# Mathematics of Signal Processing

**Charles L. Byrne**

November 17, 2004

**To Eileen**

# Contents

# Chapter 1

# Introduction

In graduate school and for the first few years as an assistant professor I concentrated on pure mathematics, mainly topology and functional analysis. Around 1979 I was drawn, largely by accident, into signal processing, collaborating with friends at the Naval Research Laboratory who were working on SONAR. I quickly found out that the intersection of the mathematics I knew and that they knew was nearly empty. For the last twenty-five years I have been trying to remedy that situation. In writing this book I have tried to gather together in one place the mathematics I wish I had known in 1979 but did not, in the hope that it will be helpful to others undertaking a similar journey.

The situations of interest to us here can be summarized as follows: the data has been obtained through some form of sensing; physical models, often simplified, describe how the data we have obtained relates to the information we seek; there usually isn't enough data and what we have is corrupted by noise and other distortions. Although applications differ from one another in their details they often make use of a common core of mathematical ideas; for example, the Fourier transform and its variants play an important role in many areas of signal and image processing, as do the language and theory of matrix analysis, iterative optimization and approximation techniques and the basics of probability and statistics. This common core provides the subject matter for this text. Applications of the core material to tomographic medical imaging, optical imaging and acoustic signal processing are included.

The term *signal processing* is used here in a somewhat restrictive sense to describe the extraction of information from measured data. I believe strongly that to get information out we must put information in. How to do this is one of the main topics of the book.

This text is designed to provide the necessary mathematical background to understand and employ signal processing techniques in an applied en-

vironment. The emphasis is on a small number of fundamental problems and essential tools, as well as on applications. Certain topics that are commonly included in textbooks are touched on only briefly or in exercises or not mentioned at all. Other topics not usually considered to be part of signal processing, but which are becoming increasingly important, such as iterative optimization methods, are included. The book, then, is a rather personal view of the subject and reflects the author's interests.

The term *signal* is not meant to imply a restriction to functions of a single variable; indeed most of what we discuss in this text applies equally to functions of one and several variables and therefore to image processing. However, there are special problems that arise in image processing, such as edge detection, and special techniques to deal with such problems; we shall not consider such techniques in this text. Topics discussed include the following: Fourier series and transforms in one and several variables; applications to acoustic and EM propagation models, transmission and emission tomography and image reconstruction; sampling and the limited data problem; matrix methods, singular value decomposition and data compression; optimization techniques in signal and image reconstruction from projections; autocorrelations and power spectra; high resolution methods; detection and optimal filtering; eigenvector-based methods for array processing and statistical filtering.

# Chapter 2

# Complex Numbers

It is standard practice in signal processing to employ complex numbers whenever possible. One of the main reasons for doing this is that it enables us to represent the important sine and cosine functions in terms of complex exponential functions and to replace trigonometric identities with the somewhat simpler rules for the manipulation of exponents.

The complex numbers are the points in the $x, y$-plane: the complex number $z = (a, b)$ is identified with the point in the plane having $a = Re(z)$, the *real part* of $z$, for its $x$-coordinate and $b = Im(z)$, the *imaginary part* of $z$, for its $y$-coordinate. We call $(a, b)$ the *rectangular form* of the complex number $z$. The *conjugate* of the complex number $z$ is $\overline{z} = (a, -b)$. We can also represent $z$ in its polar form: let the *magnitude* of $z$ be $|z| = \sqrt{a^2 + b^2}$ and the *phase angle* of $z$, denoted $\theta(z)$, be the angle in $[0, 2\pi)$ with $\cos\theta(z) = a/|z|$. Then the *polar form* for $z$ is

$$z = (|z|\cos\theta(z), |z|\sin\theta(z)).$$

Any complex number $z = (a, b)$ for which the imaginary part $Im(z) = b$ is zero is identified with (treated as the same as) its real part $Re(z) = a$; that is, we identify $a$ and $z = (a, 0)$. These real complex numbers lie along the $x$-axis in the plane, the so-called *real line*. If this were the whole story complex numbers would be unimportant; but they are not. It is the arithmetic associated with complex numbers that makes them important.

We add two complex numbers using their rectangular representations:

$$(a, b) + (c, d) = (a + c, b + d).$$

This is the same formula used to add two-dimensional vectors. We multiply complex numbers more easily when they are in their polar representations: the product of $z$ and $w$ has $|z||w|$ for its magnitude and $\theta(z)+\theta(w)$ modulo $2\pi$ for its phase angle. Notice that the complex number $z = (0, 1)$ has

$\theta(z) = \pi/2$ and $|z| = 1$, so $z^2 = (-1, 0)$, which we identify with the real number $-1$. This tells us that within the realm of complex numbers the real number $-1$ has a square root, $i = (0, 1)$; note that $-i = (0, -1)$ is also a square root of $-1$.

To multiply $z = (a, b) = a + ib$ by $w = (c, d) = c + id$ in rectangular form we simply multiply the binomials

$$(a + ib)(c + id) = ac + ibc + iad + i^2bd$$

and recall that $i^2 = -1$ to get

$$zw = (ac - bd, bc + ad).$$

If $(a, b)$ is real, that is, if $b = 0$, then $(a, b)(c, d) = (a, 0)(c, d) = (ac, ad)$, which we also write as $a(c, d)$. Therefore, we can rewrite the polar form for $z$ as

$$z = |z|(\cos\theta(z), \sin\theta(z)) = |z|(\cos\theta(z) + i\sin\theta(z)).$$

We will have yet another way to write the polar form of $z$ when we consider the complex exponential function.

**Exercise 1:** Derive the formula for dividing one complex number in rectangular form by another (non-zero) one.

**Exercise 2:** Show that for any two complex numbers $z$ and $w$ we have

$$|zw| \geq \frac{1}{2}(z\overline{w} + \overline{z}w). \tag{2.1}$$

**Hint:** Write $|zw|$ as $|z\overline{w}|$.

**Exercise 3:** Show that, for any constant $a$ with $|a| \neq 1$, the function

$$G(z) = \frac{z - \overline{a}}{1 - az}$$

has $|G(z)| = 1$ whenever $|z| = 1$.

# Chapter 3

# Complex Exponentials

The most important function in signal processing is the complex-valued function of the real variable $x$ defined by

$$h(x) = \cos(x) + i\sin(x). \tag{3.1}$$

For reasons that will become clear shortly, this function is called the *complex exponential function*. Notice that the magnitude of the complex number $h(x)$ is always equal to one, since $\cos^2(x) + \sin^2(x) = 1$ for all real $x$. Since the functions $\cos(x)$ and $\sin(x)$ are $2\pi$-periodic, that is, $\cos(x+2\pi) = \cos(x)$ and $\sin(x+2\pi) = \sin(x)$ for all $x$, the complex exponential function $h(x)$ is also $2\pi$-periodic.

In calculus we encounter functions of the form $g(x) = a^x$, where $a > 0$ is an arbitrary constant. These functions are the *exponential* functions, the most well known of which is the function $g(x) = e^x$. Exponential functions are those with the property $g(u+v) = g(u)g(v)$ for every $u$ and $v$. We show now that the function $h(x)$ in equation (3.1) has this property, so must be an exponential function; that is, $h(x) = c^x$ for some constant $c$. Since $h(x)$ has complex values, the constant $c$ cannot be a real number, however.

Calculating $h(u)h(v)$ we find

$$h(u)h(v) = (\cos(u)\cos(v) - \sin(u)\sin(v)) + i(\cos(u)\sin(v) + \sin(u)\cos(v))$$

$$= \cos(u+v) + i\sin(u+v) = h(u+v).$$

So $h(x)$ is an exponential function; $h(x) = c^x$ for some complex constant $c$. Inserting $x = 1$ we find that $c$ is

$$c = \cos(1) + i\sin(1).$$

Let's try to find another way to express $c$.

Recall from calculus that for exponential functions $g(x) = a^x$ with $a > 0$ the derivative $g'(x)$ is

$$g'(x) = a^x \ln(a) = g(x) \ln(a).$$

Since

$$h'(x) = -\sin(x) + i\cos(x) = i(\cos(x) + i\sin(x)) = ih(x)$$

we conjecture that $\ln(c) = i$; but what does this mean?

For $a > 0$ we know that $b = \ln(a)$ means that $a = e^b$. Therefore, we say that $\ln(c) = i$ means $c = e^i$; but what does it mean to take $e$ to a complex power? To define $e^i$ we turn to the Taylor series representation for the exponential function $g(x) = e^x$, defined for real $x$:

$$e^x = 1 + x + x^2/2! + x^3/3! + ....$$

Inserting $i$ in place of $x$ and using the fact that $i^2 = -1$, we find that

$$e^i = (1 - 1/2! + 1/4! - ...) + i(1 - 1/3! + 1/5! - ...);$$

note that the two series are the Taylor series for $\cos(1)$ and $\sin(1)$, respectively, so $e^i = \cos(1) + i\sin(1)$. Then the complex exponential function in equation (3.1) is

$$h(x) = (e^i)^x = e^{ix}.$$

Inserting $x = \pi$ we get

$$h(\pi) = e^{i\pi} = \cos(\pi) + i\sin(\pi) = -1$$

or

$$e^{i\pi} + 1 = 0,$$

which is the remarkable relation discovered by Euler that combines the five most important constants in mathematics, $e$, $\pi$, $i$, 1 and 0, in a single equation.

Note that $e^{2\pi i} = e^{0i} = e^0 = 1$, so

$$e^{(2\pi+x)i} = e^{2\pi i} e^{ix} = e^{ix}$$

for all $x$.

We know from calculus what $e^x$ means for real $x$ and now we also know what $e^{ix}$ means. Using these we can define $e^z$ for any complex number $z = a + ib$ by $e^z = e^{a+ib} = e^a e^{ib}$.

We know from calculus how to define $\ln(x)$ for $x > 0$ and we have just defined $\ln(c) = i$ to mean $c = e^i$. But we could also say that $\ln(c) = i(1 + 2\pi k)$ for any integer $k$; that is, the periodicity of the complex exponential function forces the function $\ln(x)$ to be multivalued.

For any nonzero complex number $z = |z|e^{i\theta(z)}$ we have

$$\ln(z) = \ln(|z|) + \ln(e^{i\theta(z)}) = \ln(|z|) + i(\theta(z) + 2\pi k),$$

for any integer $k$. If $z = a > 0$ then $\theta(z) = 0$ and $\ln(z) = \ln(a) + i(k\pi)$ for any even integer $k$; in calculus class we just take the value associated with $k = 0$. If $z = a < 0$ then $\theta(z) = \pi$ and $\ln(z) = \ln(-a) + i(k\pi)$ for any odd integer $k$. So we can define the logarithm of a negative number; it just turns out not to be a real number. If $z = ib$ with $b > 0$, then $\theta(z) = \frac{\pi}{2}$ and $\ln(z) = \ln(b) + i(\frac{\pi}{2} + 2\pi k)$, for any integer $k$; if $z = ib$ with $b < 0$ then $\theta(z) = \frac{3\pi}{2}$ and $\ln(z) = \ln(-b) + i(\frac{3\pi}{2} + 2\pi k)$ for any integer $k$.

Adding $e^{-ix} = \cos(x) - i\sin(x)$ to $e^{ix}$ given by equation (3.1) we get

$$\cos(x) = \frac{1}{2}(e^{ix} + e^{-ix});$$

subtracting, we obtain

$$\sin(x) = \frac{1}{2i}(e^{ix} - e^{-ix}).$$

These formulas allow us to extend the definition of cos and sin to complex arguments $z$:

$$\cos(z) = \frac{1}{2}(e^{iz} + e^{-iz})$$

and

$$\sin(z) = \frac{1}{2i}(e^{iz} - e^{-iz}).$$

In signal processing the complex exponential function is often used to describe functions of time that exhibit periodic behavior:

$$h(\omega t + \theta) = e^{i(\omega t+\theta)} = \cos(\omega t + \theta) + i\sin(\omega t + \theta),$$

where the *frequency* $\omega$ and *phase angle* $\theta$ are real constants, and $t$ denotes time. We can alter the magnitude by multiplying $h(\omega t + \theta)$ by a positive constant $|A|$, called the *amplitude*, to get $|A|h(\omega t + \theta)$. More generally, we can combine the amplitude and the phase, writing

$$|A|h(\omega t + \theta) = |A|e^{i\theta}e^{i\omega t} = Ae^{i\omega t},$$

where $A$ is the complex amplitude $A = |A|e^{i\theta}$. Many of the functions encountered in signal processing can be modeled as linear combinations of such complex exponential functions or *sinusoids*, as they are often called.

**Exercise 1:** Show that if $\sin\frac{x}{2} \neq 0$ then

$$E_M(x) = \sum_{m=1}^{M} e^{imx} = e^{ix(\frac{M+1}{2})}\frac{\sin(Mx/2)}{\sin(x/2)}. \tag{3.2}$$

**Hint:** Note that $E_M(x)$ is the geometric progression

$$E_M(x) = e^{ix} + (e^{ix})^2 + (e^{ix})^3 + ... + (e^{ix})^M = e^{ix}(1 - e^{iMx})/(1 - e^{ix}).$$

Now use the fact that, for any $t$, we have

$$1 - e^{it} = e^{it/2}(e^{-it/2} - e^{it/2}) = e^{it/2}(-2i)\sin(t/2).$$

**Exercise 2:** The *Dirichlet kernel* of size $M$ is defined as

$$D_M(x) = \sum_{m=-M}^{M} e^{imx}.$$

Use equation (3.2) to obtain the closed-form expression

$$D_M(x) = \frac{\sin((M + \frac{1}{2})x)}{\sin(\frac{x}{2})};$$

note that $D_M(x)$ is real-valued.
**Hint:** Reduce the problem to that of Exercise 1 by factoring appropriately.

**Exercise 3:** Use the result in equation (3.2) to obtain the closed-form expressions

$$\sum_{m=N}^{M} \cos mx = \cos(\frac{M+N}{2}x)\frac{\sin(\frac{M-N+1}{2}x)}{\sin\frac{x}{2}}$$

and

$$\sum_{m=N}^{M} \sin mx = \sin(\frac{M+N}{2}x)\frac{\sin(\frac{M-N+1}{2}x)}{\sin\frac{x}{2}}.$$

**Hint**: Recall that $\cos mx$ and $\sin mx$ are the real and imaginary parts of $e^{imx}$.

**Exercise 4:** Graph the function $E_M(x)$ for various values of $M$.

We note in passing that the function $E_M(x)$ equals $M$ for $x = 0$ and equals zero for the first time at $x = 2\pi/M$. This means that the *main lobe* of $E_M(x)$, the inverted parabola-like portion of the graph centered at $x = 0$, crosses the $x$-axis at $x = 2\pi/M$ and $x = -2\pi/M$, so its height is $M$ and its width is $4\pi/M$. As $M$ grows larger the main lobe of $E_M(x)$ gets higher and thinner.

# Chapter 4

# Hidden Periodicities

We begin with what we call the *Ferris Wheel Problem.* A Ferris Wheel is a carnival ride, or perhaps a tourist attraction, like the London Eye, consisting of a large rotating wheel supported so that its axis of rotation is parallel to the ground. Around the rim of the wheel are seats for the riders. Once the seats are filled the wheel rotates for some number of minutes, from time $t = 0$ to $t = T$ and then it slows to let the riders off. Suppose that the radius of the wheel is $R$ feet, the center of the wheel is $R + H$ feet off the ground and from time $t = 0$ to $t = T$ the wheel completes one revolution in $P$ seconds, so that its frequency of rotation is $\omega = \frac{2\pi}{P}$ radians per second.

**Exercise 1:** Determine the formulas giving the horizontal and vertical coordinates of the position of a particular rider at an arbitrary time $t$ in the time interval $[0, T]$.

Now let us make it a bit more complicated. Suppose that, instead of seats around the rim of the wheel, there is a smaller Ferris Wheel (or several identical smaller wheels distributed around the rim, for stability). To avoid confusion, let's let $R_1$ and $\omega_1$ be the radius and frequency of rotation of the original wheel and let $R_2$ and $\omega_2$ be the radius and frequency of rotation of the second wheel.

**Exercise 2:** Now find the formulas giving the horizontal and vertical coordinates of the position of a particular rider at an arbitrary time $t$ in the time interval $[0, T]$.

Continuing down this road, imagine a third wheel on the rim of the second, a fourth on the rim of the third, and so on; in fact, let there be $J$ nested Ferris wheels, the $j$-th wheel having radius $R_j$ and frequency of rotation $\omega_j$. Figure 4.1 illustrates the case of $J = 3$.

**Exercise 3:** Repeat the previous exercise, but for the case of $J$ nested wheels.

What we have been doing here is solving what is called a *direct problem*. The simplest way to explain a direct problem is to contrast it with one that is not direct, a so-called *inverse problem* [104], [177]. An inverse problem involving the Ferris Wheels is the following. Suppose our data consists of the positions of a particular rider at several distinct times, $t_1, ..., t_M$. From this data alone determine $J$, the number of nested wheels, the radii $R_j$ of the wheels, and their frequencies of rotation $\omega_j$.

Direct problems usually look ahead in time to what would happen in a certain situation. The formulas involved are usually straightforward applications of the relevant concepts and there is no data involved. In contrast, inverse problems ask us to determine what did happen, given some measurements of the outcome. The measurements may be unreliable or noisy and there may not be enough measurements to determine a single unique answer. In the inverse Ferris Wheel problem we would assume that $J$, the number of wheels, is smaller than $M$, the number of measurements. Given $M$ measurements, it is usually possible to fit those measurements exactly to a model involving more than $M$ wheels; the hard part is to let the data tell us what $J$ is. A second issue is the choosing of the times $t_m$ at which the measurements are taken. If we were to take all the measurements in rapid succession, over a very small interval of time, the problem would become much more difficult and the answer much more sensitive to slight errors in the data. Just how we should select the times $t_m$ will depend on our prior knowledge of what the frequencies of rotation might be. If some of the wheels are turning very rapidly we must sample quickly to determine that. Otherwise we get the *strobe light* type of aliasing.

The measured data giving the positions of the rider at various times is said to contain information about the *hidden periodicities* involved. There are periodicites, not always hidden, in many different data sets. For example, data giving the temperature every hour in downtown Lowell for the last one hundred million years would show several interest periodicities, or almost periodicities. Clearly there is the periodicity corresponding to the seasons of the year. There is also the periodicity associated with the passage from day to night, although this is a somewhat more complicated function of time, involving, as it does, the varying lengths of day and night in different seasons. There will be other components corresponding to the temperature changes from one day to the next, having no simple periodic aspect. On top of all this there will be components with much longer periods (so much smaller frequencies), corresponding to the climate changes from one century to the next. There will be components with even longer periods, the climate changes studied in connection with global warming, having periods of thousands of years. An interesting study is to try to

relate, or to *correlate*, the periodic components in one data set with those in another. For example, is earth weather related to the periodicities in the sun spot activity?

Many of the signals we encounter in practice contain complex exponential components having different amplitudes and frequencies. The standard model for such signals is

$$s(t) = \sum_{n=1}^{N} |A_n| e^{i(\omega_n t + \theta_n)}. \tag{4.1}$$

One of the main problems in signal processing is to determine the values of the parameters $N$, $|A_n|$, $\omega_n$ and $\theta_n$ from measurements of the function $s(t)$; that is, to determine the complex exponential components that constitute the signal $s(t)$. For example, in automated human voice recognition a particular individual speaker is identified by the combination of the $|A_n|$ and $\omega_n$ present in the speech of that person when pronouncing a certain sound. Our ears perform this identification task when we recognize the voice of a particular singer or actor. In digital speech processing the assumption is that the signal corresponding to the voicing of a particular sound has the form given in equation (4.1), at least for a short time interval (until the next sound is voiced). A second point of view is that equation (4.1) is a model to be used to perform certain operations on a signal, such as noise reduction or compression.

In some applications we do not have exact measurements of $s(t)$ but noisy estimates of what those exact values are. Our job is then to clean up the data to extract the parameter values. In restoration of old recordings the parameters are estimated from noisy measurements of the old recording and these parameters modified and inserted to recreate digitally the original sound. The noisy measurement data can then be modeled using equation (4.1) and (at least some of ) the noise removed by subtracting certain complex exponential components attributed to the noise. At the same time the quality of the signal can be enhanced by modifying the amplitudes of the components that remain. The resulting set of numbers can then be converted back into audible sound.

In radar, sonar, radio astronomy and related remote sensing applications the variable $\omega$ may not be frequency but a direction in space relative to a fixed coordinate system. In such cases the variable $t$ denotes the location in space at which the function $s(t)$ is measured. The various parts of the objects of interest send (or reflect) individual signals and the measuring devices record the superposition of all these signals. Whether the objects of interest are planes in radar, the stars in the heavens in optical or radio astronomy, submarines and ships at sea in sonar, regions of a patient's body in medical tomography or portions of the earth's surface in synthetic aperture radar imaging, the received signals must be analyzed, that is, broken down into their constituent parts, so that the individual sources of received

energy can be separately known. A nonzero value of $|A_n|$ then indicates the presence of a source (or reflector) of electromagnetic or acoustic energy at angle $\omega_n$. We measure $s(t)$ at many different locations $t$ and from that data we try to decompose the signal into its components. How well we are able to identify separate sources of energy is the *resolving capablity* of the process. Our ability to resolve will depend on several things, including the hardware we use, where we are able to measure $s(t)$ and at how many values of $t$ we are able to employ, and also the mathematical methods we use to perform the analysis of the signal.

Common to each of these applications is the need to isolate the individual complex exponential components in the measured signal. This is the *signal analysis* problem, which we consider next.

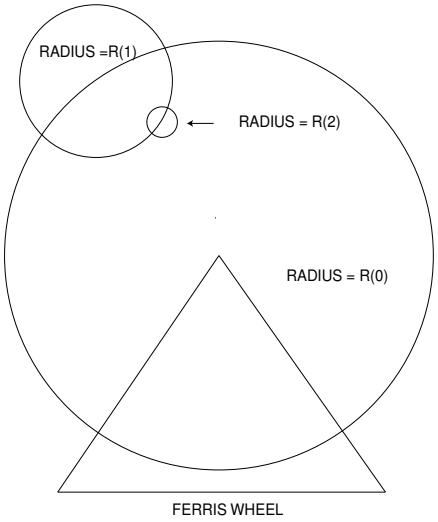



Figure 4.1: The Ferris Wheel for $J = 3$.

# Chapter 5

# Signal Analysis: A First Approach

We shall assume now that the signal we wish to analyze is $s(t)$ given by equation (4.1), which we rewrite as

$$s(t) = \sum_{n=1}^{N} A_n e^{i\omega_n t}, \tag{5.1}$$

with $A_n = |A_n|e^{i\theta_n}$ the complex amplitudes. Although we shall often speak of $t$ as a time variable, that is not essential. We assume that we have determined the value of the function $s(t)$ at $M$ points in time, called the *sampling times*. Although it is not necessary, we shall assume the sampling times are equispaced, that is, they are $t = m\Delta$, $m = 1, ..., M$, where $\Delta > 0$ is the difference between successive sampling times. So our data are the values $s(m\Delta)$, $m = 1, ..., M$. Our goal is to determine $N$, the number of complex exponential components in the signal $s(t)$, their complex amplitudes $A_n$ and the frequencies $\omega_n$. We assume that $N$ is smaller than $M$.

