ s_{l-k} = \sum_{n=0}^{N-1} e^{i2\pi n(l-k)/N} . \]  

(3.12)
At this point we realize that the last equation is a geometric series\(^1\) with a sum given by

\[
\sum_{n=0}^{N-1} u^n = \begin{cases} 
N & \text{if } u = 1 \\
\frac{u^N}{1-u} & \text{if } u \neq 1
\end{cases} \quad (3.13)
\]

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

\[
s_{l-k} = \begin{cases} 
N & \text{if } l = k \\
0 & \text{if } l \neq k
\end{cases}, \quad (3.14)
\]

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

\[
X_k = N\alpha_k, \quad k = 0, \ldots, N - 1. \quad (3.15)
\]

our inversion formula becomes:

\[
x_n = \frac{1}{N} \sum_{l=0}^{N-1} X_l e^{i2\pi ln/N}. \quad (3.16)
\]

This equation can also be written as follows:

\[
x = \frac{1}{N} F^H X. \quad (3.17)
\]

The matrix \( F^H \) is the Hermitian transpose of the matrix \( F \). It is clear that the \( N \times N \) matrix \( F \) is an orthogonal matrix,

\[
F^H F = N I_N, \quad (3.18)
\]

---

\(^1\)We have used a geometric series to find the inverse of a dipole in Chapter 2
where $I_N$ is an $N \times N$ identity matrix. Finally we have a pair of transforms, the DFT and the IDFT (inverse DFT), given by

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

(3.19)

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

(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 $\omega_k$,

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

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,

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

The new time series is an $M$-points time series with a DFT given by
CHAPTER 3. DISCRETE FOURIER TRANSFORM

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

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

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 code was utilized to
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 $\Delta \omega$. 
generate Figures (3.1) and (3.2).

% 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 have a time series of length 2N:

\[ z_0, z_1, z_2, z_3, \ldots, z_{2N-1}. \]

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

\[ Z_k = \sum_{n=0}^{2N-1} z_n e^{-i2\pi nk/(2N)}, \quad k = 0 : 2N - 1, \quad (3.22) \]
we can rewrite the last equation in terms of two time series composed of even samples \( x_0, x_2, x_4 \ldots \) and odd samples \( y_1, y_3, y_5 \ldots \), respectively.

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

\[
Z_k = X_k + e^{-i2\pi k/(2N)} Y_K, \quad k = 0 : N - 1.
\] (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 \( z \),

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

In the last equation we apply the following substitution: \( j = k - N \)

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

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

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

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

(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

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

(3.29)

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

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

(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

<table>
<thead>
<tr>
<th>Sample k</th>
<th>X_k</th>
<th>N-k (N=8)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>19.0000</td>
<td>8</td>
</tr>
<tr>
<td>1</td>
<td>-4.1213 - 1.2929i</td>
<td>7</td>
</tr>
<tr>
<td>2</td>
<td>6.0000 - 3.0000i</td>
<td>6</td>
</tr>
<tr>
<td>3</td>
<td>0.1213 + 2.7071i</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>-7.0000</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>0.1213 - 2.7071i</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>6.0000 + 3.0000i</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>-4.1213 + 1.2929i</td>
<td>1</td>
</tr>
</tbody>
</table>

It is clear 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

<table>
<thead>
<tr>
<th>$k$</th>
<th>$\omega_k$</th>
<th>$X_k$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>19.0000</td>
</tr>
<tr>
<td>1</td>
<td>0.7854</td>
<td>-4.1213 - 1.2929i</td>
</tr>
<tr>
<td>2</td>
<td>1.5708</td>
<td>6.0000 - 3.0000i</td>
</tr>
<tr>
<td>3</td>
<td>2.3562</td>
<td>0.1213 + 2.7071i</td>
</tr>
<tr>
<td>4</td>
<td>3.1416</td>
<td>-7.0000</td>
</tr>
<tr>
<td>5</td>
<td>3.9270</td>
<td>0.1213 - 2.7071i</td>
</tr>
<tr>
<td>6</td>
<td>4.7124</td>
<td>6.0000 + 3.0000i</td>
</tr>
<tr>
<td>7</td>
<td>5.4978</td>
<td>-4.1213 + 1.2929i</td>
</tr>
</tbody>
</table>

Note that the central frequency is $\omega_4 = \pi$, the last frequency is almost $2\pi$ or $\omega_7 = 2\pi - \Delta \omega$. This is because we have discretized the unit circle in the interval $[0, 2\pi)$. It does not make much sense to talk about frequencies above $\pi$ 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.
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.

### 3.2 The 2D DFT

The 2D Fourier transform is defined as follows:

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

similarly, we can define the inversion formula

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

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)

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

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

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

\[
x_{k,l} = \frac{1}{NM} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} X_{m,n} e^{i2\pi km/M} e^{i2\pi ln/N}, \quad k = 0, \ldots, M, \quad l = 0, \ldots, N. \tag{3.34}
\]
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:

\[ B(\omega) = \begin{cases} 1 & -\omega_c \leq \omega \leq \omega_c \\ 0 & \text{otherwise} \end{cases} \] (3.35)

We will assume that the filter phase is zero. In the previous expression \( \omega_c \) 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

\[ Y(\omega) = X(\omega) \cdot F(\omega) \] (3.36)

In general, it is more convenient to design short filters in the time domain and applied them via convolution\(^2\)

\[ y(t) = x(t) \ast b(t) \] (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 \( b(t) \),

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

\(^2\)note that we are working with continuous signals.
3.3. **ON THE DESIGN OF FINITE IMPULSE RESPONSE FILTERS**

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

\[
    b(t) = \frac{\omega_c \sin(\omega_c t)}{\pi} = \frac{\omega_c}{\pi} \text{sinc}(\omega_c t), \quad \infty < t < \infty.
\]

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:

\[
b_n = \Delta t b(t)|_{t=n \Delta t}
\]

the factor \( \Delta t \) 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

\[
b_n = \Delta t \frac{\omega_c}{\pi} \text{sinc}(\omega_c n \Delta t), \quad n = \ldots, -3, -2, -1, 0, 1, 2, 3 \ldots.
\]

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

\[
b_n = \Delta t \frac{\omega_c}{\pi} \text{sinc}(\omega_c n \delta t), \quad n = -L \ldots, -3, -2, -1, 0, 1, 2, 3 \ldots L.
\]

In this case we have a filter of length \( 2L + 1 \). 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:

$$w_n = 0.54 - 0.45 \cos\left(2\pi \frac{n - 1}{(N - 1)}\right), \ n = 1 : N$$

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.
3.3. ON THE DESIGN OF FINITE IMPULSE RESPONSE FILTERS

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 is given by $B^L(\omega)$ we can construct a high pass filter with the same cut-off frequency using the following expression:

$$B^H(\omega) = 1 - B^L(\omega) \quad (3.43)$$

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

$$
\begin{align*}
    b^H_k &= -b^L_k \quad k \neq 0 \\
    b^H_0 &= 1 - b^L_0 \quad k = 0
\end{align*}
\quad (3.44)
$$