**The aliasing problem:** Given our data, it is impossible for us to distinguish a frequency $\omega$ from $\omega + \frac{2\pi n}{\Delta}$, for any integer $n$. This can result in *aliasing*, if the sample spacing $\Delta$ is not sufficiently small.

For every $m$ we have

$$e^{i\omega_n m\Delta} = e^{i(\omega_n + 2\pi/\Delta)m\Delta},$$

which tells us that, using the data we have, we cannot distinguish between the frequencies $\omega_n$ and $\omega_n + 2\pi/\Delta$. We shall therefore make the assumption that $\Delta$ has been selected small enough so that $|\omega_n| \leq \pi/\Delta$ for all $n$. If we have not selected $\Delta$ small enough, we have *undersampled* and some of the

frequencies $\omega_n$ will be mistaken for lower frequencies; this is the *aliasing problem*. We describe now an approach that determines $N$, the $\omega_n$ and the $A_n$ well enough if the data is relatively noise-free, none of the $\omega_n$ are too close to one another and the $M$ is large enough.

**Our assumption:** Our first approach to solving the signal analysis problem is based on a simplifying restriction on the possible locations of the frequencies $\omega_n$. We assume that the $\omega_n$ are some of the members of the set $\{\alpha_k = -\frac{\pi}{\Delta} + k\frac{2\pi}{\Delta M},\ k = 1, 2, ..., M\}$; these are the $M$ frequencies equispaced across the interval $(-\frac{\pi}{\Delta}, \frac{\pi}{\Delta}]$. We then rewrite $s(t)$ as

$$s(t) = \sum_{k=1}^{M} B_k e^{i\alpha_k t}; \tag{5.2}$$

values of $k$ for which the $B_k$ are not zero will be the ones for which $\alpha_k$ is one of the original $\omega_n$ and $B_k = A_n$. Our data is then

$$s(m\Delta) = \sum_{k=1}^{M} B_k e^{-im\pi} e^{i2\pi km/M},$$

for $m = 1, ..., M$.

**The complex vector dot product :** For any positive integer $J$ and any two $J$ dimensional complex column vectors $\mathbf{u}$ and $\mathbf{v}$ we define the *complex vector dot product* to be

$$\mathbf{u} \cdot \mathbf{v} = \sum_{j=1}^{J} u_j \overline{v_j}.$$

Note that $\mathbf{u} \cdot \mathbf{v} = \mathbf{v}^\dagger \mathbf{u}$, where $\mathbf{v}^\dagger$, the *conjugate transpose* of the vector $\mathbf{v}$, is the row vector whose entries are the conjugates of the entries of the vector $\mathbf{v}$. Therefore, we can and do view the complex vector dot product as a special case of matrix multiplication.

As we shall see in a later chapter on the Cauchy inequality, the dot product is a way of checking how well two vectors resemble one another. This idea is used extensively in signal processing, when we form the dot product between the data vector and each of many potential component vectors, to see how much the data resembles each of them. This is called *matching* and is the basic idea in *matched filtering*, as we shall see later. We now apply this idea of matching in our first attempt at solving the signal analysis problem.

For each $j = 1, 2, ..., M$ we ask what data we would have collected had the signal $s(t)$ consisted solely of a single complex exponential $e^{i\alpha_j t}$ with frequency $\alpha_j$; the answer is $e^{i\alpha_j m\Delta}$, for $m = 1, 2, ..., M$. We now let these numbers be the entries of a vector we call $\mathbf{e}_j$; then we match $\mathbf{e}_j$ with the data vector $\mathbf{d}$ having the entries $s(m\Delta)$.

Therefore, for each $j = 1, 2, ..., M$, we let the entries of the column vector $\mathbf{e}_j$ be

$$e_{jm} = e^{i\alpha_j m\Delta} = e^{-im\pi} e^{i2\pi jm/M}.$$

Let $\mathbf{e}_j^\dagger$ denote the conjugate transpose of $\mathbf{e}_j$, that is, the row vector whose entries are $\overline{e_{jm}}$, so that the matrix multiplication $\mathbf{e}_j^\dagger \mathbf{d}$ is the complex dot product of $\mathbf{e}_j$ and $\mathbf{d}$. Then

$$\mathbf{e}_j^\dagger \mathbf{d} = \sum\nolimits_{m=1}^{M} s(m\Delta) e^{-i\alpha_j m\Delta} = \sum\nolimits_{k=1}^{M} B_k (\sum\nolimits_{m=1}^{M} e^{2\pi i(k-j)m/M}).$$

The inner sum is $E_M(x)$ for $x = 2\pi(k-j)/M$, so we can use the closed form of this sum that we derived in an exercise earlier to conclude that the inner sum equals $M$ if $k = j$ and is zero if $k \neq j$. Therefore, for each fixed $j$, as we run through the index of summation $k$, all the terms being added are zero, except when the index $k$ reaches the fixed value $j$. Therefore

$$\mathbf{e}_j^\dagger \mathbf{d} = MB_j$$

for each $j$. To isolate the original frequencies $\omega_n$ we select those $j$ for which $\mathbf{e}_j^\dagger \mathbf{d}$ is not zero; then the $A_n$ is the associated value $B_j$.

So we know how to isolate the individual complex exponential components of $s(t)$, so long as each of the $\omega_n$ is, at least approximately, one of the $\alpha_k$, which imposes the constraint that no two of the $\omega_n$ are closer to each other than $2\pi/\Delta M$; this limits our ability to resolve components whose frequencies are closer than that limit. If we know in advance that we are seeking frequencies $\omega_n$ closer than this limit we have at least two choices: increase $M$ or increase $\Delta$. The latter choice is a bit dangerous in that we risk aliasing if any of the $\omega_n$ have magnitudes close to $\pi/\Delta$ already. A third choice is to alter the method whereby we isolated the individual components. There are many ways to do this, as we shall see.

# Chapter 6

# Convolution and the Vector DFT

Convolution is an important concept in signal processing and occurs in several distinct contexts. In this chapter we shall discuss *non-periodic convolution* and *periodic convolution* of vectors. Later we shall consider the convolution of infinite sequences and of functions of a continuous variable. The reader may recall an earlier encounter with convolution in a course on differential equations. The simplest example of convolution is the non-periodic convolution of finite vectors.

### Non-periodic convolution:

Recall the algebra problem of multiplying one polynomial by another. Suppose

$$A(x) = a_0 + a_1 x + ... + a_M x^M$$

and

$$B(x) = b_0 + b_1 x + ... + b_N x^N.$$

Let $C(x) = A(x)B(x)$. With

$$C(x) = c_0 + c_1 x + ... + c_{M+N} x^{M+N},$$

each of the coefficients $c_j$, $j = 0, ..., M+N$, can be expressed in terms of the $a_m$ and $b_n$ (an easy exercise!). The vector $c = (c_0, ..., c_{M+N})$ is called the *non-periodic convolution* of the vectors $a = (a_0, ..., a_M)$ and $b = (b_0, ..., b_N)$. Non-periodic convolution can be viewed as a particular case of periodic convolution, as we see next.

## The DFT and the vector DFT:

As we just discussed, non-periodic convolution is another way of looking at the multiplication of two polynomials. This relationship between convolution on the one hand and multiplication on the other is a fundamental aspect of convolution, whenever it occurs. Whenever we have a convolution we should ask what related mathematical objects are being multiplied. We ask this question now with regard to periodic convolution; the answer turns out to be the *vector discrete Fourier transform*.

Given the $N$ by 1 vector $\mathbf{f}$ with complex entries $f_0, f_1, ..., f_{N-1}$ define the *discrete Fourier transform* (DFT) of $\mathbf{f}$ to be the function $DFT_{\mathbf{f}}(\omega)$, defined for $\omega$ in $[0, 2\pi)$, by

$$DFT_{\mathbf{f}}(\omega) = \sum_{n=0}^{N-1} f_n e^{in\omega}.$$

The terminology can be confusing, since the expression 'discrete Fourier transform' is often used to describe several slightly different mathematical objects.

For example, in the exercise that follows we are interested solely in the values $F_k = DFT_{\mathbf{f}}(2\pi k/N)$, for $k = 0, 1, ..., N-1$. In this case the DFT of the vector $\mathbf{f}$ often means simply the vector $\mathbf{F}$ whose entries are the complex numbers $F_k$, for $k = 0, ..., N-1$; for the moment let us call this the *vector DFT* of $\mathbf{f}$ and write $\mathbf{F} = vDFT_{\mathbf{f}}$. The point of Exercise 1 is to show how to use the vector DFT to perform the *periodic convolution* operation.

In some instances the numbers $f_n$ are obtained by evaluating a function $f(x)$ at some finite number of points $x_n$; that is, $f_n = f(x_n)$, for $n = 0, ..., N-1$. As we shall see later, if the $x_n$ are equispaced, the DFT provides an approximation of the Fourier transform of the function $f(x)$. Since the Fourier transform is another function of a continuous variable, and not a vector, it is appropriate, then, to view the DFT also as such a function. Since the practice is to use the term DFT to mean slightly different things in different contexts, we adopt that practice here. The reader will have to infer the precise meaning of DFT from the context.

## Periodic convolution:

Given the $N$ by 1 vectors $\mathbf{f}$ and $\mathbf{d}$ with complex entries $f_n$ and $d_n$, respectively, we define a third $N$ by 1 vector $\mathbf{f} * \mathbf{d}$, the *periodic convolution* of $\mathbf{f}$ and $\mathbf{d}$, to have the entries

$$(\mathbf{f} * \mathbf{d})_n = f_0 d_n + f_1 d_{n-1} + ... + f_n d_0 + f_{n+1} d_{N-1} + ... + f_{N-1} d_{n+1}.$$

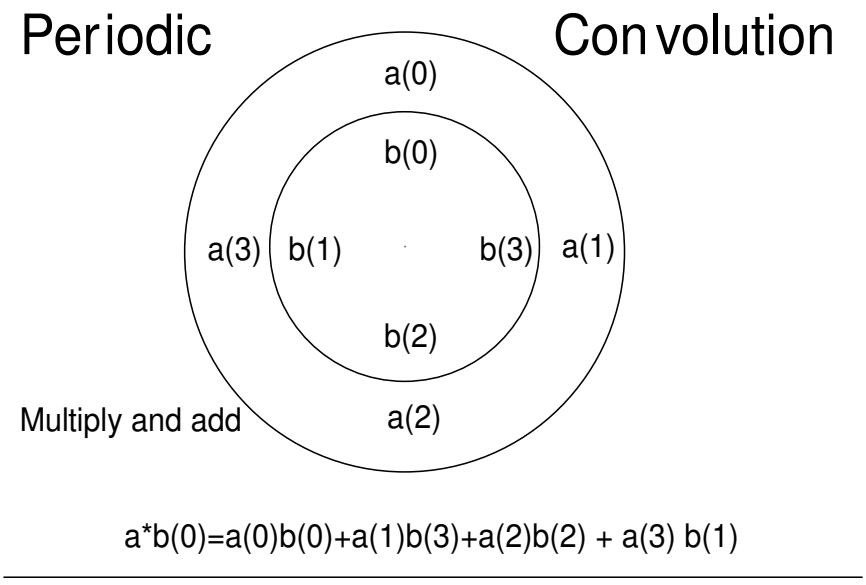
Periodic convolution is illustrated in Figure 6.1. The first exercise relates the periodic convolution to the vector DFT.

**Exercise 1:** Let $\mathbf{F} = vDFT_{\mathbf{f}}$ and $\mathbf{D} = vDFT_{\mathbf{d}}$. Define a third vector $\mathbf{E}$ having for its $k$-th entry $E_k = F_k D_k$, for $k = 0, ..., N-1$. Show that $\mathbf{E}$ is the vDFT of the vector $\mathbf{f} * \mathbf{d}$.

The vector $vDFT_{\mathbf{f}}$ can be obtained from the vector $\mathbf{f}$ by means of matrix multiplcation by a certain matrix $G$, called the *DFT matrix*. The matrix $G$ has an inverse that is easily computed and can be used to go from $\mathbf{F} = vDFT_{\mathbf{f}}$ back to the original $\mathbf{f}$. The details are in Exercise 2.

**Exercise 2:** Let $G$ be the $N$ by $N$ matrix whose entries are $G_{jk} = e^{i(j-1)(k-1)2\pi/N}$. The matrix $G$ is sometimes called the *DFT matrix*. Show that the inverse of $G$ is $G^{-1} = \frac{1}{N}G^{\dagger}$, where $G^{\dagger}$ is the conjugate transpose of the matrix $G$. Then $\mathbf{f} * \mathbf{d} = G^{-1}\mathbf{E} = \frac{1}{N}G^{\dagger}\mathbf{E}$.

As we mentioned above, nonperiodic convolution is really a special case of periodic convolution. Extend the $M+1$ by 1 vector $a$ to an $M+N+1$ by 1 vector by appending $N$ zero entries; similarly, extend the vector $b$ to an $M+N+1$ by 1 vector by appending zeros. The vector $c$ is now the periodic convolution of these extended vectors. Therefore, since we have an efficient algorithm for performing periodic convolution, namely the Fast Fourier Transform algorithm (FFT), we have a fast way to do the periodic (and thereby nonperiodic) convolution and polynomial multiplication.



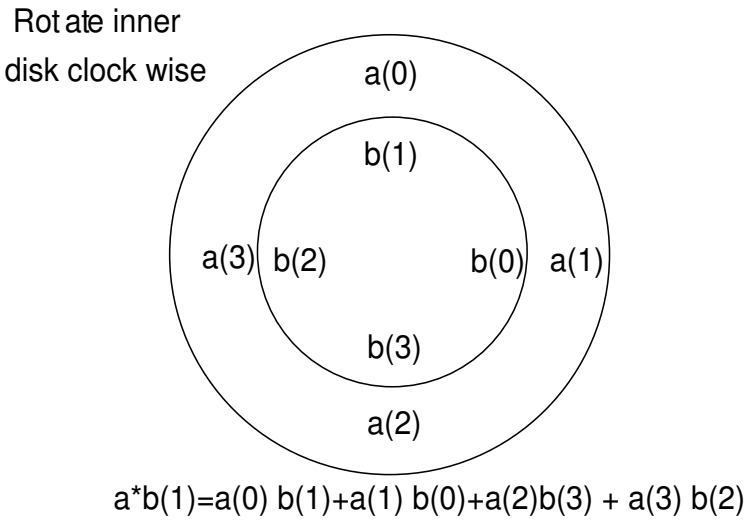




Figure 6.1: Periodic convolution of vectors $a = (a(0), a(1), a(2), a(3))$ and $b = (b(0), b(1), b(2), b(3))$.

# Chapter 7

# Signal Analysis: A Second Approach

As before, we assume that we have data vector $\mathbf{d}$ with entries $s(m\Delta)$, $m = 1, ..., M$ from the signal $s(t)$ given by equation (26.9). Unlike in our first approach, we do not now make any assumptions about the location of the frequencies $\omega_n$, except that $|\omega_n| < \pi/\Delta$.

For each $\omega$ in the interval $(-\pi/\Delta, \pi/\Delta)$ let $e_\omega$ be the column vector with entries $e^{i\omega m\Delta}$, $m = 1, ..., M$. The output of the matched filter $\mathbf{e}_\omega^\dagger \mathbf{d}$, as a function of the continuous variable $\omega$ in the interval $(-\pi/\Delta, \pi/\Delta)$ is

$$DFT_{\mathbf{d}}(\omega) = \sum_{m=1}^{M} s(m\Delta)e^{-i\omega m\Delta}$$

$$= \sum_{n=1}^{N} A_n(\sum_{m=1}^{M} e^{i(\omega_n-\omega)m\Delta}).$$

We know from our earlier calculations that

$$\sum_{m=1}^{M} e^{i(\omega_n-\omega)m\Delta} = e^{i\frac{M+1}{2}(\omega_n-\omega)} \sin(\frac{M}{2}(\omega_n - \omega))/(\sin\frac{1}{2}(\omega_n - \omega)),$$

which equals $M$ if $\omega = \omega_n$. If the $\omega_n$ are well separated then this sum is significantly smaller if $\omega$ is not near $\omega_n$. So if the $\omega_n$ are well separated and $M$ is significantly larger than $N$ the function $DFT_{\mathbf{d}}(\omega)$ will be near $MA_n$ when $\omega = \omega_n$, for each $n$, and will be near zero otherwise. Of course we cannot calculate $DFT_{\mathbf{d}}(\omega)$ for each $\omega$; for the purposes of plotting we select sufficiently many values of $\omega$ and calculate $|DFT_{\mathbf{d}}(\omega)|$ at these points. Later we shall study a fast algorithm, known as the *fast Fourier transform* (FFT), which does this calculation for us in an efficient manner.

**Exercise 1:** Let $N = 2$ and $\omega_1 = -\alpha$, $\omega_2 = \alpha$ for some $\alpha > 0$ in $(-\pi, \pi)$. Let $A_1 = A_2 = 1$. Select a value of $M$ that is greater than two and

calculate the values $f(m)$ for $m = 1, ..., M$. Plot the graph of the function $DFT_{\mathbf{d}}(\omega)$ on $(-\pi, \pi)$. Repeat the exercise for various values of $M$ and values of $\alpha$ closer to zero. Notice how $DFT_{\mathbf{d}}(0)$ behaves as $\alpha$ goes to zero. For each fixed value of $M$ there will be a critical value of $\alpha$ such that, for any smaller values of $\alpha$, $DFT_{\mathbf{d}}(0)$ will be larger than $DFT_{\mathbf{d}}(\alpha)$. This is *loss of resolution*.

As the exercise has shown, for each fixed value of $M$ there will be a limit to our ability to resolve closely spaced frequencies using $DFT_{\mathbf{d}}(\omega)$. If we are unable to increase the $M$ we can try other methods of isolating the frequencies. We shall discuss these other methods later.

# Chapter 8

# Cauchy's Inequality

So far our methods for analyzing the measured signal have been based on the idea of matching the data against various potential complex exponential components to see which ones match best. The matching is done using the complex dot product, $\mathbf{e}_\omega^\dagger \mathbf{d}$. In the ideal case this dot product is large, for those values of $\omega$ that correspond to an actual component of the signal; otherwise it is small. Why this should be the case is the Cauchy-Schwarz inequality (or sometimes, depending on the context, just Cauchy's inequality, just Schwarz's inequality, or, in the Russian literature, Bunyakovsky's inequality).

**The complex vector dot product:** Let $\mathbf{u} = (a, b)$ and $\mathbf{v} = (c, d)$ be two vectors in two-dimensional space. Let $\mathbf{u}$ make the angle $\alpha > 0$ with the positive $x$-axis and $\mathbf{v}$ the angle $\beta > 0$. Let $||\mathbf{u}|| = \sqrt{a^2 + b^2}$ denote the length of the vector $\mathbf{u}$. Then $a = ||\mathbf{u}|| \cos\alpha$, $b = ||\mathbf{u}|| \sin\alpha$, $c = ||\mathbf{v}|| \cos\beta$ and $d = ||\mathbf{v}|| \sin\beta$. So $\mathbf{u} \cdot \mathbf{v} = ac + bd = ||\mathbf{u}||||\mathbf{v}||(\cos\alpha\cos\beta + \sin\alpha\sin\beta = ||\mathbf{u}||\,||\mathbf{v}||\cos(\alpha - \beta)$. Therefore, we have

$$\mathbf{u} \cdot \mathbf{v} = ||\mathbf{u}||\,||\mathbf{v}||\cos\theta, \tag{8.1}$$

where $\theta = \alpha - \beta$ is the angle between $\mathbf{u}$ and $\mathbf{v}$. Cauchy's inequality is

$$|\mathbf{u} \cdot \mathbf{v}| \leq ||\mathbf{u}||\,||\mathbf{v}||,$$

with equality if and only if $\mathbf{u}$ and $\mathbf{v}$ are parallel.

Cauchy's inequality extends to vectors of any size with complex entries. For example, the complex $M$-dimensional vectors $\mathbf{e}_\omega$ and $\mathbf{e}_\theta$ defined earlier both have length equal to $\sqrt{M}$ and

$$|\mathbf{e}_\omega^\dagger \mathbf{e}_\theta| \leq M,$$

with equality if and only if $\omega$ and $\theta$ differ by an integer multiple of $\pi$.

From equation (8.1) we know that the dot product $\mathbf{u} \cdot \mathbf{v}$ is zero if and only if the angle between these two vectors is a right angle; we say then that $\mathbf{u}$ and $\mathbf{v}$ are mutually *orthogonal*. Orthogonality was at the core of our first approach to signal analysis: the vectors $\mathbf{e}_j$ and $\mathbf{e}_k$ are orthogonal if $k \neq j$. The notion of orthogonality is fundamental in signal processing and we shall return to it repeatedly in what follows. The idea of using the dot product to measure how similar two vectors are is called *matched filtering*; it is a popular method in signal detection and estimation of parameters.

**Proof of Cauchy's inequality:** To prove Cauchy's inequality for the complex vector dot product we write $\mathbf{u} \cdot \mathbf{v} = |\mathbf{u} \cdot \mathbf{v}|e^{i\theta}$. Let $t$ be a real variable and consider

$$\begin{aligned} 0 \leq ||e^{-i\theta}\mathbf{u} - t\mathbf{v}||^2 &= (e^{-i\theta}\mathbf{u} - t\mathbf{v}) \cdot (e^{-i\theta}\mathbf{u} - t\mathbf{v}) \\ &= ||\mathbf{u}||^2 - t[(e^{-i\theta}\mathbf{u}) \cdot \mathbf{v} + \mathbf{v} \cdot (e^{-i\theta}\mathbf{u})] + t^2||\mathbf{v}||^2 \\ &= ||\mathbf{u}||^2 - t[(e^{-i\theta}\mathbf{u}) \cdot \mathbf{v} + \overline{(e^{-i\theta}\mathbf{u}) \cdot \mathbf{v}}] + t^2||\mathbf{v}||^2 \\ &= ||\mathbf{u}||^2 - 2Re(te^{-i\theta}(\mathbf{u} \cdot \mathbf{v})) + t^2||\mathbf{v}||^2 \\ = ||\mathbf{u}||^2 - 2Re(t|\mathbf{u} \cdot \mathbf{v}|) + t^2||\mathbf{v}||^2 &= ||\mathbf{u}||^2 - 2t|\mathbf{u} \cdot \mathbf{v}| + t^2||\mathbf{v}||^2. \end{aligned}$$

This is a nonnegative quadratic polynomial in the variable $t$, so cannot have two distinct real roots. Therefore, the discriminant $4|\mathbf{u} \cdot \mathbf{v}|^2 - 4||\mathbf{v}||^2||\mathbf{u}||^2$ must be non-positive; that is, $|\mathbf{u} \cdot \mathbf{v}|^2 \leq ||\mathbf{u}||^2||\mathbf{v}||^2$. This is Cauchy's inequality.

**Exercise 1:** Use Cauchy's inequality to show that

$$||\mathbf{u} + \mathbf{v}|| \leq ||\mathbf{u}|| + ||\mathbf{v}||;$$

this is called the *triangle inequality*.

A careful examination of the proof just presented shows that we did not explicitly use the definition of the complex vector dot product, but only certain of its properties. This suggested to mathematicians the possibility of abstracting these properties and using them to define a more general concept, an *inner product*, between objects more general than complex vectors, such as infinite sequences, random variables and matrices. Such an inner product can then be used to define the *norm* of these objects and thereby a distance between such objects. Once we have an inner product defined we also have available the notions of orthogonality and best approximation. We shall treat all of these topics in a later chapter.

# Chapter 9

# Orthogonal Vectors

Consider the problem of writing the two-dimensional real vector $(3, -2)$ as a linear combination of the vectors $(1, 1)$ and $(1, -1)$; that is, we want to find constants $a$ and $b$ so that $(3, -2) = a(1, 1) + b(1, -1)$. One way to do this, of course, is to compare the components: $3 = a + b$ and $-2 = a - b$; we can then solve this simple system for the $a$ and $b$. In higher dimensions this way of doing it becomes harder, however. A second way is to make use of the dot product and orthogonality.

The dot product of two vectors $(x, y)$ and $(w, z)$ in $R^2$ is $(x, y) \cdot (w, z) = xw+yz$. If the dot product is zero then the vectors are said to be *orthogonal*; the two vectors $(1, 1)$ and $(1, -1)$ are orthogonal. We take the dot product of both sides of $(3, -2) = a(1, 1) + b(1, -1)$ with $(1, 1)$ to get

$$1 = (3, -2) \cdot (1, 1) = a(1, 1) \cdot (1, 1) + b(1, -1) \cdot (1, 1) = a(1, 1) \cdot (1, 1) + 0 = 2a,$$

so we see that $a = \frac{1}{2}$. Similarly, taking the dot product of both sides with $(1, -1)$ gives

$$5 = (3, -2) \cdot (1, -1) = a(1, 1) \cdot (1, -1) + b(1, -1) \cdot (1, -1) = 2b,$$

so $b = \frac{5}{2}$. Therefore $(3, -2) = \frac{1}{2}(1, 1) + \frac{5}{2}(1, -1)$. The beauty of this approach is that it does not get much harder as we go to higher dimensions.

Since the cosine of the angle $\theta$ between vectors $\mathbf{u}$ and $\mathbf{v}$ is

$$\cos\theta = \mathbf{u} \cdot \mathbf{v} / ||\mathbf{u}|| \, ||\mathbf{v}||,$$

where $||\mathbf{u}||^2 = \mathbf{u} \cdot \mathbf{u}$, the projection of vector $\mathbf{v}$ onto the line through the origin parallel to $\mathbf{u}$ is

$$\text{Proj}_{\mathbf{u}}(\mathbf{v}) = \frac{\mathbf{u} \cdot \mathbf{v}}{\mathbf{u} \cdot \mathbf{u}} \mathbf{u}.$$

Therefore the vector $\mathbf{v}$ can be written as

$$\mathbf{v} = \text{Proj}_{\mathbf{u}}(\mathbf{v}) + (\mathbf{v} - \text{Proj}_{\mathbf{u}}(\mathbf{v})),$$

where the first term on the right is parallel to **u** and the second one is orthogonal to **u**.

How do we find vectors that are mutually orthogonal? Suppose we begin with $(1,1)$. Take a second vector, say $(1,2)$, that is not parallel to $(1,1)$ and write it as we did **v** earlier; that is, as a sum of two vectors, one parallel to $(1,1)$ and the second orthogonal to $(1,1)$. The projection of $(1,2)$ onto the line parallel to $(1,1)$ passing through the origin is

$$\frac{(1,1)\cdot(1,2)}{(1,1)\cdot(1,1)}(1,1) = \frac{3}{2}(1,1) = (\frac{3}{2},\frac{3}{2})$$

so

$$(1,2) = (\frac{3}{2},\frac{3}{2}) + ((1,2) - (\frac{3}{2},\frac{3}{2})) = (\frac{3}{2},\frac{3}{2}) + (-\frac{1}{2},\frac{1}{2}).$$

The vectors $(-\frac{1}{2},\frac{1}{2}) = -\frac{1}{2}(1,-1)$ and, therefore, $(1,-1)$ are then orthogonal to $(1,1)$. This approach is the basis for the *Gram-Schmidt* method for constructing a set of mutually orthogonal vectors.

**Exercise 1:** Use the Gram-Schmidt approach to find a third vector in $R^3$ orthogonal to both $(1,1,1)$ and $(1,0,-1)$.

Orthogonality is a convenient tool that can be exploited whenever we have an inner product defined.

# Chapter 10

# Discrete Linear Filters

Let $\mathbf{g} = (g_1, ..., g_M)^T$ be an $M$-dimensional complex column vector. The discrete linear filter obtained from $\mathbf{g}$ operates on any other $M$-dimensional column vector $\mathbf{h} = (h_1, ..., h_M)^T$ through the complex dot product: when the input of the filter is $\mathbf{h}$ the output of the filter is

$$\mathbf{g}^\dagger \mathbf{h} = \mathbf{h} \cdot \mathbf{g} = \sum\nolimits_{m=1}^{M} h_m \overline{g}_m.$$

Earlier we analyzed the signal $s(t)$ by applying the discrete linear filters $\mathbf{g} = \mathbf{e}_\omega$ to the data vector $\mathbf{d}$ to obtain the function $\mathbf{e}_\omega^\dagger \mathbf{d}$ of the variable $\omega$. Such discrete linear filters are usually called *matched filters* because we use the dot product to determine the degree of similarity between the two vectors.

The term *discrete linear filter* also applies to the somewhat more general *convolution filter* whereby vectors $\mathbf{g}$ and $\mathbf{h}$ are used to produce a third vector $\mathbf{f} = \mathbf{g} * \mathbf{h}$, the periodic convolution of $\mathbf{g}$ and $\mathbf{h}$, whose entries $f_n$ are

$$f_n = \sum\nolimits_{m=1}^{M} g_m h_{n-m}, \tag{10.1}$$

where, for notational convenience, we define $h_{n-m} = h_{n-m+M}$ whenever the index $n - m$ is less than one. Figure 10.1 illustrates the action of this convolution filter.

To better understand the action of this filtering operation we associate with each of the vectors $\mathbf{f}$, $\mathbf{g}$ and $\mathbf{h}$ a function of $\omega$: let

$$DFT_{\mathbf{g}}(\omega) = \sum\nolimits_{m=1}^{M} g_m e^{im\omega}$$

for $\omega$ in the interval $[-\pi, \pi]$; similarly define the functions $DFT_{\mathbf{f}}(\omega)$ and $DFT_{\mathbf{h}}(\omega)$. Notice that these functions are the discrete Fourier transforms (DFT) discussed earlier. We have the option here of considering the vector

discrete Fourier transforms instead. However, since we shall also discuss the theoretical case in which we have doubly infinite sequences $\{f_n\}_{n=-\infty}^{\infty}$, it is more convenient to view the DFT as a function of the continuous variable $\omega$ throughout the discussion. As we saw in an earlier exercise, when $\mathbf{f} = \mathbf{g} * \mathbf{h}$ we also have

$$DFT_{\mathbf{f}}(\omega) = DFT_{\mathbf{g}}(\omega)DFT_{\mathbf{h}}(\omega)$$

for the values $\omega = \frac{2\pi}{M}n$, $n = 1, 2, ..., M$.

**Time-invariant linear systems:** Although in practice all digital filtering is performed using finite length vectors, it is convenient, in theoretical discussions, to permit the use of infinite sequences. Suppose now that $g = \{g_n\}_{n=-\infty}^{+\infty}$ and $h = \{h_n\}_{n=-\infty}^{+\infty}$ are infinite sequences of complex numbers. As above, we use $g$ to obtain a convolution filter that, having $h$ as the input, will have as output the convolution of sequences $g$ and $h$. This is the infinite sequence $f = g * h$ with entries

$$f_n = \sum\nolimits_{m=-\infty}^{+\infty} g_m h_{n-m}.$$

This situation is commonly described by saying that the sequence $\{g_n\}$ represents a *time-invariant linear system* in which the input sequence is convolved with $\{g_n\}$ to produce the output sequence.

When dealing with infinite sequences we must be concerned with the convergence of any infinite series we encounter. In Walnut's book [180] and elsewhere an infinite sequence $\{h_n\}$ is called a *signal* if it is *absolutely summable*; that is,

$$\sum_{n=-\infty}^{\infty} |h_n| < +\infty.$$

The sequences $\{g_n\}$ used to define convolution filters are also required to be absolutely summable, so that the output $f = g * h$ is also absolutely summable and $\{f_n\}$ is therefore a signal. However, the requirement that all signals be absolutely summable is a bit restrictive. For that reason most authors, including Walnut, consider wider classes of sequences, such as *absolutely square summable* $h = \{h_n\}$ for which we have

$$\sum_{n=-\infty}^{\infty} |h_n|^2 < +\infty,$$

bounded sequences and sequences obtained from finitely nonzero ones by periodic extension. Concepts such as stability can be defined in different ways, depending on the type of signals being considered. Our discussion here will be more formal and less rigorous. The reader should remember

that integrals and infinite sums make sense only after appropriate assumptions are made.

We associate with doubly infinite sequences a function of $\omega$: for each $\omega$ in the interval $[-\pi, \pi]$ let

$$G(\omega) = \sum_{n=-\infty}^{+\infty} g_n e^{in\omega}. \tag{10.2}$$

Define $F(\omega)$ and $H(\omega)$ similarly. Because the sequences are infinite we have a multiplication theorem that is somewhat stronger than with the vector DFT.

**Exercise 1:** Show that $F(\omega) = G(\omega)H(\omega)$ for all $\omega$ in $[-\pi, \pi]$.

We see from the exercise that the convolution filter obtained from the sequence $\{g_n\}$ can be understood in terms of how it affects the individual complex exponential components that make up the input. The filter converts each $H(\omega)$ into $F(\omega) = G(\omega)H(\omega)$. If $G(\omega) = 0$ for certain values of $\omega$ then whenever $h(t)$ has a complex exponential component corresponding to that value of $\omega$ it will be removed upon filtering.

Convolution filters have the important property that they amplify or depress sinusoidal inputs without distorting the frequency. Let $\omega$ be an arbitrary but fixed frequency in the interval $[-\pi, \pi]$ and let the input to the filter be the doubly infinite sequence $h$ with entries $h_n = e^{-in\omega}$; that is, a pure sinusoid with frequency $-\omega$. Then the output sequence is $f$ with entries

$$f_n = e^{-in\omega} \sum_{m=-\infty}^{\infty} g_m e^{im\omega}.$$

So the output is again a pure sinusoid, with the same frequency as the input, but with amplitude $G(\omega)$ instead of one.

The function $G(\omega)$ in equation (10.2) is a *Fourier series*. Here we began with an essentially arbitrary sequence $g$ of complex numbers and formed the function $G$. In a number of applications we begin with a function $G(\omega)$ that is either defined on an interval of length $2\pi$ or is defined for all $\omega$ and is $2\pi$-periodic. We then seek the complex numbers $g_n$ so that the Fourier series obtained using these $g_n$ gives us back the original function $G$ as in equation (10.2). This is called the *Fourier series expansion* of the function $G(\omega)$.

Given the function $H(\omega)$ on $[-\pi, \pi]$ the numbers $h_n$ can be determined: we have

$$h_n = \int_{-\pi}^{\pi} H(\omega) e^{-in\omega} \frac{d\omega}{2\pi}. \tag{10.3}$$

This follows from the orthogonality of the functions $e^{in\omega}$ over the interval $[-\pi, \pi]$, as we shall discuss in the next chapter. We can interpret equation

(10.3) as expressing the sequence $h = \{h_n\}$ as a continuously infinite superposition of pure sinusoids, each with their own frequency $-\omega$ and amplitude $H(\omega)/2\pi$. We know that the output from the individual sinusoidal input $\{e^{-in\omega}\}$ is $G(\omega)\{e^{-in\omega}\}$. By the linearity of the filter, the output from the input sequence $h$ with entries given by equation (10.3) is therefore the sequence $f$ with entries

$$f_n = \int_{-\pi}^{\pi} G(\omega)H(\omega)e^{-in\omega}\frac{d\omega}{2\pi}.$$

Since we also have

$$f_n = \int_{-\pi}^{\pi} F(\omega)e^{-in\omega}\frac{d\omega}{2\pi},$$

we are led once again to $F(\omega) = G(\omega)H(\omega)$.

Suppose that the input to the filter is an impulsive sequence; that is, let the input be the sequence $h = \delta^0$ with entries $h_n = 0$ for $n \neq 0$ and $h_0 = 1$. Then the output is the sequence $f$ with entries $f_n = g_n$. The sequence $g = \{g_n\}$ used to build the discrete linear filter is therefore called the *impulse response* sequence of the filter and the function $G(\omega)$ is the *filter function*.

**Exercise 2:** The *three-point moving average* filter is defined as follows: given the input sequence $\{h_n, n = -\infty, ..., \infty\}$ the output sequence is $\{f_n, n = -\infty, ..., \infty\}$, with

$$f_n = (h_{n-1} + h_n + h_{n+1})/3.$$

Let $g_m = 1/3$, if $m = 0, 1, -1$ and $g_m = 0$, otherwise. Then we have

$$f_n = \sum_{m=-\infty}^{\infty} g_m h_{n-m},$$

so that $f$ is the convolution of $h$ and $g$. Let $F(\omega)$ be defined for $\omega$ in the interval $[-\pi, \pi]$ by equation (10.2); similarly define $G$ and $H$. To recover $h$ from $f$ we might proceed as follows: calculate $F$, then divide $F$ by $G$ to get $H$, then compute $h$ from $H$; does this always work?
If we let $h$ be the sequence $\{..., 1, 1, 1, ...\}$ then $f = h$; if we take $h$ to be the sequence $\{..., 3, 0, 0, 3, 0, 0, ...\}$ then we again get $f = \{..., 1, 1, 1, ...\}$. Therefore, we cannot expect to recover $h$ from $f$ in general. We know that $G(\omega) = \frac{1}{3}(1 + 2\cos(\omega))$; what does this have to do with the problem of recovering $h$ from $f$?

**Hint:** Compute $H$. Where are the zeros of $G$?

If we take the input sequence to our convolution filter the sequence $h$ with entries

$$h_n = \overline{g}_{-n}$$

then the output sequence is $f$ with entries

$$f_n = \sum\nolimits_{m=-\infty}^{+\infty} g_m \overline{g}_{m-n}$$

and $F(\omega) = |G(\omega)|^2$. The sequence $f$ is called the *autocorrelation sequence* for $g$ and $|G(\omega)|^2$ is the *power spectrum* of $g$. The Cauchy inequality is valid for infinite sequences also: with the length of $f$ defined by

$$||f|| = (\sum\nolimits_{n=-\infty}^{+\infty} |f_n|^2)^{1/2}$$

and the inner product of $f$ and $g$ given by

$$\langle f, g \rangle = \sum\nolimits_{n=-\infty}^{+\infty} f_n \overline{g}_n$$

we have

$$|\langle f, g \rangle| \leq ||f||\,||g||,$$

with equality if and only if $g$ is a constant multiple of $f$.

**Exercise 3:** Let $f$ be the autocorrelation sequence for $g$. Show that $f_{-n} = \overline{f}_n$ and $f_0 \geq |f_n|$ for all $n$.

**The z-transform:** It is common to consider the case in which the input to a time-invariant linear system $g = \{g_n\}$ is a discrete random process $\{X_n\}$; that is, each $X_n$ is a random variable [152], [158]. The output sequence $\{Y_n\}$ given by

$$Y_n = \sum_{m=-\infty}^{+\infty} g_m X_{n-m}$$

is then a second discrete random process whose statistics are related to those of the input, as well as to properties of the sequence $g$. By analogy with what we did earlier, we would like to be able to form the functions

$$X(\omega) = \sum_{n=-\infty}^{+\infty} X_n e^{in\omega}$$

and

$$Y(\omega) = \sum_{n=-\infty}^{+\infty} Y_n e^{in\omega}$$

and use them to study the action of the system on random input. For the series for $X(\omega)$ to converge we would at least want

$$\sum_{n=-\infty}^{+\infty} |X_n|^2 < +\infty.$$

This poses a problem, because the random processes $\{X_n\}$ we usually consider do not go to zero as $|n| \to +\infty$. For this reason we need a somewhat more general tool, the z-transform.

Given a doubly infinite sequence sequence $g = \{g_n\}_{n=-\infty}^{+\infty}$ we associate with $g$ its *z-transform*, the function of the complex variable $z$ given by

$$G(z) = \sum\nolimits_{n=-\infty}^{+\infty} g_n z^{-n}.$$

Doubly infinite series of this form are called *Laurent series* and occur in the representation of functions analytic in an annulus. Note that if we take $z = e^{-i\omega}$ then $G(z)$ becomes $G(\omega)$ as defined by equation (10.2). The $z$-transform is a somewhat more flexible tool in that we are not restricted to those sequence $g$ for which the $z$-transform is defined for $z = e^{-i\omega}$.

The linear system determined by $g$ is said to be *stable* [150] if the output sequence is bounded in absolute value whenever the input sequence is.

**Exercise 4:** Show that the linear system determined by $g$ is stable if and only if $\sum_{n=-\infty}^{+\infty} |g_n| < +\infty$.

**Hint:** If $\sum_{n=-\infty}^{+\infty} |g_n| = +\infty$, consider as input the bounded sequence $f_n = \overline{g_{-n}}/|g_n|$ and show that $h_0 = +\infty$.

**Exercise 5:** Consider the linear system determined by the sequence $g_0 = 2$, $g_n = (\frac{1}{2})^{|n|}$, for $n \neq 0$. Show that this system is stable. Calculate the z-transform of $\{g_n\}$ and determine its region of convergence.

The time-invariant linear system determined by $g$ is said to be a *causal system* if the sequence $\{g_n\}$ is itself causal; that is, $g_n = 0$ for $n < 0$.

**Exercise 6:** Show that the function $G(z) = (z - z_0)^{-1}$ is the z-transform of a causal sequence $g$, where $z_0$ is a fixed complex number. What is the region of convergence? Show that the resulting linear system is stable if and only if $|z_0| < 1$.

**Continuous time-invariant linear systems:** An *operator* $T$ associates with function $f$ another function $Tf$. For example, $Tf$ could be the derivative of $f$, if $f$ is differentiable, or $Tf$ could be $F$, the Fourier transform of $f$. The operator $T$ is called *linear* if $T(f + h) = Tf + Th$ and

$T(\alpha f) = \alpha T f$ for any functions $f$ and $h$ and scalar $\alpha$. For any real number $\tau$ let $f_\tau(t) = f(t+\tau)$. We say that $T$ is *time-invariant* if $h = Tf$ implies that $h_\tau = Tf_\tau$. Suppose we fix a function $g$ and define $Tf = f * g$; such an operator is called a *convolution operator*. Convolution operators are linear and time-invariant. As we shall see, time-invariant linear systems are convolution operators.

**Exercise 7:** Let $f(t) = e^{-i\omega t}$ for some fixed real number $\omega$. Let $h = Tf$, where $T$ is linear and time-invariant. Show that there is a constant $c$ so that $h(t) = cf(t)$. Since the constant $c$ may depend on $\omega$ we rewrite $c$ as $G(\omega)$.

**Exercise 8:** Let $T$ be as in the previous exercise. For

$$f(t) = \int_{-\infty}^{+\infty} F(\omega) e^{-i\omega t} d\omega / 2\pi$$

and $h = Tf$ show that $H(\omega) = F(\omega)G(\omega)$ for each $\omega$. Conclude that $T$ is a convolution operator whose function $g(t)$ is the inverse FT of $G(\omega)$.

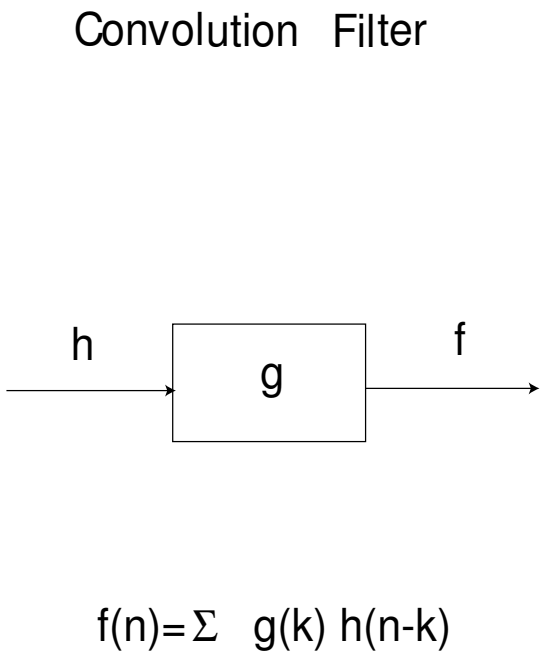



Figure 10.1: Convolution filter $g$ operating on input $h$ to produce out put $f$.

# Chapter 11

# Inner Products

The proof of Cauchy's inequality rests not on the actual definition of the complex vector dot product, but rather on four of its most basic properties. We use these properties to extend the concept of complex vector dot product to that of *inner product*. Later in this chapter we shall give several examples of inner products, applied to a variety of mathematical objects, including infinite sequences, functions, random variables and matrices. For now, let us denote our mathematical objects by $\mathbf{u}$ and $\mathbf{v}$ and the inner product between them as $\langle \mathbf{u}, \mathbf{v} \rangle$. The objects will then be said to be members of an *inner product space*. We are interested in inner products because they provide a notion of orthogonality, which is fundamental to best approximation and optimal estimation.

**Defining an inner product:** The four basic properties that will serve to define an inner product are as follows:

**1:** $\langle \mathbf{u}, \mathbf{u} \rangle \geq 0$, with equality if and only if $\mathbf{u} = \mathbf{0}$;

**2.** $\langle \mathbf{v}, \mathbf{u} \rangle = \overline{\langle \mathbf{u}, \mathbf{v} \rangle}$;

**3.** $\langle \mathbf{u}, \mathbf{v} + \mathbf{w} \rangle = \langle \mathbf{u}, \mathbf{v} \rangle + \langle \mathbf{u}, \mathbf{w} \rangle$;

**4.** $\langle c\mathbf{u}, \mathbf{v} \rangle = c\langle \mathbf{u}, \mathbf{v} \rangle$ for any complex number $c$.

The inner product is the basic ingredient in Hilbert space theory. Using the inner product, we define the *norm* of $\mathbf{u}$ to be

$$||\mathbf{u}|| = \sqrt{\langle \mathbf{u}, \mathbf{u} \rangle}$$

and the distance between $\mathbf{u}$ and $\mathbf{v}$ to be $||\mathbf{u} - \mathbf{v}||$.

**The Cauchy-Schwarz inequality:** Because these four properties were all we needed to prove the Cauchy inequality for the complex vector dot product, we obtain the same inequality whenever we have an inner product. This more general inequality is the Cauchy-Schwarz inequality:

$$|\langle \mathbf{u}, \mathbf{v} \rangle| \leq \sqrt{\langle \mathbf{u}, \mathbf{u} \rangle} \sqrt{\langle \mathbf{v}, \mathbf{v} \rangle}$$

or

$$|\langle \mathbf{u}, \mathbf{v} \rangle| \leq ||\mathbf{u}|| \, ||\mathbf{v}||,$$

with equality if and only if there is a scalar $c$ such that $\mathbf{v} = c\mathbf{u}$. We say that the vectors $\mathbf{u}$ and $\mathbf{v}$ are *orthogonal* if $\langle \mathbf{u}, \mathbf{v} \rangle = 0$. We turn now to some examples.

**Inner products of infinite sequences:** Let $\mathbf{u} = \{u_n\}$ and $\mathbf{v} = \{v_n\}$ be infinite sequences of complex numbers. The inner product is then

$$\langle \mathbf{u}, \mathbf{v} \rangle = \sum u_n \overline{v_n},$$

and

$$||\mathbf{u}|| = \sqrt{\sum |u_n|^2}.$$

The sums are assumed to be finite; the index of summation $n$ is singly or doubly infinite, depending on the context. The Cauchy-Schwarz inequality says that

$$|\sum u_n \overline{v_n}| \leq \sqrt{\sum |u_n|^2} \sqrt{\sum |v_n|^2}.$$

**Inner product of functions:** Now suppose that $\mathbf{u} = f(x)$ and $\mathbf{v} = g(x)$. Then

$$\langle \mathbf{u}, \mathbf{v} \rangle = \int f(x) \overline{g(x)} dx$$

and

$$||\mathbf{u}|| = \sqrt{\int |f(x)|^2 dx}.$$

The integrals are assumed to be finite; the limits of integration depend on the support of the functions involved. The Cauchy-Schwarz inequality now says that

$$|\int f(x) \overline{g(x)} dx| \leq \sqrt{\int |f(x)|^2 dx} \sqrt{\int |g(x)|^2 dx}.$$

**Inner product of random variables:** Now suppose that $\mathbf{u} = X$ and $\mathbf{v} = Y$ are random variables. Then

$$\langle \mathbf{u}, \mathbf{v} \rangle = E(X\overline{Y})$$

and

$$||\mathbf{u}|| = \sqrt{E(|X|^2)},$$

which is the standard deviation of $X$ if the mean of $X$ is zero. The expected values are assumed to be finite. The Cauchy-Schwarz inequality now says that

$$|E(X\overline{Y})| \leq \sqrt{E(|X|^2)}\sqrt{E(|Y|^2)}.$$

If $E(X) = 0$ and $E(Y) = 0$ the random variables $X$ and $Y$ are orthogonal if and only if they are *uncorrelated*.

**Inner product of complex matrices:** Now suppose that $\mathbf{u} = A$ and $\mathbf{v} = B$ are complex matrices. Then

$$\langle \mathbf{u}, \mathbf{v} \rangle = \text{trace}(B^\dagger A)$$

and

$$||\mathbf{u}|| = \sqrt{\text{trace}(A^\dagger A)},$$

where the trace of a square matrix is the sum of the entries on the main diagonal. As we shall see later, this inner product is simply the complex vector dot product of the vectorized versions of the matrices involved. The Cauchy-Schwarz inequality now says that

$$|\text{trace}(B^\dagger A)| \leq \sqrt{\text{trace}(A^\dagger A)}\sqrt{\text{trace}(B^\dagger B)}.$$

**Weighted inner products of complex vectors:** Let $\mathbf{u}$ and $\mathbf{v}$ be complex vectors and let $Q$ be a Hermitian positive-definite matrix; that is, $Q^\dagger = Q$ and $\mathbf{u}^\dagger Q\mathbf{u} > 0$ for all nonzero vectors $\mathbf{u}$ .The inner product is then

$$\langle \mathbf{u}, \mathbf{v} \rangle = \mathbf{v}^\dagger Q\mathbf{u}$$

and

$$||\mathbf{u}|| = \sqrt{\mathbf{u}^\dagger Q\mathbf{u}}.$$

We know from the eigenvector decomposition of $Q$ that $Q = C^\dagger C$ for some matrix $C$. Therefore the inner product is simply the complex vector dot product of the vectors $C\mathbf{u}$ and $C\mathbf{v}$. The Cauchy-Schwarz inequality says that

$$|\mathbf{v}^\dagger Q\mathbf{u}| \leq \sqrt{\mathbf{u}^\dagger Q\mathbf{u}}\sqrt{\mathbf{v}^\dagger Q\mathbf{v}}.$$

**The weighted inner product of functions:** Now suppose that $\mathbf{u} = f(x)$ and $\mathbf{v} = g(x)$ and $w(x) > 0$. Then define

$$\langle \mathbf{u}, \mathbf{v} \rangle = \int f(x)\overline{g(x)}w(x)dx$$

and

$$||\mathbf{u}|| = \sqrt{\int |f(x)|^2 w(x)dx}.$$

The integrals are assumed to be finite; the limits of integration depend on the support of the functions involved. This inner product is simply the inner product of the functions $f(x)\sqrt{w(x)}$ and $g(x)\sqrt{w(x)}$. The Cauchy-Schwarz inequality now says that

$$|\int f(x)\overline{g(x)}w(x)dx| \leq \sqrt{\int |f(x)|^2 w(x)dx}\sqrt{\int |g(x)|^2 w(x)dx}.$$

Once we have an inner product defined we can speak about orthogonality and best approximation. Important in that regard is the orthogonality principle, the topic of the next chapter.

# Chapter 12

# The Orthogonality Principle

Imagine that you are standing and looking down at the floor. The point $B$ on the floor that is closest to $N$, the tip of your nose, is the unique point on the floor such that the vector from $B$ to any other point $A$ on the floor is perpendicular to the vector from $N$ to $B$; that is, $\langle BN, BA\rangle = 0$. This is a simple illustration of the *orthogonality principle*. Whenever we have an inner product defined we can speak of orthogonality and apply the orthogonality principle to find best approximations.

**The orthogonality principle:** Let $\mathbf{u}$ and $\mathbf{v}^1, ..., \mathbf{v}^N$ be members of an inner product space. For all choices of scalars $a_1, ..., a_N$ we can compute the distance from $\mathbf{u}$ to the member $a_1\mathbf{v}^1 + ...a_N\mathbf{v}^N$. Then we minimize this distance over all choices of the scalars; let $b_1, ..., b_N$ be this best choice. The *orthogonality principle* tells us that the member $\mathbf{u} - (b_1\mathbf{v}^1 + ...b_N\mathbf{v}^N)$ is orthogonal to the member $(a_1\mathbf{v}^1 + ... + a_N\mathbf{v}^N) - (b_1\mathbf{v}^1 + ...b_N\mathbf{v}^N)$, that is,

$$\langle \mathbf{u} - (b_1\mathbf{v}^1 + ...b_N\mathbf{v}^N), (a_1\mathbf{v}^1 + ... + a_N\mathbf{v}^N) - (b_1\mathbf{v}^1 + ...b_N\mathbf{v}^N) = 0,$$

for every choice of scalars $a_n$. We can then use the orthogonality principle to find the best choice $b_1., , , .b_N$.

For each fixed index value $j$ in the set $\{1, ..., N\}$ let $a_n = b_n$ if $j$ is not equal to $n$ and $a_j = b_j + 1$. Then we have

$$0 = \langle \mathbf{u} - (b_1\mathbf{v}^1 + ...b_N\mathbf{v}^N), \mathbf{v}^j\rangle,$$

or

$$\langle \mathbf{u}, \mathbf{v}^j\rangle = \sum\nolimits_{n=1}^{N} b_n\langle \mathbf{v}^n, \mathbf{v}^j\rangle,$$

for each $j$. The $\mathbf{v}^n$ are known, so we can calculate the inner products $\langle \mathbf{v}^n, \mathbf{v}^j \rangle$ and solve this system of equations for the best $b_n$.

We shall encounter a number of particular cases of the orthogonality principle in subsequent chapters. The example of the *least squares* solution of a system of linear equations provides a good example of the use of this principle.

**The least squares solution:** Let $V\mathbf{a} = \mathbf{u}$ be a system of $M$ linear equations in $N$ unknowns. For $n = 1, ..., N$ let $\mathbf{v}^n$ be the $n$-th column of the matrix $V$ . For any choice of the vector $\mathbf{a}$ with entries $a_n$, $n = 1, ..., N$ the vector $V\mathbf{a}$ is

$$V\mathbf{a} = \sum\nolimits_{n=1}^{N} a_n \mathbf{v}^n.$$

Solving $V\mathbf{a} = \mathbf{u}$ amounts to representing the vector $\mathbf{u}$ as a linear combination of the columns of $V$ .

If there is no solution of $V\mathbf{a} = \mathbf{u}$ then we can look for the best choice of coefficients so as to minimize the distance $||\mathbf{u} - (a_1\mathbf{v}^1 + ... + a_N\mathbf{v}^N)||$. The matrix with entries $\langle \mathbf{v}^n, \mathbf{v}^j \rangle$ is $V^\dagger V$ and the vector with entries $\langle \mathbf{u}, \mathbf{v}^j \rangle$ is $V^\dagger \mathbf{u}$. According to the orthogonality principle we must solve the system of equations $V^\dagger \mathbf{u} = V^\dagger V\mathbf{a}$, which leads to the least squares solution.

**Exercise 1:** Find polynomial functions $f(x)$, $g(x)$ and $h(x)$ that are orthogonal on the interval $[0, 1]$ and have the property that every polynomial of degree two or less can be written as a linear combination of these three functions.

**Exercise 2:** Show that the functions $e^{inx}$, $n$ an integer, are orthogonal on the interval $[-\pi, \pi]$. Let $f(x)$ have the Fourier expansion

$$f(x) = \sum\nolimits_{n=-\infty}^{\infty} a_n e^{inx},\ |x| \leq \pi.$$

Use orthogonality to find the coefficients $a_n$.

We have seen that orthogonality can be used to determine the coefficients in the Fourier series representation of a function. There are other useful representations in which orthogonality also plays a role; wavelets is one such. Let $f(x)$ be defined on the closed interval $[0, X]$. Suppose that we change the function $f(x)$ to a new function $g(x)$ by altering the values for $x$ within a small interval, keeping the remaining values the same: then all of the Fourier coefficients change. Looked at another way, a localized disturbance in the function $f(x)$ affects all of its Fourier coefficients. It would be helpful to be able to represent $f(x)$ as a sum of orthogonal functions in such a way that localized changes in $f(x)$ affect only a small number of the components in the sum. One way to do this is with wavelets, as we shall see shortly.

# Chapter 13

# Fourier Transforms and Fourier Series

In a previous chapter we studied the problem of isolating the individual complex exponential components of the signal function $s(t)$, given the data vector $\mathbf{d}$ with entries $s(m\Delta)$, $m = 1, ..., M$, where $s(t)$ is

$$s(t) = \sum\nolimits_{n=1}^{N} A_n e^{i\omega_n t};$$

we assume that $|\omega_n| < \pi/\Delta$. The second approach we considered involved calculating the function

$$DFT_{\mathbf{d}}(\omega) = \sum\nolimits_{m=1}^{M} s(m\Delta)e^{-i\omega m\Delta}$$

for $|\omega| < \pi/\Delta$. This sum is an example of a (finite) Fourier series. As we just saw, we can extend the concept of Fourier series to include infinite sums. In fact, we can generalize to summing over a continuous variable, using integrals in place of summation; this is what is done in the definition of the Fourier transform.

### The Fourier transform:

In our discussion of linear filtering we saw that if $f$ is a finite vector $\mathbf{f} = (f_1, ..., f_M)^T$ or an infinite sequence $f = \{f_m\}_{m=-\infty}^{+\infty}$ then it is convenient to consider the function $F(\omega)$ defined for $|\omega| \leq \pi$ by the finite or infinite Fourier series expression

$$F(\omega) = \sum f_m e^{im\omega}.$$

If $f(x)$ is a function of the real variable $x$, we can associate with $f$ the function $F(\omega)$, the *Fourier transform* (FT) of $f(x)$, defined for all real $\omega$

by

$$F(\omega) = \int f(x) e^{ix\omega} dx. \tag{13.1}$$

Once we have $F(\omega)$ we can recover $f(x)$ as the *inverse Fourier transform* (IFT) of $F(\omega)$:

$$f(x) = \int F(\omega) e^{-ix\omega} d\omega / 2\pi. \tag{13.2}$$

We say then that the functions $f$ and $F$ form a Fourier transform pair. It may happen that one or both of the integrals above will fail to be defined in the usual way and will be interpreted as the principal value of the integral [97].

Note that the definitions of the FT and IFT just given may differ slightly from the ones found elsewhere; our definitions are those of Bochner and Chandrasekharan [18]. The differences are minor and involve only the placement of the quantity $2\pi$ and of the minus sign in the exponent. One sometimes sees the FT of the function $f$ denoted $\hat{f}$; here we shall reserve the symbol $\hat{f}$ for estimates of the function $f$.

As an example of a Fourier transform pair let $F(\omega)$ be the function $\chi_\Omega(\omega)$ that equals one for $|\omega| \leq \Omega$ and is zero otherwise. Then the inverse Fourier transform of $\chi_\Omega(\omega)$ is

$$f(x) = \int_{-\Omega}^{\Omega} e^{-i\omega x} d\omega / 2\pi = \frac{\sin(\Omega x)}{\pi x}.$$

The function $\frac{\sin(x)}{x}$ is called the *sinc* function, $\mathrm{sinc}\,(x)$.

## Fourier series:

If there is a positive $\Omega$ such that the Fourier transform $F(\omega)$ of the function $f(x)$ is zero for $|\omega| > \Omega$ then the function $f(x)$ is said to be $\Omega$-*bandlimited* and $F(\omega)$ has *bandwidth* $\Omega$; in this case the function $F(\omega)$ can be written, on the interval $[-\Omega, \Omega]$, as an infinite discrete sum of complex exponentials. For $|\omega| \leq \Omega$ we have

$$F(\omega) = \sum\nolimits_{n=-\infty}^{+\infty} f_n e^{in\omega \frac{\pi}{\Omega}}. \tag{13.3}$$

We determine the coefficients $f_n$ in much the same way as in earlier discussions.

We know that the integral

$$\int_{-\Omega}^{\Omega} e^{i(n-m)\omega \frac{\pi}{\Omega}} d\omega$$

equals zero if $m \neq n$ and equals $2\Omega$ for $m = n$. Therefore,

$$f_m = \frac{1}{2\Omega} \int_{-\Omega}^{\Omega} F(\omega) e^{-im\omega \frac{\pi}{\Omega}} d\omega \tag{13.4}$$

for each integer $m$. If we wish, we can also write the coefficient $f_m$ in terms of the inverse Fourier transform $f(x)$ of the function $F(\omega)$: the right side of equation (13.4) also equals $\frac{\pi}{\Omega} f(m\frac{\pi}{\Omega})$, from which we conclude that $f_m = \frac{\pi}{\Omega} f(m\frac{\pi}{\Omega})$.

**The Shannon Sampling Theorem:** Now that we have found the coefficients of the Fourier series for $F(\omega)$ we can write

$$F(\omega) = \frac{\pi}{\Omega} \sum_{n=-\infty}^{\infty} f(n\frac{\pi}{\Omega}) e^{in\omega \frac{\pi}{\Omega}} \tag{13.5}$$

for $|\omega| \leq \Omega$. We apply the formula in equation (13.2) to get

$$f(x) = \sum_{n=-\infty}^{\infty} f(n\frac{\pi}{\Omega}) \frac{\sin(\Omega x - n\pi)}{\Omega x - n\pi}. \tag{13.6}$$

This is the famous *Shannon sampling theorem*, which tells us that if $F(\omega)$ is zero outside $[-\Omega, \Omega]$, then $f(x)$ is completely determined by the infinite sequence of values $\{f(n\frac{\pi}{\Omega})\}_{n=-\infty}^{+\infty}$. If $F(\omega)$ is continuous and $F(-\Omega) = F(\Omega)$ then $F(\omega)$ has a continuous periodic extension to all of the real line. Then the Fourier series in equation (13.3) converges to $F(\omega)$ for every $\omega$ at which the function $F(\omega)$ has a left and right derivative. In general, if $F(-\Omega) \neq F(\Omega)$, or if $F(\omega)$ is discontinuous for some $\omega$ in $(-\Omega, \Omega)$, the series will still converge, but to the average of the one-sided limits $F(\omega + 0)$ and $F(\omega - 0)$, again, provided that $F(\omega)$ has one-sided derivatives at that point. If

$$\int_{-\Omega}^{\Omega} |F(\omega)|^2 d\omega < \infty$$

then

$$\sum_{n=-\infty}^{+\infty} |f(n\frac{\pi}{\Omega})|^2 < \infty$$

and the series in equation (13.6) converges to $f(x)$ in the $L^2$ sense. If, in addition, we have

$$\sum_{n=-\infty}^{+\infty} |f(n\frac{\pi}{\Omega})| < \infty,$$

then the series converges uniformly to $f(x)$ for $x$ on the real line. There are many books that can be consulted for details concerning convergence of Fourier series, such as [16] and [97].

Let $f = \{f_m\}$ and $g = \{g_m\}$ be the sequences of Fourier coeffcients for the functions $F(\omega)$ and $G(\omega)$, respectively, defined on the interval $[-\pi, \pi]$; that is

$$F(\omega) = \sum_{m=-\infty}^{\infty} f_m e^{im\omega}, |\omega| \leq \pi.$$

**Exercise 1:** Use the orthogonality of the functions $e^{im\omega}$ on $[-\pi, \pi]$ to establish *Parseval's equation*:

$$\langle f, g \rangle = \sum_{m=-\infty}^{\infty} f_m \overline{g_m} = \int_{-\pi}^{\pi} F(\omega)\overline{G(\omega)} d\omega/2\pi,$$

from which it follows that

$$\langle f, f \rangle = \int_{-\infty}^{\infty} |F(\omega)|^2 d\omega/2\pi.$$

Similar results hold for the Fourier transform, as we shall see in the next chapter.

**Exercise 2:** Let $f(x)$ be defined for all real $x$ and let $F(\omega)$ be its FT. Let

$$g(x) = \sum_{k=-\infty}^{\infty} f(x + 2\pi k),$$

assuming the sum exists. Show that $g$ is a $2\pi$ -periodic function. Compute its Fourier series and use it to derive the *Poisson summation formula*:

$$\sum_{k=-\infty}^{\infty} f(2\pi k) = \frac{1}{2\pi} \sum_{n=-\infty}^{\infty} F(n).$$

In certain applications our main interest is the function $f(x)$, for which we have finitely many (usually noisy) values. For example, $x$ may be the time variable $t$ and $f(t)$ may be a short segment of spoken speech that we wish to analyze. We model $f(t)$ as a finite, infinite discrete or continuous sum of complex exponentials, that is, as a Fourier series or Fourier transform, in order to process the data, to remove the noise, to compress the data and to identify the parameters.

In remote sensing applications (such as radar, sonar, tomography), on the other hand, we have again noisy values of $f(x)$, but it is not $f(x)$ that interests us. Instead, we are interested in $F(\omega)$, the Fourier transform of $f(x)$ or the sequence $F_n$ of the complex Fourier coeffcients of $f(x)$, if $f(x) = 0$ outside some finite interval. We cannot measure these quantities directly, so we must content ourselves with estimating them from our measurements of $f(x)$.

In yet a third class of applications, such as linear filtering, we are concerned with constructing a digital procedure for performing certain operations on any signal we might receive as input. In such cases our goal is to construct the sequence $g_n$ for which the associated Fourier series $G(\omega)$ will have a desired shape. For example, we may want the filter to eliminate all complex exponential components of the input signal whose frequency is not in the interval $[-\Omega, \Omega]$. Then we would want $G(\omega)$ to be one for $\omega$ within this interval and zero outside. To achieve this we would take the sequence $g_n$ to be

$$g_n = \frac{\sin(\Omega n)}{\pi n}.$$

In these applications there is no $f(x)$ to be analyzed nor $F(\omega)$ to be estimated.

# Chapter 14

# Fourier Series and Analytic Functions

We first encounter infinite series expansions for functions in calculus, when we study Maclaurin and Taylor series. Fourier series are usually first met in a much different context, such as partial differential equations and boundary value problems. Laurent expansions come later, when we study functions of a complex variable. There are, nevertheless, important connections among these different types of infinite series expansions, which provide the subject for this chapter.

Suppose that $f(z)$ is analytic in an annulus containing the unit circle $C = \{z \,|\, |z| = 1\}$. Then $f(z)$ has a Laurent series expansion

$$f(z) = \sum_{n=-\infty}^{\infty} f_n z^n$$

valid for $z$ within that annulus. Substituting $z = e^{i\theta}$ we get $f(\theta)$, defined for $\theta$ in the interval $[-\pi, \pi]$ by

$$f(\theta) = f(e^{i\theta}) = \sum_{n=-\infty}^{\infty} f_n e^{in\theta};$$

here the Fourier series for $f(\theta)$ is derived from the Laurent series for the analytic function $f(z)$. If $f(z)$ is actually analytic in $(1+\epsilon)D$, where $D = \{z \,|\, |z| < 1\}$ is the open unit disk, then $f(z)$ has a Taylor series expansion and the Fourier series for $f(\theta)$ contains only terms corresponding to nonnegative $n$.

As an example, consider the rational function

$$f(z) = \frac{1}{z - \frac{1}{2}} - \frac{1}{z - 3} = -\frac{5}{2}/(z - \frac{1}{2})(z - 3).$$

In an annulus containing the unit circle this function has the Laurent series expansion

$$f(z) = \sum_{n=-\infty}^{-1} 2^{n+1} z^n + \sum_{n=0}^{\infty} (\frac{1}{3})^{n+1} z^n;$$

replacing $z$ with $e^{i\theta}$ we obtain the Fourier series for the function $f(\theta) = f(e^{i\theta})$ defined for $\theta$ in the interval $[-\pi, \pi]$.

The function $F(z) = 1/f(z)$ is analytic for all complex $z$, but because it has a root inside the unit circle, its reciprocal, $f(z)$, is not analytic in a disk containing the unit circle. Consequently, the Fourier series for $f(\theta)$ is doubly infinite. We saw in the chapter on complex varables that the function $G(z) = \frac{z-\overline{a}}{1-az}$ has $|G(e^{i\theta})| = 1$. With $a = 2$ and $H(z) = F(z)G(z)$ we have

$$H(z) = \frac{1}{5}(z-3)(z-2)$$

and its reciprocal has the form

$$1/H(z) = \sum_{n=0}^{\infty} a_n z^n.$$

Because

$$G(e^{i\theta})/H(e^{i\theta}) = 1/F(e^{i\theta})$$

it follows that

$$|1/H(e^{i\theta})| = |1/F(e^{i\theta})| = |f(\theta)|$$

and so

$$|f(\theta)| = |\sum_{n=0}^{\infty} a_n e^{in\theta}|.$$

Multiplication by $G(z)$ permits us to move a root from inside $C$ to outside $C$ without altering the magnitude of the function's values on $C$.

The relationships that obtain between functions defined on $C$ and functions analytic (or harmonic) in $D$ form the core of *harmonic analysis* [114]. The factorization $F(z) = H(z)/G(z)$ above is a special case of the *inner-outer factorization* for functions in Hardy spaces; the function $H(z)$ is an *outer function* and the functions $G(z)$ and $1/G(z)$ are *inner functions*.

Instead of starting with an analytic function and restricting it to the unit circle, we often begin with a function $f(e^{i\theta})$ defined on the unit circle, or, equivalently, a function of the form $f(\theta)$ for $\theta$ in $[-\pi, \pi]$, and wish to view this function as the restriction to the unit circle of a function that is analytic in a region containing the unit circle. One application of this idea is the Fejér-Riesz factorization theorem.

**Theorem 14.1** *Let $h(\theta)$ be a finite trigonometric polynomial*

$$h(\theta) = \sum_{n=-N}^{N} h_n e^{in\theta}$$

*such that $h(\theta) \geq 0$ for all $\theta$ in the interval $[-\pi, \pi]$. Then there is*

$$y(\theta) = \sum_{n=0}^{N} y_n e^{in\theta}$$

*with $h(\theta) = |y(\theta)|^2$. The function $y(z)$ is unique if we require, in addition, that all its roots be outside $D$.*

To prove this theorem we consider the function

$$h(z) = \sum_{n=-N}^{N} h_n z^n,$$

which is analytic in an annulus containing the unit circle, with $h(e^{i\theta}) = h(\theta)$. The rest of the proof is contained in the following exercise.

**Exercise 1:** Use the fact that $h_{-n} = \overline{h}_n$ to show that $z_j$ is a root of $h(z)$ if and only if $1/\overline{z}_j$ is also a root. From the nonnegativity of $h(e^{i\theta})$ conclude that if $h(z)$ has a root on the unit circle then it has even multiplicity. Take $y(z)$ to be proportional to the product of factors $z - z_j$ for all the $z_j$ outside $D$; for roots on $C$ include them with half their multiplicities.

The Fejér-Riesz theorem is used in the derivation of Burg's maximum entropy method for spectrum estimation. The problem there is to estimate a function $R(\theta) > 0$ knowing only the values

$$r_n = \frac{1}{2\pi} \int_{-\pi}^{\pi} R(\theta) e^{-in\theta} d\theta,$$

for $|n| \leq N$. The approach is to estimate $R(\theta)$ by the function $S(\theta) > 0$ that maximizes the so-called Burg entropy, $\int_{-\pi}^{\pi} \log S(\theta) d\theta$, subject to the data constraints.

The Euler-Lagrange equation from the calculus of variations allows us to conclude that $S(\theta)$ has the form

$$S(\theta) = 1/\sum_{n=-N}^{N} h_n e^{in\theta}.$$

The function

$$h(\theta) = \sum_{n=-N}^{N} h_n e^{in\theta}$$

is nonnegative, so, by the Fejér-Riesz theorem, it factors as $h(\theta) = |y(\theta)|^2$. We then have $S(\theta)\overline{y(\theta)} = 1/y(\theta)$. Since all the roots of $y(z)$ lie outside $D$ and none are on $C$, the function $1/y(z)$ is analytic in a region containing $C$ and $D$ so it has a Taylor series expansion in that region. Restricting this Taylor series to $C$ we obtain a one-sided Fourier series having zero terms for the negative indices.

**Exercise 2:** Show that the coefficients $y_n$ in $y(z)$ satisfy a system of linear equations whose coefficients are the $r_n$.

**Hint:** Compare the coefficients of the terms on both sides of the equation $S(\theta)\overline{y}(\theta) = 1/y(\theta)$ that correspond to negative indices.

**The Hilbert transform for sequences:** If $g(\omega)$ has the Fourier series expansion

$$g(\omega) = \sum_{n=-\infty}^{\infty} g_n e^{-in\omega},$$

the *conjugate Fourier series* [125] is

$$h(\omega) = \sum_{n=-\infty}^{\infty} (-i\,\mathrm{sgn}(n)) g_n e^{-in\omega}.$$

Then

$$f(\omega) = g(\omega) + ih(\omega) = g_0 + 2\sum_{n=1}^{\infty} g_n e^{in\omega}$$

is a one-sided Fourier series. In harmonic analysis the sequence $\{h_n\}$ is said to be the *conjugate* of the sequence $\{g_n\}$; in signal processing it is called its *Hilbert transform*. As we shall see in a subsequent chapter, the Hilbert transform occurs in several different contexts.

# Chapter 15

# More on the Fourier Transform

We begin with exercises that treat basic properties of the FT and then introduce several examples of Fourier transform pairs.

**Exercise 1:** Let $F(\omega)$ be the FT of the function $f(x)$. Use the definitions of the FT and IFT given in equations (13.1) and (13.2) to establish the following basic properties of the Fourier transform operation:

**Symmetry:** The FT of the function $F(x)$ is $2\pi f(-\omega)$. For example, the FT of the function $f(x) = \frac{\sin(\Omega x)}{\pi x}$ is $\chi_\Omega(\omega)$, so the FT of $g(x) = \chi_\Omega(x)$ is $G(\omega) = 2\pi\frac{\sin(\Omega\omega)}{\pi\omega}$.

**Conjugation:** The FT of $\overline{f(x)}$ is $\overline{F(-\omega)}$.

**Scaling:** The FT of $f(ax)$ is $\frac{1}{|a|}F(\frac{\omega}{a})$ for any nonzero constant $a$.

**Shifting:** The FT of $f(x-a)$ is $e^{-ia\omega}F(\omega)$.

**Modulation:** The FT of $f(x)\cos(\omega_0 x)$ is $\frac{1}{2}[F(\omega+\omega_0) + F(\omega-\omega_0)]$.

**Differentiation:** The FT of the $n$-th derivative, $f^{(n)}(x)$ is $(-i\omega)^n F(\omega)$. The IFT of $F^{(n)}(\omega)$ is $(ix)^n f(x)$.

**Convolution in $x$:** Let $f, F$, $g, G$ and $h, H$ be FT pairs, with

$$h(x) = \int f(y)g(x-y)dy,$$

so that $h(x) = (f * g)(x)$ is the convolution of $f(x)$ and $g(x)$. Then $H(\omega) = F(\omega)G(\omega)$. For example, if we take $g(x) = \overline{f(-x)}$, then

$$h(x) = \int f(x+y)\overline{f(y)}dy = \int f(y)\overline{f(y-x)}dy = r_f(x)$$

is the *autocorrelation function* associated with $f(x)$ and

$$H(\omega) = |F(\omega)|^2 = R_f(\omega) \geq 0$$

is the *power spectrum* of $f(x)$.

**Convolution in $\omega$:** Let $f, F$, $g, G$ and $h, H$ be FT pairs, with $h(x) = f(x)g(x)$. Then $H(\omega) = \frac{1}{2\pi}(F * G)(\omega)$.

**Exercise 2:** Show that the Fourier transform of $f(x) = e^{-\alpha^2 x^2}$ is $F(\omega) = \frac{\sqrt{\pi}}{\alpha}e^{-(\frac{\omega}{2\alpha})^2}$. Hint: Calculate the derivative $F'(\omega)$ by differentiating under the integral sign in the definition of $F$ and integrating by parts. Then solve the resulting differential equation.

Let $u(x)$ be the *Heaviside function* that is $+1$ if $x \geq 0$ and 0 otherwise. Let $\chi_X(x)$ be the *characteristic function* of the interval $[-X, X]$ that is $+1$ for $x$ in $[-X, X]$ and 0 otherwise. Let $\text{sgn}(x)$ be the *sign function* that is $+1$ if $x > 0$, $-1$ if $x < 0$ and zero for $x = 0$.

**Exercise 3:** Show that the FT of the function $f(x) = u(x)e^{-ax}$ is $F(\omega) = \frac{1}{a - i\omega}$, for every positive constant $a$.

**Exercise 4:** Show that the FT of $f(x) = \chi_X(x)$ is $F(\omega) = 2\frac{\sin(X\omega)}{\omega}$.

**Exercise 5:** Show that the IFT of the function $F(\omega) = 2i/\omega$ is $f(x) = \text{sgn}(x)$.

**Hints:** write the formula for the inverse Fourier transform of $F(\omega)$ as

$$f(x) = \frac{1}{2\pi}\int_{-\infty}^{+\infty} \frac{2i}{\omega}\cos\omega x d\omega - \frac{i}{2\pi}\int_{-\infty}^{+\infty} \frac{2i}{\omega}\sin\omega x d\omega$$

which reduces to

$$f(x) = \frac{1}{\pi}\int_{-\infty}^{+\infty} \frac{1}{\omega}\sin\omega x d\omega,$$

since the integrand of the first integral is odd. For $x > 0$ consider the Fourier transform of the function $\chi_x(t)$. For $x < 0$ perform the change of variables $u = -x$.

We saw earlier that the $F(\omega) = \chi_\Omega(\omega)$ has for its inverse Fourier transform the function $f(x) = \frac{\sin \Omega x}{\pi x}$; note that $f(0) = \frac{\Omega}{\pi}$ and $f(x) = 0$ for the first time when $\Omega x = \pi$ or $x = \frac{\pi}{\Omega}$. For any $\Omega$-bandlimited function $g(x)$ we have $G(\omega) = G(\omega)\chi_\Omega(\omega)$, so that, for any $x_0$, we have

$$g(x_0) = \int_{-\infty}^{\infty} g(x) \frac{\sin \Omega(x - x_0)}{\pi(x - x_0)} dx.$$

We describe this by saying that the function $f(x) = \frac{\sin \Omega x}{\pi x}$ has the *sifting property* for all $\Omega$-bandlimited functions $g(x)$.

As $\Omega$ grows larger, $f(0)$ approaches $+\infty$, while $f(x)$ goes to zero for $x \neq 0$. The limit is therefore not a function; it is a *generalized function* called *the Dirac delta function at zero*, denoted $\delta(x)$. For this reason the function $f(x) = \frac{\sin \Omega x}{\pi x}$ is called an *approximate delta function*. The FT of $\delta(x)$ is the function $F(\omega) = 1$ for all $\omega$. The Dirac delta function $\delta(x)$ enjoys the *sifting property* for all $g(x)$; that is,

$$g(x_0) = \int_{-\infty}^{\infty} g(x)\delta(x - x_0)dx.$$

It follows from the sifting and shifting properties that the FT of $\delta(x - x_0)$ is the function $e^{ix_0\omega}$.

The formula for the inverse FT nows says

$$\delta(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{-ix\omega} d\omega. \tag{15.1}$$

If we try to make sense of this integral according to the rules of calculus we get stuck quickly. The problem is that the integral formula doesn't mean quite what it does ordinarily and the $\delta(x)$ is not really a function, but an operator on functions; it is sometimes called a *distribution*. The Dirac deltas are mathematical fictions, not in the bad sense of being lies or fakes, but in the sense of being made up for some purpose. They provide helpful descriptions of impulsive forces, probability densities in which a discrete point has nonzero probability, or, in array processing, objects far enough away to be viewed as occupying a discrete point in space.

We shall treat the relationship expressed by equation (15.1) as a formal statement, rather than attempt to explain the use of the integral in what is surely an unconventional manner. Nevertheless, it is possible to motivate this relationship by proving that, for any $x \neq 0$,

$$\int_{-\infty}^{\infty} e^{-ix\omega} d\omega = 0.$$

Assume, for convenience, that $x > 0$. Notice first that we can write

$$\int_{-\infty}^{\infty} e^{-ix\omega} d\omega = \sum_{k=-\infty}^{\infty} \int_{\frac{2\pi}{x}k}^{\frac{2\pi}{x}(k+1)} e^{-ix\omega} d\omega.$$

Since

$$e^{-ix\omega} = e^{-ix(\omega+\frac{2\pi}{x})}$$

we can write

$$\int_{\frac{2\pi}{x}k}^{\frac{2\pi}{x}(k+1)} e^{-ix\omega}d\omega = \int_{-\frac{\pi}{x}}^{\frac{\pi}{x}} e^{-ix\omega}d\omega$$

$$= \int_0^{\frac{\pi}{x}} [e^{-ix\omega} + e^{-ix(\omega-\frac{\pi}{x})}]d\omega$$

$$= \frac{1}{x}\int_0^{\pi} [e^{-i\omega}(1+e^{i\pi})]d\omega$$

$$= \frac{1}{x}(1+e^{i\pi})\int_0^{\pi} e^{-i\omega}d\omega = 0.$$

Clearly, when $x = 0$ the integrand is one for all $\omega$, which leads to the delta function supported at zero.

If we move the discussion into the $\omega$ domain and define the Dirac delta function $\delta(\omega)$ to be the FT of the function that has the value $\frac{1}{2\pi}$ for all $x$, then the FT of the complex exponential function $\frac{1}{2\pi}e^{-i\omega_0 x}$ is $\delta(\omega-\omega_0)$, visualized as a "spike" at $\omega_0$, that is, a generalized function that has the value $+\infty$ at $\omega = \omega_0$ and zero elsewhere. This is a useful result, in that it provides the motivation for considering the Fourier transform of a signal $s(t)$ containing hidden periodicities. If $s(t)$ is a sum of complex exponentials with frequencies $-\omega_n$ then its Fourier transform will consist of Dirac delta functions $\delta(\omega-\omega_n)$. If we then estimate the Fourier transform of $s(t)$ from sampled data, we are looking for the peaks in the Fourier transform that approximate the infinitely high spikes of these delta functions.

**Exercise 6:** Use the fact that $\text{sgn}(x) = 2u(x)-1$ and the previous exercise to show that $f(x) = u(x)$ has the FT $F(\omega) = i/\omega + \pi\delta(\omega)$.

Generally, the functions $f(x)$ and $F(\omega)$ are complex-valued, so that we may speak about their real and imaginary parts. The next exercise explores the connections that hold among these real-valued functions.

**Exercise 7:** Let $f(x)$ be arbitrary and $F(\omega)$ its Fourier transform. Let $F(\omega) = R(\omega) + iX(\omega)$, where $R$ and $X$ are real-valued functions, and similarly, let $f(x) = f_1(x) + if_2(x)$, where $f_1$ and $f_2$ are real-valued. Find relationships between the pairs $R$,$X$ and $f_1$,$f_2$.

**Exercise 8:** Let $f, F$ be a FT pair. Let $g(x) = \int_{-\infty}^{x} f(y)dy$. Show that the FT of $g(x)$ is $G(\omega) = \pi F(0)\delta(\omega) + \frac{iF(\omega)}{\omega}$.

**Hint:** For $u(x)$ the Heaviside function we have

$$\int_{-\infty}^{x} f(y)dy = \int_{-\infty}^{\infty} f(y)u(x-y)dy.$$

We can use properties of the Dirac delta functions to extend the Parseval equation to Fourier transforms, where it is usually called the *Parseval-Plancherel* equation.

**Exercise 9:** Let $f(x), F(\omega)$ and $g(x), G(\omega)$ be Fourier transform pairs. Use equation (15.1) to establish the Parseval-Plancherel equation

$$\langle f, g\rangle = \int f(x)\overline{g(x)}dx = \frac{1}{2\pi}\int F(\omega)\overline{G(\omega)}d\omega,$$

from which it follows that

$$||f||^2 = \langle f, f\rangle = \int |f(x)|^2 dx = \frac{1}{2\pi}\int |F(\omega)|^2 d\omega.$$

**Exercise 10:** We define the *even part* of $f(x)$ to be the function

$$f_e(x) = \frac{f(x) + f(-x)}{2},$$

and the *odd part* of $f(x)$ to be

$$f_o(x) = \frac{f(x) - f(-x)}{2};$$

define $F_e$ and $F_o$ similarly for $F$ the FT of $f$. Let $F(\omega) = R(\omega) + iX(\omega)$ be the decomposition of $F$ into its real and imaginary parts. We say that $f$ is a *causal function* if $f(x) = 0$ for all $x < 0$. Show that, if $f$ is causal, then $R$ and $X$ are related; specifically, show that $X$ is the *Hilbert transform* of $R$, that is,

$$X(\omega) = \frac{1}{\pi}\int_{-\infty}^{\infty} \frac{R(\alpha)}{\omega - \alpha}d\alpha.$$

**Hint:** If $f(x) = 0$ for $x < 0$ then $f(x)\text{sgn}(x) = f(x)$. Apply the convolution theorem, then compare real and imaginary parts.

**Exercise 11:** The one-sided *Laplace transform* (LT) of $f$ is $\mathcal{F}$ given by

$$\mathcal{F}(z) = \int_0^{\infty} f(x)e^{-zx}dx.$$

Compute $\mathcal{F}(z)$ for $f(x) = u(x)$, the Heaviside function. Compare $\mathcal{F}(-i\omega)$ with the FT of $u$.

# Chapter 16

# The Uncertainty Principle

We saw earlier that the Fourier transform of the function $f(x) = e^{-\alpha^2 x^2}$ is

$$F(\omega) = \frac{\sqrt{\pi}}{\alpha} e^{-(\frac{\omega}{2\alpha})^2}.$$

This Fourier transform pair illustrates well the general fact that the more concentrated $f(x)$ is, the more spread out $F(\omega)$ is. In particular, it is impossible for both $f$ and $F$ to have bounded support. We prove the following inequality:

$$\frac{\int x^2 |f(x)|^2 dx}{\int |f(x)|^2 dx} \frac{\int \omega^2 |F(\omega)|^2 d\omega}{\int |F(\omega)|^2 d\omega} \geq \frac{1}{4}. \tag{16.1}$$

This inequality is the mathematical version of Heisenberg's Uncertainty Principle.

The Parseval-Plancherel equation tells us that

$$\int f(x)\overline{g(x)} dx = \frac{1}{2\pi} \int F(\omega)\overline{G(\omega)} d\omega$$

for any Fourier transform pairs $f, F$ and $g, G$. In particular, if $g = f$ we get

$$\int |f(x)|^2 dx = \frac{1}{2\pi} \int |F(\omega)|^2 d\omega.$$

We'll need Parseval's theorem in the proof of the uncertainty principle as well as this result from an earlier exercise (see equation (2.1)): for any two complex numbers $z$ and $w$ we have

$$|zw| \geq \frac{1}{2}(z\overline{w} + \overline{z}w).$$

We have

$$\frac{1}{2\pi}\int x^2|f(x)|^2dx\int \omega^2|F(\omega)|^2d\omega$$
$$=\frac{1}{2\pi}\int |xf(x)|^2dx\int |\omega F(\omega)|^2d\omega$$
$$=\frac{1}{2\pi}\int |xf(x)|^2dx\int |f'(x)|^2dx$$
$$\geq (\int |xf'(x)f(x)|dx)^2 \geq (\int \frac{x}{2}[f'(x)\overline{f(x)}+f(x)\overline{f'(x)}]dx)^2$$
$$=\frac{1}{4}(\int x(\frac{d}{dx}|f(x)|^2)dx)^2$$
$$=\frac{1}{4}(\int |f(x)|^2dx)^2=\frac{1}{8\pi}\int |f(x)|^2dx\int |F(\omega)|^2d\omega.$$

This completes the proof of the inequality (16.1).

To better understand the significance of this inequality, we reformulate it in terms of the variances of probability densities. Suppose that

$$\int |f(x)|^2dx=\int |F(\omega)|^2d\omega=1,$$

so that we may view $|f(x)|^2$ and $|F(\omega)|^2$ as probability density functions associated with random variables $X$ and $Y$, respectively. From probability theory we know that the expected values $E(X)$ and $E(Y)$ are given by

$$m=E(X)=\int x|f(x)|^2dx$$

and

$$M=E(Y)=\int \omega|F(\omega)|^2d\omega.$$

Let

$$g(x)=f(x+m)e^{iMx},$$

so that the Fourier transform of $g(x)$ is

$$G(\omega)=F(\omega+M)e^{i(M-\omega)m}.$$

Then $|g(x)|^2=|f(x+m)|^2$ and $|G(\omega)|^2=|F(\omega+M)|^2$; we also have

$$\int x|g(x)|^2dx=0$$

and

$$\int \omega|G(\omega)|^2d\omega=0.$$

The point here is that we can assume that $m = 0$ and $M = 0$. Consequently the variance of $X$ is

$$var(X) = \int x^2 |f(x)|^2 dx$$

and the variance of $Y$ is

$$var(Y) = \int \omega^2 |F(\omega)|^2 d\omega.$$

The variances measure how spread out the functions $|f(x)|^2$ and $|F(\omega)|^2$ are around their respective means. From the inequality (16.1) we know that the product of these variances is not smaller than $\frac{1}{4}$.

**Exercise 1:** Show, by examining the proof of inequality (16.1), that if the inequality is an equation for some $f$ then $f'(x) = kxf(x)$, so that $f(x) = e^{-\alpha^2 x^2}$ for some $\alpha > 0$.
**Hint:** What can be said when Cauchy's inequality is an equation?

# Chapter 17

# Directional Transmission

An important example of the use of the DFT is the design of directional transmitting or receiving arrays of antennas. In this chapter we concentrate on the transmission case; we shall return to array processing and consider the passive or receiving case in a later chapter.

Parabolic mirrors behind car headlamps reflect the light from the bulb, concentrating it directly ahead. Whispering at one focal point of an elliptical room can be heard clearly at the other focal point. When I call to someone across the street I cup my hands in the form of a megaphone to concentrate the sound in that direction. In all these cases the transmitted signal has acquired *directionality*. In the case of the elliptical room, not only does the soft whispering reflect off the walls toward the opposite focal point, but the travel times are independent of where on the wall the reflections occur; otherwise, the differences in time would make the received sound unintelligible. Parabolic satellite dishes perform much the same function, concentrating incoming signals coherently. In this chapter we discuss the use of amplitude and phase modulation of transmitted signals to concentrate the signal power in certain directions. Following the lead of Richard Feynman in [91], we use radio broadcasting as a concrete example of the use of directional transmission.

Radio broadcasts are meant to be received and the amount of energy that reaches the receiver depends on the amount of energy put into the transmission as well as on the distance from the transmitter to the receiver. If the transmitter broadcasts a spherical wave front, with equal power in all directions, the energy in the signal is the same over the spherical wavefronts, so that the energy per unit area is proportional to the reciprocal of the surface area of the front. This means that, for omni-directional broadcasting, the energy per unit area, that is, the energy supplied to any receiver, falls off as the distance squared. The amplitude of the received signal is then proportional to the reciprocal of the distance.

Suppose you owned a radio station in Los Angeles. Most of the population resides along the north-south coast, with fewer to the east, in the desert, and fewer still to the west, in the Pacific Ocean. You might well want to transmit the radio signal in a way that concentrates most of the power north and south. But how can you do this? The answer is to broadcast directionally. By shaping the wavefront to have most of its surface area north and south you will enable to have the broadcast heard by more people without increasing the total energy in the transmission. To achieve this shaping you can use an array of multiple antennas.

**Multiple antenna arrays:** We place $2N+1$ transmitting antennas a distance $\Delta > 0$ apart along an east-west axis, as shown in Figure 71.1. For convenience, let the locations of the antennas be $n\Delta$, $n = -N, ..., N$. To begin with, let us suppose that we have a fixed frequency $\omega$ and each of the transmitting antennas sends out the same signal $f_n(t) = \frac{1}{\sqrt{2N+1}}\cos(\omega t)$. With this normalization the total energy is independent of $N$. Let $(x, y)$ be an arbitrary location on the ground and let $\mathbf{s}$ be the vector from the origin to the point $(x, y)$. Let $\theta$ be the angle measured counterclockwise from the positive horizontal axis to the vector $\mathbf{s}$. Let $D$ be the distance from $(x, y)$ to the origin. Then, if $(x, y)$ is sufficiently distant from the antennas, the distance from $n\Delta$ on the horizontal axis to $(x, y)$ is approximately $D - n\Delta\cos(\theta)$. The signals arriving at $(x, y)$ from the various antennas will have travelled for different times and so will be out of phase with one another to a degree that depends on the location of $(x, y)$.

Since we are concerned only with wavefront shape, we omit for now the distance-dependence in the amplitude of the received signal. The signal received at $(x, y)$ is proportional to

$$f(\mathbf{s}, t) = \frac{1}{\sqrt{2N+1}} \sum_{n=-N}^{N} \cos(\omega(t - t_n)),$$

where

$$t_n = \frac{1}{c}(D - n\Delta\cos(\theta))$$

and $c$ is the speed of propagation of the signal. Writing

$$\cos(\omega(t - t_n)) = \cos(\omega(t - \frac{D}{c}) + n\gamma\cos(\theta))$$

for $\gamma = \frac{\omega\Delta}{c}$, we have

$$\cos(\omega(t-t_n)) = \cos(\omega(t-\frac{D}{c}))\cos(n\gamma\cos(\theta)) - \sin(\omega(t-\frac{D}{c}))\sin(n\gamma\cos(\theta)).$$

Therefore the signal received at $(x, y)$ is

$$f(\mathbf{s}, t) = \frac{1}{\sqrt{2N+1}} A(\theta)\cos(\omega(t - \frac{D}{c})) \tag{17.1}$$

for

$$A(\theta) = \frac{\sin((N+\frac{1}{2})\gamma\cos(\theta))}{\sin(\frac{1}{2}\gamma\cos(\theta))};$$

when the denominator equals zero the signal equals $\sqrt{2N+1}\cos(\omega(t-\frac{D}{c}))$.

We see from equation (17.1) that the maximum power is in the north-south direction. What about the east-west direction? In order to have negligible signal power wasted in the east-west direction we want the numerator in equation (17.1) to be zero when $\theta = 0$. This means that $\Delta = m\lambda/(2N+1)$, where $\lambda = 2\pi c/\omega$ is the wavelength and $m$ is some positive integer. Recall that the wavelength for broadcast radio is tens to hundreds of meters.

**Exercise 1:** Graph the function $A(\theta)$ in polar coordinates for various choices of $N$ and $\Delta$.

**Phase and Amplitude Modulation:** In the previous section the signal broadcast from each of the antennas was the same. Now we look at what directionality can be obtained by using different amplitudes and phases at each of the antennas. Let the signal broadcast from the antenna at $n\Delta$ be

$$f_n(t) = |A_n|\cos(\omega t - \phi_n) = |A_n|\cos(\omega(t-\tau_n)),$$

for some amplitude $|A_n| > 0$ and phase $\phi_n = \omega\tau_n$. Now the signal received at $\mathbf{s}$ is proportional to

$$f(\mathbf{s},t) = \sum_{n=-N}^{N} |A_n|\cos(\omega(t-t_n-\tau_n)). \tag{17.2}$$

If we wish, we can repeat the calculations done earlier to see what the effect of the amplitude and phase changes is. Using complex notation simplifies things somewhat.

Let us consider a complex signal; suppose that the signal transmitted from the antenna at $n\Delta$ is $g_n(t) = |A_n|e^{i\omega(t-\tau_n)}$. Then the signal received at location $\mathbf{s}$ is proportional to

$$g(\mathbf{s},t) = \sum_{n=-N}^{N} |A_n|e^{i\omega(t-t_n-\tau_n)}.$$

Then we have

$$g(\mathbf{s},t) = B(\theta)e^{i\omega(t-\frac{D}{c})}$$

for $A_n = |A_n|e^{-i\phi_n}$ and $x = \frac{\omega\Delta}{c}\sin(\theta)$. Note that the complex amplitude function $B(\theta)$ depends on our choices of $N$ and $\Delta$ and takes the form of a finite Fourier series or DFT. We can design $B(\theta)$ to approximate the

desired directionality by choosing the appropriate complex coefficients $A_n$ and selecting the amplitudes $|A_n|$ and phases $\phi_n$ accordingly. We can generalize further by allowing the antennas to be spaced irregularly along the east-west axis, or even distributed irregularly over a two-dimensional area on the ground.

**Exercise 2:** Use the Fourier transform of the characteristic function of an interval to design a transmitting array that maximally concentrates signal power within the sectors northwest to northeast and southwest to southeast.

**Maximal concentration in a sector:** Suppose we want to concentrate the transmission power in the directions represented by $x \in [a, b]$ where $[a, b]$ is a subinterval of $[-\pi, \pi]$. Let $\mathbf{u} = (A_{-N}, ..., A_N)^T$ be the vector of coefficients for the function

$$B(x) = \sum_{n=-N}^{N} A_n e^{-inx}.$$

**Exercise 3:** Show that

$$\frac{1}{2\pi} \int_{-\pi}^{\pi} |B(x)|^2 dx = \mathbf{u}^\dagger \mathbf{u},$$

and

$$\frac{1}{2\pi} \int_{a}^{b} |B(x)|^2 dx = \mathbf{u}^\dagger Q \mathbf{u},$$

where $Q$ is the matrix with entries

$$Q_{mn} = \frac{1}{2\pi} \int_{a}^{b} \exp(i(n-m)x) dx.$$

Maximizing the concentration of power within the interval $[a, b]$ is then equivalent to finding the vector $\mathbf{u}$ that maximizes the ratio $\mathbf{u}^\dagger Q \mathbf{u} / \mathbf{u}^\dagger \mathbf{u}$. The matrix $Q$ is positive-definite, all its eigenvalues are positive and the optimal $\mathbf{u}$ is the eigenvector of $Q$ associated with the largest eigenvalue. This largest eigenvalue is the desired ratio and is always less that one. As $N$ increases this ratio approaches one, for any fixed sub-interval $[a, b]$.

The figures below show that transmission pattern $A(\theta)$ for various choices of $m$ and $N$. In Figure 17.2 $N = 5$ for each plot and the $m$ changes, illustrating the effect of changing the spacing of the array elements. The plots in Figure 17.3 differ from those in Figure 17.2 only in that $N = 21$ now. In Figure 17.4 we allow the $m$ to be less than one, showing the loss of the nulls in the east and west directions.

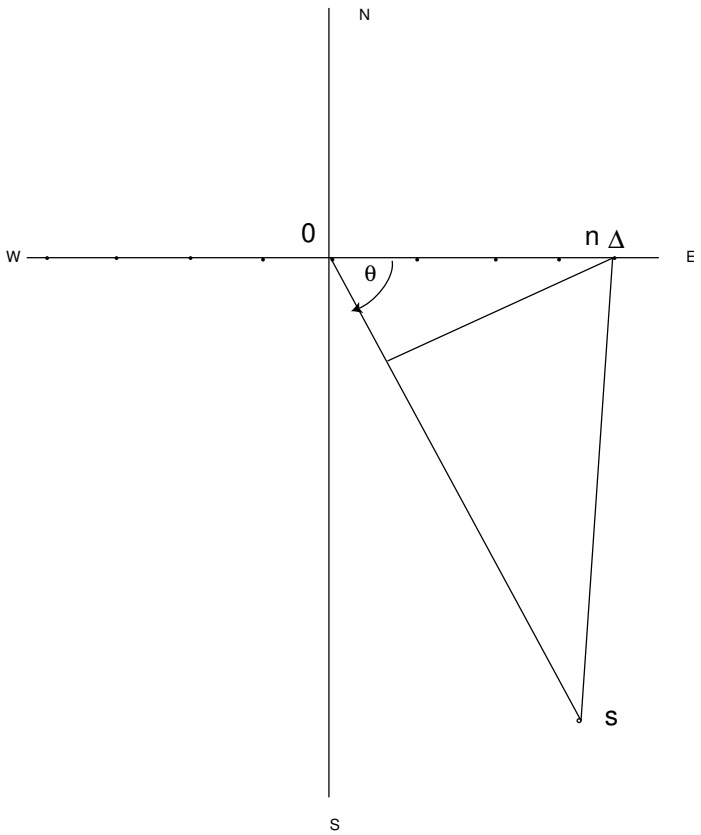



Figure 17.1:

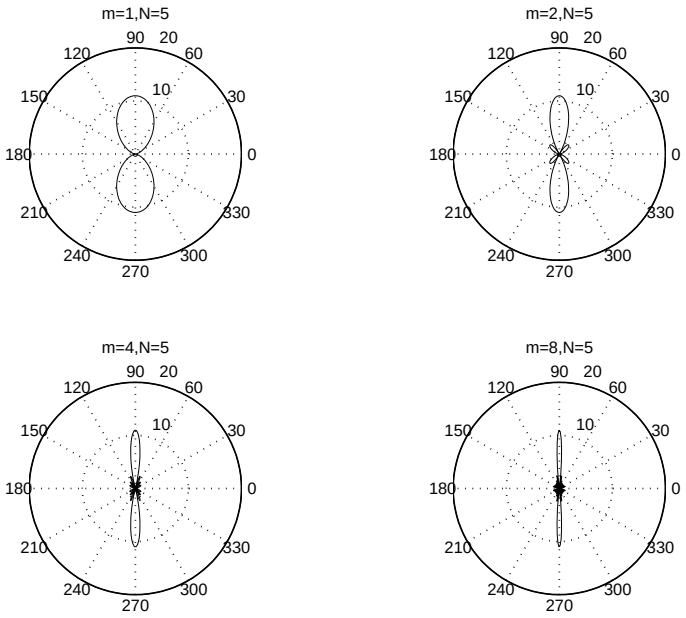



Figure 17.2: Transmission Pattern $A(\theta)$: $m = 1, 2, 4, 8$ and $N = 5$

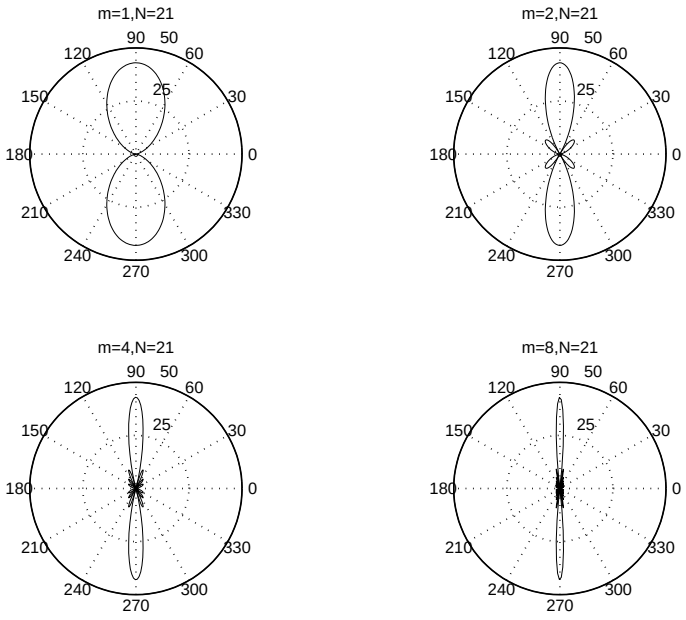



Figure 17.3: Transmission Pattern $A(\theta)$: $m = 1, 2, 4, 8$ and $N = 21$

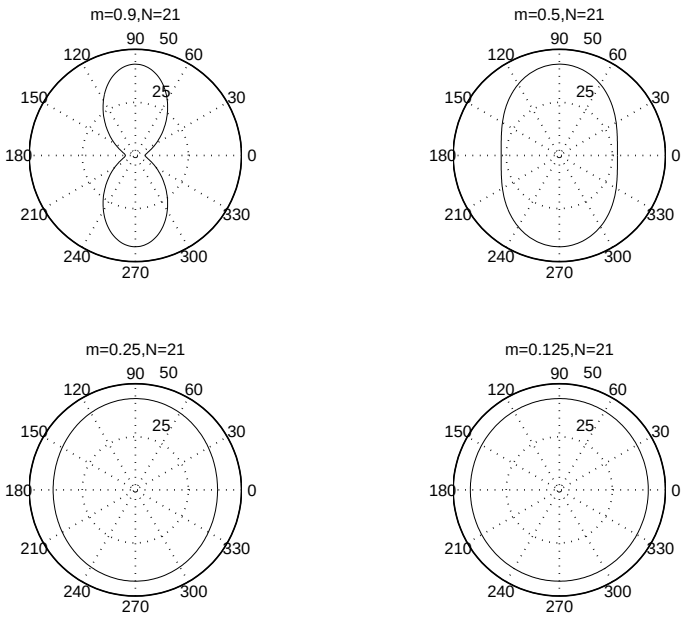



Figure 17.4: Transmission Pattern $A(\theta)$: $m = 0.9, 0.5, 0.25, 0.125$ and $N = 21$

# Chapter 18

# Analysis and Synthesis

An important theme that runs through most of mathematics, from the geometry of the early Greeks to modern signal processing, is *analysis and synthesis*, or, less formally, *breaking up and putting back together*. The Greeks estimated the area of a circle by breaking it up into sectors that approximated triangles. The Riemann approach to integration involves breaking up the area under a curve into pieces that approximate rectangles or other simple shapes. Viewed differently, the Riemann approach is first to approximate the function to be integrated by a step function and then to integrate the step function.

Euclid includes a good deal of number theory along with his geometry; there also we find analysis and synthesis. His theorem that every positive integer is divisible by a prime is analysis; division does the breaking up and the simple pieces are the primes. The fundamental theorem of arithmetic, which asserts that every positive integer can be written in an essentially unique way as the product of powers of primes, is synthesis, with the putting together done by multiplication.

Analysis and synthesis in signal processing refers to the effort to study complicated functions in terms of simpler ones. The individual power functions, $x^n$, are not particularly interesting by themselves, but when finitely many of them are scaled and added to form a polynomial, interesting functions can result, as the famous approximation theorem of Weierstrass confirms [127]:

**Theorem 18.1** *If $f : [a, b] \to R$ is continous and $\epsilon > 0$ is given we can find a polynomial $P$ such that $|f(x) - P(x)| \leq \epsilon$ for every $x$ in $[a, b]$.*

The idea of building complicated functions from powers is carried a step further with the use of infinite series, such as Taylor series. The sine function, for example, can be represented for all real $x$ by the infinite power

series

$$\sin x = x - \frac{1}{3!}x^3 + \frac{1}{5!}x^5 - \frac{1}{7!}x^7 + ....$$

The most interesting thing to note about this is that the sine function has properties that none of the individual power functions possess: for example, it is bounded and periodic. So we see that an infinite sum of simple functions can be qualitatively different from the components in the sum. If we take the sum of only finitely many terms in the Taylor series for the sine function we get a polynomial, which cannot provide a good approximation of the sine function for all $x$; that is, the finite sum does not approximate the sine function uniformly over the real line. The approximation is better for $x$ near zero and poorer as we move away from zero. However, for any selected $x$ and for any $\epsilon > 0$ there is a positive integer $N$, depending on the $x$ and on the $\epsilon$, with the sum of the first $N$ terms of the series within $\epsilon$ of $\sin x$; that is, the series converges pointwise to $\sin x$ for each real $x$. In Fourier analysis the trigonometric functions themselves are viewed as the simple functions and we try to build more complicated functions as (possibly infinite) sums of trig functions. In wavelet analysis we have more freedom to design the simple functions to fit the problem at hand.

When we speak of *signal analysis* we often mean that we believe the signal to be a superposition of simpler signals of a known type and we wish to know which of these simpler signals are involved and to what extent. For example, received sonar or radar data may be the superposition of individual components corresponding to spatially localized targets of interest. As we shall see in our discussion of the ambiguity function and of wavelets, we want to tailor the family of simpler signals to fit the physical problem being considered.

Sometimes it is not the individual components that are significant by themselves, but groupings of these components. For example, if our received signal is believed to consist of a lower frequency signal of interest plus a noise component employing both low and high frequencies, we can remove some of the noise by performing a low-pass filtering. This amounts to analyzing the received signal to determine what its low-pass and high-pass components are. We formulate this operation mathematically using the Fourier transform, which decomposes the received signal $f(t)$ into complex exponential function components corresponding to different frequencies.

More generally, we may analyze a signal $f(t)$ by calculating certain inner products $\langle f, g_n \rangle, n = 1, ..., N$. We may wish to encode the signal using these $N$ numbers, or to make a decision about the signal, such as recognizing a voice. If the signal is a two-dimensional image, say a fingerprint, we may want to construct a data-base of these $N$-dimensional vectors, for identification. In such a case we are not necessarily claiming that the signal $f(t)$ is a superposition of the $g_n(t)$ in any sense, nor do we necessarily expect to reconstruct $f(t)$ at some later date from the stored inner products.

For example, one might identify a piece of music using only the upward or downward progression of the first few notes.

There are many cases, on the other hand, in which we do wish to reconstruct the signal $f(t)$ from measurements or stored compressed versions. In such cases we need to consider this when we design the measuring or compression procedures. For example, we may have values of the signal or its Fourier transform at some finite number of points and want to recapture $f(t)$ itself. Even in those cases mentioned above in which reconstruction is not desired, such as the fingerprint case, we do wish to be reasonably sure that similar vectors of inner products correspond to similar signals and distinct vectors of inner products correspond to distinct signals, within the obvious limitations imposed by the finiteness of the stored inner products. The twin processes of analysis and synthesis are dealt with mathematically using the notions of *frames* and *bases*.

**Frames:** Although in practice we deal with finitely many measurements or inner product values, it is convenient, in theoretical discussions, to imagine that the signal $f(t)$ has been associated with an infinite sequence of inner products $\{\langle f, g_n\rangle, n = 1, 2, ...\}$. It is also convenient to assume that $||f||^2 = \int_{-\infty}^{\infty} |f(t)|^2 dt < +\infty$; that is, we assume that $f$ is in the Hilbert space $H = L^2$. The sequence $\{g_n | n = 1, 2, ...\}$ in any Hilbert space $H$ is called a *frame* for $H$ if there are positive constants $A \leq B$ such that, for all $f$ in $H$,

$$A||f||^2 \leq \sum_{n=1}^{\infty} |\langle f, g_n\rangle|^2 \leq B||f||^2. \tag{18.1}$$

The inequalities in (18.1) define the *frame property*. A frame is said to be *tight* if $A = B$.

To motivate this definition, suppose that $f = g - h$. If $g$ and $h$ are nearly equal, then $f$ is near zero, so that $||f||^2$ is near zero. Consequently, the numbers $|\langle f, g_n\rangle|^2$ are all small, meaning that $\langle g, g_n\rangle$ is nearly equal to $\langle h, g_n\rangle$ for each $n$. Conversely, if $\langle g, g_n\rangle$ is nearly equal to $\langle h, g_n\rangle$ for each $n$, then the numbers $|\langle f, g_n\rangle|^2$ are all small. Consequently $||f||^2$ is small, from which we conclude that $g$ is close to $h$. The *analysis* operator is the one that takes us from $f$ to the sequence $\{\langle f, g_n\rangle\}$, while the *synthesis* operator takes us from the sequence $\{\langle f, g_n\rangle\}$ to $f$. This discussion of frames and related notions is based on the treatment in Christensen's book [66].

In the case of finite dimensional space, any finite set $\{g_n,\ n = 1, ..., N\}$ is a frame for the space $H$ of all $f$ that are linear combinations of the $g_n$.

**Exercise 1:** An interesting example of a frame in $H = R^2$ is the so-called *Mercedes frame*: let $g_1 = (0, 1)$, $g_2 = (-\sqrt{3}/2, -1/2)$ and $g_3 = (\sqrt{3}/2, -1/2)$. Show that for this frame $A = B = 3/2$, so the Mercedes frame is tight.

The frame property in (18.1) provides a necessary condition for stable application of the decomposition and reconstruction operators. But it does more than that- it actually provides a reconstruction algorithm. The *frame operator* $S$ is given by

$$Sf = \sum_{n=1}^{\infty} \langle f, g_n \rangle \, g_n.$$

The frame property implies that the frame operator is invertible. The *dual frame* is the sequence $\{S^{-1}g_n,\, n = 1, 2, ...\}$.

**Exercise 2:** Use the definitions of the frame operator $S$ and the dual frame to obtain the following reconstruction formulas:

$$f = \sum_{n=1}^{\infty} \langle f, g_n \rangle \, S^{-1} g_n;$$

and

$$f = \sum_{n=1}^{\infty} \langle f, S^{-1} g_n \rangle \, g_n.$$

If the frame is tight then the dual frame is $\{\frac{1}{A} g_n,\, n = 1, 2, ...\}$; if the frame is not tight, inversion of the frame operator is done only approximately.

**Bases, Riesz bases and orthonormal bases:** The sequence $\{g_n,\, n = 1, 2, ...\}$ in $H$ is a *basis* for $H$ if, for every $f$ in $H$, there is a unique sequence $\{c_n,\, n = 1, 2, ...\}$ with

$$f = \sum_{n=1}^{\infty} c_n g_n.$$

A basis is called a *Riesz basis* if it is also a frame for $H$. It can be shown that a frame is a Riesz basis if the removal of any one element causes the loss of the frame property; since the second inequality in (18.1) is not lost, it follows that it is the first inequality that can now be violated for some $f$. A basis is an *orthonormal basis* for $H$ if $||g_n|| = 1$ for all $n$ and $\langle g_n, g_m \rangle = 0$ for distinct $m$ and $n$.

We know that the complex exponentials

$$\{e_n(t) = \frac{1}{\sqrt{2\pi}} e^{int},\, -\infty < n < \infty\}$$

form an orthonormal basis for the Hilbert space $L^2(-\pi, \pi)$ consisting of all $f$ supported on $(-\pi, \pi)$ with $\int_{-\pi}^{\pi} |f(t)|^2 dt < +\infty$. Every such $f$ can be written as

$$f(t) = \frac{1}{\sqrt{2\pi}} \sum_{n=-\infty}^{+\infty} a_n e^{int},$$

for

$$a_n = \langle f, e_n \rangle = \frac{1}{\sqrt{2\pi}} \int_{-\pi}^{\pi} f(t) e^{-int} dt.$$

Consequently, this is true for every $f$ in $L^2(-\pi/2, \pi/2)$, although the set of functions $\{g_n\}$ formed by restricting the $\{e_n\}$ to the interval $(-\pi/2, \pi/2)$ is no longer a basis for $H = L^2(-\pi/2, \pi/2)$. It is still a tight frame with $A = 1$, but is no longer normalized, since the norm of $g_n$ in $L^2(-\pi/2, \pi/2)$ is $1/\sqrt{2}$. An orthonormal basis can be characterized as any sequence with $||g_n|| = 1$ for all $n$ that is a tight frame with $A = 1$. The sequence $\{\sqrt{2}g_{2k}, k = -\infty, ..., \infty\}$ is an orthonormal basis for $L^2(-\pi/2, \pi/2)$, as is the sequence $\{\sqrt{2}g_{2k+1}, k = -\infty, ..., \infty\}$. The sequence $\{\langle f, g_n \rangle, n = -\infty, ..., \infty\}$ is redundant; the half corresponding either to the odd $n$ or the half corresponding to the even $n$ suffices to recover $f$. Because of this redundancy we can tolerate more inaccuracy in measuring these values; indeed, this is one of the main attractions of frames in signal processing.

# Chapter 19

# Ambiguity Functions

We turn now to signal processing problems arising in *radar*. Not only does radar provide an important illustration of the application of the theory of Fourier transforms and matched filters, but it also serves to motivate several of the mathematical concepts we shall encounter in our discussion of wavelets. The connection between radar signal processing and wavelets is discussed in some detail in Kaiser's book [123].

In radar a real-valued function $\psi(t)$ representing a time-varying voltage is converted by an antenna in transmission mode into a propagating electromagnetic wave. When this wave encounters a reflecting target an echo is produced. The antenna, now in receiving mode, picks up the echo $f(t)$, which is related to the original signal by

$$f(t) = A\psi(t - d(t)),$$

where $d(t)$ is the time required for the original signal to make the round trip from the antenna to the target and return back at time $t$. The amplitude $A$ incorporates the reflectivity of the target as well as attenuation suffered by the signal. As we shall see shortly, the delay $d(t)$ depends on the distance from the antenna to the target and, if the target is moving, on its radial velocity. The main signal processing problem is to determine target range and radial velocity from knowledge of $f(t)$ and $\psi(t)$.

If the target is stationary, at a distance $r_0$ from the antenna, then $d(t) = 2r_0/c$, where $c$ is the speed of light. In this case the original signal and the received echo are related simply by

$$f(t) = A\psi(t - b),$$

for $b = 2r_0/c$. When the target is moving so that its distance to the antenna, $r(t)$, is time-dependent, the relationship between $f$ and $\psi$ is more complicated.

**Exercise 1:** Suppose the target has radial velocity $v$, with $v > 0$ indicating away from the antenna. Show that the delay function $d(t)$ is now

$$d(t) = 2\frac{r_0 + vt}{c + v}$$

and $f(t)$ is related to $\psi(t)$ according to

$$f(t) = A\psi(\frac{t-b}{a}), \tag{19.1}$$

for

$$a = \frac{c+v}{c-v}$$

and

$$b = \frac{2r_0}{c-v}.$$

Show also that if we select $A = (\frac{c-v}{c+v})^{1/2}$ then energy is preserved; that is, $||f|| = ||\psi||$.

**Exercise 2:** Let $\Psi(\omega)$ be the Fourier transform of the signal $\psi(t)$. Show that the Fourier transform of the echo $f(t)$ in equation (19.1) is then

$$F(\omega) = Aae^{ib\omega}\Psi(a\omega). \tag{19.2}$$

The basic problem is to determine $a$ and $b$, and therefore the range and radial velocity of the target, from knowledge of $f(t)$ and $\psi(t)$. An obvious approach is to do a matched filter.

## The wideband cross-ambiguity function:

Note that the received echo $f(t)$ is related to the original signal by the operations of rescaling and shifting. We therefore match the received echo with all the shifted and rescaled versions of the original signal. For each $a > 0$ and real $b$ let

$$\psi_{a,b}(t) = \psi(\frac{t-b}{a}).$$

The *wideband cross-ambiguity function* (WCAF) is

$$(W_\psi f)(b, a) = \frac{1}{\sqrt{a}}\int_{-\infty}^{\infty} f(t)\psi_{a,b}(t)dt. \tag{19.3}$$

In the ideal case the values of $a$ and $b$ for which the WCAF takes on its largest absolute value should be the true values of $a$ and $b$.

More generally, there will be many individual targets or sources of echos, each having their own values of $a$, $b$ and $A$. The resulting received echo

function $f(t)$ is a superposition of the individual functions $\psi_{a,b}(t)$, which, for technical reasons, we write as

$$f(t) = \int_{-\infty}^{\infty} \int_{0}^{\infty} D(b,a)\psi_{a,b}(t)\frac{dadb}{a^2}. \tag{19.4}$$

We then have the inverse problem of determining $D(b,a)$ from $f(t)$.

Equation (19.4) provides a representation of the echo $f(t)$ as a superposition of rescaled translates of a single function, namely the original signal $\psi(t)$. We shall encounter this representation again in our discussion of wavelets, where the signal $\psi(t)$ is called the *mother wavelet* and the WCAF is called the *integral wavelet transform*. One reason for discussing radar and ambiguity functions now is to motivate some of the wavelet theory. Our discussion here follows closely the treatment in [123], where Kaiser emphasizes the important connections between wavelets and radar ambiguity functions.

As we shall see in the chapter on wavelets, we can recover the signal $f(t)$ from the WCAF using the following inversion formula: at points $t$ where $f(t)$ is continuous we have

$$f(t) = \frac{1}{C_\psi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (W_\psi f)(b,a)\psi(\frac{t-b}{a})\frac{dadb}{a^2},$$

with

$$C_\psi = \int_{-\infty}^{\infty} \frac{|\Psi(\omega)|^2}{|\omega|} d\omega$$

for $\Psi(\omega)$ the Fourier transform of $\psi(t)$. The obvious conjecture is then that the distribution functon $D(b,a)$ is then

$$D(b,a) = \frac{1}{C_\psi}(W_\psi f)(b,a).$$

However, this is not generally the case. Indeed, there is no particular reason why the physically meaningful function $D(b,a)$ must have the form $(W_\psi g)(b,a)$ for some function $g$. So the inverse problem of estimating $D(b,a)$ from $f(t)$ is more complicated. One approach mentioned in [123] involves transmitting more than one signal $\psi(t)$ and estimating $D(b,a)$ from the echos corresponding to each of the several different transmitted signals.

## The narrowband cross-ambiguity function:

The real signal $\psi(t)$ with Fourier transform $\Psi(\omega)$ is said to be a *narrowband signal* if there are constants $\alpha$ and $\gamma$ such that the conjugate-symmetric function $\Psi(\omega)$ is concentrated on $\alpha \leq |\omega| \leq \gamma$ and $\frac{\gamma-\alpha}{\gamma+\alpha}$ is nearly equal to

zero, which means that $\alpha$ is very much greater than $\beta = \frac{\gamma-\alpha}{2}$. The center frequency is $\omega_c = \frac{\gamma+\alpha}{2}$.

**Exercise 3:** Let $\phi = 2\omega_c v/c$. Show that $a\omega_c$ is approximately equal to $\omega_c + \phi$.

It follows then that, for $\omega > 0$, $F(\omega)$, the Fourier transform of the echo $f(t)$, is approximately $Aae^{ib\omega}\Psi(\omega + \phi)$. Because the Doppler shift affects positive and negative frequencies differently it is convenient to construct a related signal having only positive frequency components.

Let $G(\omega) = 2F(\omega)$ for $\omega > 0$ and $G(\omega) = 0$ otherwise. Let $g(t)$ be the inverse Fourier transform of $G(\omega)$. Then the complex-valued function $g(t)$ is called the *analytic signal* associated with $f(t)$. The function $f(t)$ is the real part of $g(t)$; the imaginary part of $g(t)$ is the *Hilbert transform* of $f(t)$. Then the *demodulated analytic signal* associated with $f(t)$ is $h(t)$ with Fourier transform $H(\omega) = G(\omega+\omega_c)$. Similarly, let $\gamma(t)$ be the demodulated analytic signal associated with $\psi(t)$.

**Exercise 4:** Show that the demodulated analytic signals $h(t)$ and $\gamma(t)$ are related by

$$h(t) = Be^{i\phi t}\gamma(t-b) = B\gamma_{\phi,b}(t),$$

for $B$ a time-independent constant.
**Hint:** Use the fact that $\Psi(\omega) = 0$ for $0 \leq \omega < \alpha$ and $\phi < \alpha$.

To determine the range and radial velocity in the narrowband case we again use the matched filter, forming the *narrowband cross-ambiguity function* (NCAF)

$$N_h(\phi, b) = \langle h, \gamma_{\phi,b}\rangle = \int_{-\infty}^{\infty} h(t)e^{-i\phi t}\overline{\gamma(t-b)}dt. \tag{19.5}$$

Ideally, the values of $\phi$ and $b$ corresponding to the largest absolute value of $N_h(\phi, b)$ will be the true ones, from which the range and radial velocity can be determined. For each fixed value of $b$ the NCAF is the Fourier transform of the function $h(t)\overline{\gamma(t-b)}$, evaluated at $\omega = -\phi$; so the NCAF contains complete information about the function $h(t)$. In the chapter on wavelets we shall consider the NCAF in a different light, with $\gamma$ playing the role of a window function and the NCAF the short-time Fourier transform of $h(t)$, describing the frequency content of $h(t)$ near the time $b$.

In the more general case in which the narrowband echo function $f(t)$ is a superposition of narrowband reflections,

$$f(t) = \int_{-\infty}^{\infty}\int_{0}^{\infty} D(b, a)\psi_{a,b}(t)\frac{dadb}{a^2},$$

we have

$$h(t) = \int_{-\infty}^{\infty} \int_{0}^{\infty} D_{NB}(b, \phi) e^{i\phi t} \gamma(t - b) d\phi db,$$

where $D_{NB}(b, \phi)$ is the narrowband distribution of reflecting target points, as a function of $b$ and $\phi = 2\omega_c v/c$. The inverse problem now is to estimate this distribution, given $h(t)$.

**Range estimation:** If the transmitted signal is $\psi(t) = e^{i\omega t}$ and the target is stationary at range $r$, then the echo received is $f(t) = Ae^{i\omega(t-b)}$, where $b = 2r/c$. So our information about $r$ is that we know the value $e^{2i\omega r/c}$. Because of the periodicity of the complex exponential function, this is not enough information to determine $r$; we need $e^{2i\omega r/c}$ for a variety of values of $\omega$. To obtain these values we can transmit a signal whose frequency changes with time, such as a *chirp* of the form

$$\psi(t) = e^{i\omega t^2}$$

with the frequency $2\omega t$ at time $t$.

# Chapter 20

# Time-Frequency Analysis

There are applications in which the frequency composition of the signal of interest will change over time. A good analogy is a piece of music, in which notes at certain frequencies are heard for a while and then are replaced by notes at other frequencies. We do not usually care what the overall contribution of, say, middle C, is to the song, but do want to know which notes are to be sounded when and for how long. Analyzing such non-stationary signals requires tools other than the Fourier transform: the short-time Fourier transform is one such tool; wavelet expansion is another.

The inverse Fourier transform formula

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) e^{-i\omega t} d\omega$$

provides a representation of the function of time $f(t)$ as a superposition of sinusoids $e^{-i\omega t}$ with frequencies $\omega$. The value at $\omega$ of the Fourier transform

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

is the complex amplitude associated with the sinusoidal component $e^{-i\omega t}$. It quantifies the contribution to $f(t)$ made by that sinusoid, over all of $t$. To determine each individual number $F(\omega)$ we need $f(t)$ for all $t$. It is implicit that the frequency content has not changed over time.

**The short-time Fourier transform:** To estimate the frequency content of the signal $f(t)$ around the time $t = b$ we could proceed as follows. Multiply $f(t)$ by the function that is equal to $\frac{1}{2\epsilon}$ on the interval $[b-\epsilon, b+\epsilon]$ and zero otherwise. Then take the Fourier transform. The multiplication step is called *windowing*.

To see how well this works, consider the case in which $f(t) = \exp(-i\omega_0 t)$ for all $t$. The Fourier transform of the windowed signal is then

$$\exp(i(\omega - \omega_0)b)\frac{\sin(\epsilon(\omega - \omega_0))}{\epsilon(\omega - \omega_0)}.$$

This function attains its maximum value of one at $\omega = \omega_0$. But, the first zeros of the function are at $|\omega - \omega_0| = \frac{\pi}{\epsilon}$, which says that as $\epsilon$ gets smaller the windowed Fourier transform spreads out more and more around $\omega = \omega_0$; that is, better time localization comes at the price of worse frequency localization. To achieve a somewhat better result we can change the window function.

The standard normal (or Gaussian) curve is

$$g(t) = \frac{1}{\sqrt{2\pi}} \exp(-\frac{1}{2}t^2),$$

which has its peak at $t = 0$ and falls off to zero symmetrically on either side. For $\sigma > 0$ let

$$g_\sigma(t) = \frac{1}{\sigma} g(t/\sigma).$$

Then the function $g_\sigma(t - b)$ is centered at $t = b$ and falls off on either side, more slowly for large $\sigma$, faster for smaller $\sigma$. Also we have

$$\int_{-\infty}^{\infty} g_\sigma(t - b)dt = 1$$

for each $b$ and $\sigma > 0$. Such functions were used by Gabor [96] for *windowing* signals and are called *Gabor windows*.

Gabor's idea was to multiply $f(t)$, the signal of interest, by the window $g_\sigma(t - b)$ and then to take the Fourier transform, obtaining the *short-time Fourier transform* (STFT)

$$G_b^\sigma(\omega) = \int_{-\infty}^{\infty} f(t)g_\sigma(t - b)e^{i\omega t}dt.$$

Since $g_\sigma(t - b)$ falls off to zero on either side of $t = b$, multiplying by this window essentially restricts the signal to a neighborhood of $t = b$. The STFT then measures the frequency content of the signal, near the time $t = b$. The STFT therefore performs a *time-frequency analysis* of the signal.

We focus more tightly around the time $t = b$ by choosing a small value for $\sigma$. Because of the uncertainty principle, the Fourier transform of the window $g_\sigma(t - b)$ grows wider as $\sigma$ gets smaller; the *time-frequency window* remains constant [67]. This causes the STFT to involve greater blurring in the frequency domain. In short, to get good resolution in frequency, we

need to observe for a longer time; if we focus on a small time interval, we pay the price of reduced frequency resolution. This is unfortunate because when we focus on a short interval of time, it is to uncover a part of the signal that is changing within that short interval, which means it must have high frequency components within that interval. There is no reason to believe that the spacing is larger between those high frequencies we wish to resolve than between lower frequencies associated with longer time intervals. We would like to have the same resolving capability when focusing on a short time interval that we have when focusing on a longer one.

**The Wigner-Ville distribution:** In [143] Meyer describes Ville's approach to determining the instantaneous power spectrum of the signal, that is, the energy in the signal $f(t)$ that corresponds to time $t$ and frequency $\omega$. The goal is to find a function $W_f(t,\omega)$ having the properties

$$\int W_f(t,\omega)d\omega/2\pi = |f(t)|^2,$$

which is the total energy in the signal at time $t$, and

$$\int W_f(t,\omega)dt = |F(\omega)|^2,$$

which is the total energy in the Fourier transform at frequency $\omega$. Because these two properties do not specify a unique $W_f(t,\omega)$ two additional properties are usually required:

$$\int\int W_f(t,\omega)\overline{W_g(t,\omega)}dtd\omega/2\pi = |\int f(t)\overline{g(t)}dt|^2,$$

and for $f(t) = g_\sigma(t-b)\exp(i\alpha t)$

$$W_f(t,\omega) = 2\exp(-\sigma^{-2}(t-b)^2)\exp(-\sigma^2(\omega-\alpha)^2).$$

The *Wigner-Ville distribution* of $f(t)$, given by

$$WV_f(t,\omega) = \int_{-\infty}^{\infty} f(t+\frac{\tau}{2})\overline{f(t-\frac{\tau}{2})}\exp(-i\omega\tau)d\tau,$$

has all four of the desired properties. The Wigner-Ville distribution is always real-valued, but its values need not be nonnegative.

In [81] De Bruijn defines the *score* of a signal $f(t)$ to be $H(x,y;f,f)$, where

$$H(x,y;f_1,f_2) = 2\int_{-\infty}^{\infty} f_1(x+t)\overline{f_2(x-t)}e^{-4\pi iyt}dt.$$

**Exercise 1:** Relate the narrowband cross-ambiguity function to the De Bruijn's score and the Wigner-Ville distribution.

# Chapter 21

# Wavelets

The fantastic increase in computer power over the last few decades has made possible, even routine, the use of digital procedures for solving problems that were believed earlier to be intractible, such as the modeling of large-scale systems. At the same time, it has created new applications unimagined previously, such as medical imaging. In some cases the mathematical formulation of the problem is known and progress has come with the introduction of efficient computational algorithms, as with the Fast Fourier Transform. In other cases, the mathematics is developed, or perhaps rediscovered, as needed by the people involved in the applications. Only later it is realized that the theory already existed, as with the development of computerized tomography without Radon's earlier work on reconstruction of functions from their line integrals.

It can happen that applications give a theoretical field of mathematics a rebirth; such seems to be the case with *wavelets* [117]. Sometime in the 1980's researchers working on various problems in electrical engineering, quantum mechanics, image processing and elsewhere became aware that what the others were doing was related to their own work. As connections became established, similarities with the earlier mathematical theory of approximation in functional analysis were noticed. Meetings began to take place and a common language began to emerge around this reborn area, now called wavelets. There are a number of good books on wavelets, such as [123], [16] and [180].

Fourier analysis and synthesis concerns the decomposition, filtering, compressing and reconstruction of signals using complex exponential functions as the building blocks; wavelets provides a framework in which other building blocks, better suited to the problem at hand, can be used. As always, efficient algorithms provide the bridge between theory and practice.

Since their development in the 1980's wavelets have been used for many purposes. In the discussion to follow we focus on the problem of analyzing a

signal whose frequency composition is changing over time. As we saw in our discussion of the narrowband cross-ambiguity function in radar, the need for such *time-frequency* analysis has been known for quite a while. Other methods, such as Gabor's short time Fourier transform and the Wigner-Ville distribution, have also been considered for this purpose.

**The integral wavelet transform:** For real numbers $b$ and $a \neq 0$ the *integral wavelet transform* (IWT) of the signal $f(t)$ relative to the *basic wavelet* (or *mother* wavelet) $\psi(t)$ is

$$(W_\psi f)(b,a) = |a|^{-\frac{1}{2}} \int_{-\infty}^{\infty} f(t)\psi(\frac{t-b}{a})dt.$$

This function is also the *wideband cross-ambiguity function* in radar. The function $\psi(t)$ is also called a window function and, like Gaussian functions, it will be relatively localized in time. However, it must also have properties quite different from those of Gabor's Gaussian windows; in particular, we want

$$\int_{-\infty}^{\infty} \psi(t)dt = 0.$$

An example is the *Haar wavelet* $\psi_{Haar}(t)$ that has the value $+1$ for $0 \leq t < \frac{1}{2}$, $-1$ for $\frac{1}{2} \leq t < 1$ and zero otherwise.

As the scaling parameter $a$ grows larger the wavelet $\psi(t)$ grows wider, so choosing a small value of the scaling parameter permits us to focus in a neighborhood of the time $t = b$. The IWT then registers the contribution to $f(t)$ made by components with features on the scale determined by $a$, in the neightborhood of $t = b$. Calculations involving the uncertainty principle reveal that the IWT provides a flexible time-frequency window that narrows when we observe high frequency components and widens for lower frequencies [67].

Given the integral wavelet transform $(W_\psi f)(b,a)$ it is natural to ask how we might recover the signal $f(t)$. The following inversion formula answers that question: at points $t$ where $f(t)$ is continuous we have

$$f(t) = \frac{1}{C_\psi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (W_\psi f)(b,a)\psi(\frac{t-b}{a})\frac{da}{a^2}db,$$

with

$$C_\psi = \int_{-\infty}^{\infty} \frac{|\Psi(\omega)|^2}{|\omega|}d\omega$$

for $\Psi(\omega)$ the Fourier transform of $\psi(t)$.

**Wavelet series expansions:** The Fourier series expansion of a function $f(t)$ on a finite interval is a representation of $f(t)$ as a sum of orthogonal

complex exponentials. Localized alterations in $f(t)$ affect every one of the components of this sum. Wavelets, on the other hand, can be used to represent $f(t)$ so that localized alterations in $f(t)$ affect only a few of the components of the wavelet expansion. The simplest example of a wavelet expansion is with respect to the Haar wavelets.

**Exercise 1:** Let $w(t) = \psi_{Haar}(t)$. Show that the functions $w_{jk}(t) = w(2^j t - k)$ are mutually orthogonal on the interval $[0, 1]$, where $j = 0, 1, ...$ and $k = 0, 1, ..., 2^j - 1$.

These functions $w_{jk}(t)$ are the *Haar wavelets*. Every continuous function $f(t)$ defined on $[0, 1]$ can be written as

$$f(t) = c_0 + \sum\nolimits_{j=0}^{\infty} \sum\nolimits_{k=0}^{2^j-1} c_{jk} w_{jk}(t)$$

for some choice of $c_0$ and the $c_{jk}$. Notice that the *support of the function* $w_{jk}(t)$, the interval on which it is nonzero, gets smaller as $j$ increases. Therefore, the components corresponding to higher values of $j$ in the Haar expansion of $f(t)$ come from features that are localized in the variable $t$; such features are transients that live for only a short time. Such transient components affect all of the Fourier coefficients but only those Haar wavelet coefficients corresponding to terms supported in the region of the disturbance. This ability to isolate localized features is the main reason for the popularity of wavelet expansions.

The orthogonal functions used in the Haar wavelet expansion are themselves discontinuous, which presents a bit of a problem when we represent continuous functions. Wavelets that are themselves continuous, or better still, differentiable, should do a better job representing smooth functions.

We can obtain other wavelet series expansions by selecting a basic wavelet $\psi(t)$ and defining $\psi_{jk}(t) = 2^{j/2}\psi(2^j t - k)$, for integers $j$ and $k$. We then say that the function $\psi(t)$ is an *orthogonal wavelet* if the family $\{\psi_{jk}\}$ is an orthonormal basis for the space of square-integrable functions on the real line, the Hilbert space $L^2(R)$. This means that for every such $f(t)$ there are coefficients $c_{jk}$ so that

$$f(t) = \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} c_{jk} \psi_{jk}(t),$$

with convergence in the mean-square sense. The coefficients $c_{jk}$ are found using the IWT:

$$c_{jk} = (W_\psi f)(\frac{k}{2^j}, \frac{1}{2^j}).$$

It is also of interest to consider wavelets $\psi$ for which $\{\psi_{jk}\}$ form a basis, but not an orthogonal one, or, more generally, form a *frame*, in which the series representations of $f(t)$ need not be unique.

As with Fourier series, wavelet series expansion permits the filtering of certain components, as well as signal compression. In the case of Fourier series, we might attribute high frequency components to noise and achieve a smoothing by setting to zero the coefficients associated with these high frequencies. In the case of wavelet series expansions, we might attribute to noise localized small-scale disturbances and remove them by setting to zero the coefficients corresponding to the appropriate $j$ and $k$. For both Fourier and wavelet series expansions we can achieve compression by ignoring those components whose coefficients are below some chosen level.

**Multiresolution analysis:** One way to study wavelet series expansions is through *multiresolution analysis* (MRA). Let us begin with an example involving bandlimited functions. This example is called the *Shannon* MRA.

Let $V_0$ be the collection of functions $f(t)$ whose Fourier transform $F(\omega)$ is zero for $|\omega| > \pi$; so $V_0$ is the collection of $\pi$-bandlimited functions. Let $V_1$ be the collection of functions $f(t)$ whose Fourier transform $F(\omega)$ is zero for $|\omega| > 2\pi$; so $V_1$ is the collection of $2\pi$-bandlimited functions. In general, for each integer $j$, let $V_j$ be the collection of functions $f(t)$ whose Fourier transform $F(\omega)$ is zero for $|\omega| > 2^j\pi$; so $V_j$ is the collection of $2^j\pi$-bandlimited functions.

**Exercise 2:** Show that if the function $f(t)$ is in $V_j$ then the function $g(t) = f(2t)$ is in $V_{j+1}$.

We then have a nested sequence of sets of functions $\{V_j\}$, with $V_j \subseteq V_{j+1}$ for each integer $j$. The intersection of all the $V_j$ is the set containing only the zero function. Every function in $L^2(R)$ is arbitrarily close to a function in at least one of the sets $V_j$; more mathematically, we say that the union of the $V_j$ is dense in $L^2(R)$. In addition, we have $f(t)$ in $V_j$ if and only if $g(t) = f(2t)$ is in $V_{j+1}$. In general, such a collection of sets of functions is called a *multiresolution analysis* for $L^2(R)$. Once we have a MRA for $L^2(R)$ how do we get a wavelet series expansion?

A function $\phi(t)$ is called a *scaling function* or sometimes the *father wavelet* for the MRA if the collection of integer translates $\{\phi(t-k)\}$ forms a basis for $V_0$ (more precisely, a Riesz basis). Then, for each fixed $j$, the functions $\phi_{jk}(t) = \phi(2^j t - k)$, for integer $k$, will form a basis for $V_j$. In the case of the Shannon MRA the scaling function is $\phi(t) = \frac{\sin \pi t}{\pi t}$. But how do we get a basis for all of $L^2(R)$?

**The Haar multiresolution analysis:** To see how to proceed, it is helpful to return to the Haar wavelets. Let $\phi_{Haar}(t)$ be the function that has the value $+1$ for $0 \leq t < 1$ and zero elsewhere. Let $V_0$ be the collection of all functions in $L^2(R)$ that are linear combinations of integer translates of $\phi(t)$;

that is, all functions $f(t)$ that are constant on intervals of the form $[k, k+1)$, for all integers $k$. Now $V_1$ is the collection of all functions $g(t)$ of the form $g(t) = f(2t)$, for some $f(t)$ in $V_0$. Therefore, $V_1$ consists of all functions in $L^2(R)$ that are constant on intervals of the form $[k/2, (k+1)/2)$.

Every function in $V_0$ is also in $V_1$ and every function $g(t)$ in $V_1$ can be written uniquely as a sum of a function $f(t)$ in $V_0$ and a function $h(t)$ in $V_1$ that is orthogonal to every function in $V_0$. For example, the function $g(t)$ that takes the value $+3$ for $0 \leq t < 1/2$, $-1$ for $1/2 \leq t < 1$ and zero elsewhere can be written as $g(t) = f(t) + h(t)$ where $h(t)$ has the value $+2$ for $0 \leq t < 1/2$, $-2$ for $1/2 \leq t < 1$ and zero elsewhere, and $f(t)$ takes the value $+1$ for $0 \leq t < 1$ and zero elsewhere. Clearly, $h(t)$, which is twice the Haar wavelet function, is orthogonal to all functions in $V_0$.

**Exercise 3:** Show that the function $f(t)$ can be written uniquely as $f(t) = d(t) + e(t)$, where $d(t)$ in $V_{-1}$ and $e(t)$ is in $V_0$ and is orthogonal to every function in $V_{-1}$. Relate the function $e(t)$ to the Haar wavelet function.

**Wavelets and multiresolution analysis:** To get an orthogonal wavelet expansion from a general MRA we write the set $V_1$ as the direct sum $V_1 = V_0 \oplus W_0$, so every function $g(t)$ in $V_1$ can be uniquely written as $g(t) = f(t) + h(t)$, where $f(t)$ is a function in $V_0$ and $h(t)$ is a function in $W_0$, with $f(t)$ and $h(t)$ orthogonal. Since the scaling function or father wavelet $\phi(t)$ is in $V_1$ it can be written as

$$\phi(t) = \sum_{k=-\infty}^{\infty} p_k \phi(2t - k), \tag{21.1}$$

for some sequence $\{p_k\}$ called the *two-scale sequence* for $\phi(t)$. This most important identity is the *scaling relation* for the father wavelet. The mother wavelet is defined using a similar expression

$$\psi(t) = \sum_k (-1)^k \overline{p_{1-k}} \phi(2t - k). \tag{21.2}$$

We define

$$\phi_{jk}(t) = 2^{j/2} \phi(2^j t - k) \tag{21.3}$$

and

$$\psi_{jk}(t) = 2^{j/2} \psi(2^j t - k). \tag{21.4}$$

The collection $\{\psi_{jk}(t),\ -\infty < j, k < \infty\}$ then forms an orthogonal wavelet basis for $L^2(R)$. For the Haar MRA the two-scale sequence is $p_0 = p_1 = 1$ and $p_k = 0$ for the rest.

**Exercise 4:** Show that the two-scale sequence $\{p_k\}$ has the properties

$$p_k = 2 \int \phi(t)\overline{\phi(2t-k)}dt;$$

$$\sum_{k=-\infty}^{\infty} p_{k-2m}\overline{p_k} = 0,$$

for $m \neq 0$ and equals two when $m = 0$.

**Signal processing using wavelets:** Once we have an orthogonal wavelet basis for $L^2(R)$ we can use the basis to represent and process a signal $f(t)$. Suppose, for example, that $f(t)$ is bandlimited but essentially zero for $t$ not in $[0, 1]$ and we have samples $f(\frac{k}{M})$, $k = 0, ..., M$. We assume that the sampling rate $\Delta = \frac{1}{M}$ is faster than the Nyquist rate so that the Fourier transform of $f(t)$ is zero outside, say, the interval $[0, 2\pi M]$. Roughly speaking, the $W_j$ component of $f(t)$, given by

$$g_j(t) = \sum_{k=0}^{2^j-1} \beta_k^j \psi_{jk}(t),$$

with $\beta_k^j = \langle f(t), \psi_{jk}(t)\rangle$, corresponds to the components of $f(t)$ with frequencies $\omega$ between $2^{j-1}$ and $2^j$. For $2^j > 2\pi M$ we have $\beta_k^j = 0$, so $g_j(t) = 0$. Let $J$ be the smallest integer greater than $\log_2(2\pi) + \log_2(M)$. Then $f(t)$ is in the space $V_J$ and has the expansion

$$f(t) = \sum_{k=0}^{2^J-1} \alpha_k^J \phi_{Jk}(t),$$

for $\alpha_k^J = \langle f(t), \phi_{Jk}(t)\rangle$. It is common practice, but not universally approved, to take $M = 2^J$ and to estimate the $\alpha_k^J$ by the samples $f(\frac{k}{M})$. Once we have the sequence $\{\alpha_k^J\}$ we can begin the decomposition of $f(t)$ into components in $V_j$ and $W_j$ for $j < J$. As we shall see, the algorithms for the decomposition and subsequent reconstruction of the signal are quite similar to the FFT.

**Decomposition and reconstruction:** The decomposition and reconstruction algorithms both involve the equation

$$\sum_k a_k^j \phi_{jk} = \sum_m a_m^{j-1}\phi_{(j-1),m} + b_m^{j-1}\psi_{(j-1),m}\,; \tag{21.5}$$

in the decomposition step we know the $\{a_k^j\}$ and want the $\{a_m^{j-1}\}$ and $\{b_m^{j-1}\}$, while in the reconstruction step we know the $\{a_m^{j-1}\}$ and $\{b_m^{j-1}\}$ and want the $\{a_k^j\}$.

Using equations (21.1) and (21.3) we obtain

$$\phi_{(j-1),l} = 2^{-1/2} \sum_k p_k \phi_{j,(k+2l)} = 2^{-1/2} \sum_k p_{k-2l} \phi_{jk}; \tag{21.6}$$

using equations (21.2), (21.3) and (21.4) we get

$$\psi_{(j-1),l} = 2^{-1/2} \sum_k (-1)^k \overline{p_{1-k+2l}} \phi_{jk}. \tag{21.7}$$

Therefore

$$\langle \phi_{jk}, \phi_{(j-1),l} \rangle = 2^{-1/2} \overline{p_{k-2l}}; \tag{21.8}$$

this comes from substituting $\phi_{(j-1),l}$ as in equation (21.6) into the second term in the inner product. Similarly, we have

$$\langle \phi_{jk}, \psi_{(j-1),l} \rangle = 2^{-1/2} (-1)^k p_{1-k+2l}. \tag{21.9}$$

These relationships are then used to derive the decomposition and reconstruction algorithms.

*The decomposition step:* To find $a_l^{j-1}$ we take the inner product of both sides of equation (21.5) with the function $\phi_{(j-1),l}$. Using equation (21.8) and the fact that $\phi_{(j-1),l}$ is orthogonal to all the $\phi_{(j-1),m}$ except for $m = l$ and is orthogonal to all the $\psi_{(j-1),m}$, we obtain

$$2^{-1/2} \sum_k a_k^j \overline{p_{k-2l}} = a_l^{j-1};$$

similarly, using equation (21.9), we get

$$2^{-1/2} \sum_k a_k^j (-1)^k p_{1-k+2l} = b_l^{j-1}.$$

The decomposition step is to apply these two equations to get the $\{a_l^{j-1}\}$ and $\{b_l^{j-1}\}$ from the $\{a_k^j\}$.

*The reconstruction step:* Now we use equations (21.6) and (21.7) to substitute into the right hand side of equation (21.5). Combining terms, we get

$$a_k^j = 2^{-1/2} \sum_l a_l^{j-1} p_{k-2l} + b_l^{j-1} (-1)^k \overline{p_{1-k+2l}}.$$

This takes us from the $\{a_l^{j-1}\}$ and $\{b_l^{j-1}\}$ to the $\{a_k^j\}$.

We have assumed that we have already obtained the scaling function $\phi(t)$ with the property that $\{\phi(t-k)\}$ is an orthogonal basis for $V_0$. But how do we actually obtain such functions?

**Generating the scaling function:** The scaling function $\phi(t)$ is generated from the two-scale sequence $\{p_k\}$ using the following iterative procedure. Start with $\phi_0(t) = \phi_{Haar}(t)$, the Haar scaling function that is one on $[0,1]$ and zero elsewhere. Now, for each $n = 1, 2, ...$ define

$$\phi_n(t) = \sum_{k=-\infty}^{\infty} p_k \phi_{n-1}(2t-k).$$

Provided that the sequence $\{p_k\}$ has certain properties to be discussed below, this sequence of functions converges and the limit is the desired scaling function.

The properties of $\{p_k\}$ that are needed can be expressed in terms of properties of the function

$$P(z) = \frac{1}{2} \sum_{k=-\infty}^{\infty} p_k z^k.$$

For the Haar MRA this function is $P(z) = \frac{1}{2}(1+z)$. We require that

**1.** $P(1) = 1$;

**2.** $|P(e^{i\theta})|^2 + |P(e^{i(\theta+\pi)})|^2 = 1$ for $0 \le \theta \le \pi$;

and

**3.** $|P(e^{i\theta})| > 0$ for $-\frac{\pi}{2} \le \theta \le \frac{\pi}{2}$.

**Generating the two-scale sequence:** The final piece of the puzzle is the generation of the sequence $\{p_k\}$ itself, or, equivalently, finding a function $P(z)$ with the properties listed above. The following example, also used in [16], illustrates Daubechies' method.

We begin with the identity

$$\cos^2 \frac{\theta}{2} + \sin^2 \frac{\theta}{2} = 1$$

and then raise both sides to an odd power $n = 2N-1$. Here we use $N = 2$, obtaining

$$1 = \cos^6 \frac{\theta}{2} + 3\cos^4 \frac{\theta}{2} \sin^2 \frac{\theta}{2}$$
$$+ \cos^6 \frac{(\theta+\pi)}{2} + 3\cos^4 \frac{(\theta+\pi)}{2} \sin^2 \frac{(\theta+\pi)}{2}.$$

We then let

$$|P(e^{i\theta})|^2 = \cos^6 \frac{\theta}{2} + 3\cos^4 \frac{\theta}{2} \sin^2 \frac{\theta}{2},$$

so that

$$|P(e^{i\theta})|^2 + |P(e^{i(\theta+\pi)})|^2 = 1$$

for $0 \leq \theta \leq \pi$. Now we have to find $P(e^{i\theta})$.

Writing

$$|P(e^{i\theta})|^2 = \cos^4 \frac{\theta}{2}[\cos^2 \frac{\theta}{2} + 3\sin^2 \frac{\theta}{2}],$$

we have

$$P(e^{i\theta}) = \cos^2 \frac{\theta}{2}[\cos \frac{\theta}{2} + \sqrt{3}i \sin \frac{\theta}{2}]e^{i\alpha(\theta)},$$

where the real function $\alpha(\theta)$ is arbitrary. Selecting $\alpha(\theta) = 3\frac{\theta}{2}$ we get

$$P(e^{i\theta}) = p_0 + p_1 e^{i\theta} + p_2 e^{2i\theta} + p_3 e^{3i\theta},$$

for

$$p_0 = \frac{1+\sqrt{3}}{4};$$

$$p_1 = \frac{3+\sqrt{3}}{4};$$

$$p_2 = \frac{3-\sqrt{3}}{4};$$

$$p_3 = \frac{1-\sqrt{3}}{4};$$

and all the other coefficients are zero. The resulting Daubechies' wavelet is compactly supported and continuous, but not differentiable [16]. Figure 21.1 shows the scaling function and mother wavelet for $N = 2$. When larger values of $N$ are used the resulting wavelet, often denoted $\psi_N(t)$, which is again compactly supported, has approximately $N/5$ continuous derivatives.

These notions extend to non-orthogonal wavelet bases and to frames. Algorithms similar to the fast Fourier transform provide the wavelet decomposition and reconstruction of signals. The recent text by Boggess and Narcowich [16] is a nice introduction to this fast-growing area; the more advanced book by Chui [67] is also a good source. Wavelets in the context of Riesz bases and frames are discussed in Christensen's book [66].

**Wavelets and filter banks:** In [172] Strang and Nguyen take a somewhat different approach to wavelets, emphasizing the role of filters and matrices. To illustrate one of their main points we consider the two-point moving average filter.

The two-point moving average filter transforms an input sequence $x = \{x(n)\}$ to output $y = \{y(n)\}$ with $y(n) = \frac{1}{2}x(n) + \frac{1}{2}x(n-1)$. The filter

$h = \{h(k)\}$ has $h(0) = h(1) = \frac{1}{2}$ and all the remaining $h(n)$ are zero. This filter is a *finite impulse response* (FIR) low-pass filter and is not invertible; the input sequence with $x(n) = (-1)^n$ has output zero. Similarly, the two-point moving difference filter $g = \{g(k)\}$ with $g(0) = \frac{1}{2}$, $g(1) = -\frac{1}{2}$ and the rest zero, is a FIR high-pass filter, also not invertible. However, if we perform these filters in parallel, as a filter bank, no information is lost and the input can be completely reconstructed, with a unit delay. In addition, the outputs of the two filters contain redundancy that can be removed by *decimation*, which is taken here to mean *downsampling*, that is, throwing away every other term of a sequence.

The authors treat the more general problem of obtaining perfect reconstruction of the input from the output of a filter bank of low- and high-pass filters followed by downsampling. The properties that must be required of the filters are those we encountered earlier with regard to the two-scale sequences for the father and mother wavelets. When the filter operations are construed as matrix multiplications the decomposition and reconstruction algorithms become matrix factorizations.

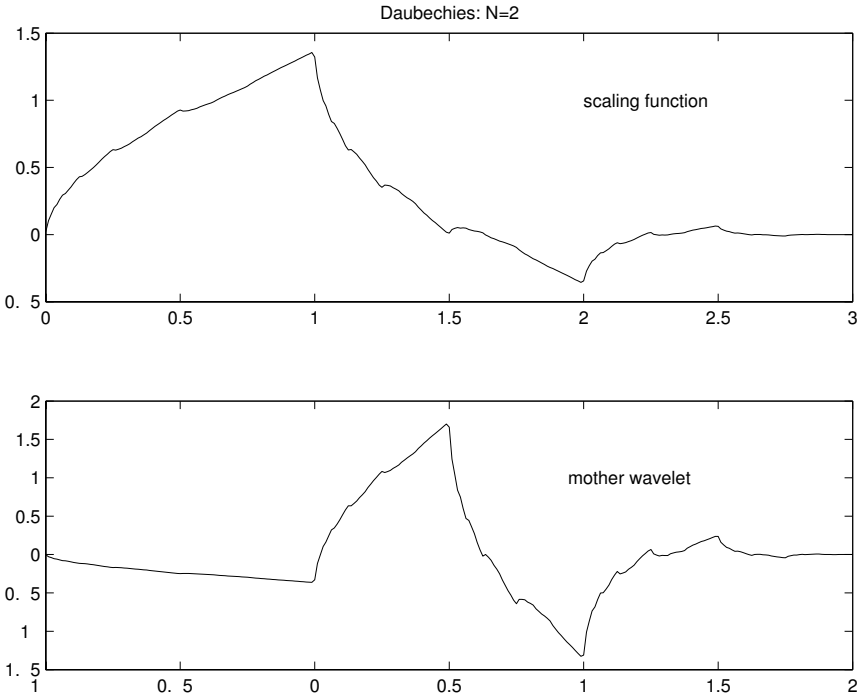



Figure 21.1: Daubechies' scaling function and mother wavelet for $N = 2$.

# Chapter 22

# The FT in Higher Dimensions

The Fourier transform is also defined for functions of several real variables $f(x_1, ..., x_N) = f(\mathbf{x})$. The multidimensional FT arises in image processing, scattering, transmission tomography, and many other areas.

We adopt the usual vector notation that $\omega$ and $\mathbf{x}$ are $N$-dimensional real vectors. We say that $F(\omega)$ is the *N-dimensional Fourier transform* of the possibly complex-valued function $f(\mathbf{x})$ if the following relation holds:

$$F(\omega) = \int_{-\infty}^{\infty} ... \int_{-\infty}^{\infty} f(\mathbf{x}) e^{i\omega \cdot \mathbf{x}} d\mathbf{x},$$

where $\omega \cdot \mathbf{x}$ denotes the vector dot product and $d\mathbf{x} = dx_1 dx_2 ... dx_N$. In most cases we then have

$$f(\mathbf{x}) = \int_{-\infty}^{\infty} ... \int_{-\infty}^{\infty} F(\omega) e^{-i\omega \cdot \mathbf{x}} d\omega / (2\pi)^N;$$

we describe this by saying that $f(\mathbf{x})$ is the *inverse Fourier transform* of $F(\omega)$.

Consider the FT of a function of two variables $f(x, y)$:

$$F(\alpha, \beta) = \int \int f(x, y) e^{i(x\alpha + y\beta)} dx dy.$$

We convert to polar coordinates using $(x, y) = r(\cos\theta, \sin\theta)$ and $(\alpha, \beta) = \rho(\cos\omega, \sin\omega)$. Then

$$F(\rho, \omega) = \int_0^{\infty} \int_{-\pi}^{\pi} f(r, \theta) e^{ir\rho\cos(\theta - \omega)} r dr d\theta. \tag{22.1}$$

Say that a function $f(x,y)$ of two variables is a *radial* function if $x^2+y^2 = x_1^2+y_1^2$ implies $f(x,y) = f(x_1,y_1)$, for all points $(x,y)$ and $(x_1,y_1)$; that is, $f(x,y) = g(\sqrt{x^2+y^2})$ for some function $g$ of one variable.

**Exercise 1:** Show that if $f$ is radial then its FT $F$ is also radial. Find the FT of the radial function $f(x,y) = \frac{1}{\sqrt{x^2+y^2}}$.

**Hints:** Insert $f(r,\theta) = g(r)$ in equation (22.1) to obtain

$$F(\rho,\omega) = \int_0^\infty \int_{-\pi}^{\pi} g(r) e^{ir\rho\cos(\theta-\omega)} r dr d\theta$$

or

$$F(\rho,\omega) = \int_0^\infty r g(r) [\int_{-\pi}^{\pi} e^{ir\rho\cos(\theta-\omega)} d\theta] dr. \tag{22.2}$$

Show that the inner integral is independent of $\omega$ and then use the fact that

$$\int_{-\pi}^{\pi} e^{ir\rho\cos\theta} d\theta = 2\pi J_0(r\rho),$$

with $J_0$ the 0-th order Bessel function, to get

$$F(\rho,\omega) = H(\rho) = 2\pi \int_0^\infty r g(r) J_0(r\rho) dr. \tag{22.3}$$

The function $H(\rho)$ is called the *Hankel transform* of $g(r)$. Summarizing, we say that if $f(x,y)$ is a radial function obtained using $g$ then its Fourier transform $F(\alpha,\beta)$ is also a radial function, obtained using the Hankel transform of $g$.

# Chapter 23

# Characteristic Functions

The Fourier transform shows up in probability theory in the guise of the *characteristic function* of a random variable. The characteristic function is related to, but more general than, the moment-generating function and serves much the same purposes.

A real-valued random variable $X$ is said to have the probability density function (pdf) $f(x)$ if, for any interval $[a, b]$, the probability that $X$ takes its value within this interval is given by the integral $\int_a^b f(x)dx$. To be a pdf $f(x)$ must be nonnegative and $\int_{-\infty}^{\infty} f(x)dx = 1$. The *characteristic function* of $X$ is then

$$F(\omega) = \int_{-\infty}^{\infty} f(x)e^{ix\omega}dx.$$

The formulas for differentiating the Fourier transform are quite useful in determining the moments of a random variable.

The *expected value* of $X$ is

$$E(X) = \int_{-\infty}^{\infty} xf(x)dx,$$

and for any real-valued function $g(x)$ the expected value of the random variable $g(X)$ is

$$E(g(X)) = \int_{-\infty}^{\infty} g(x)f(x)dx.$$

The $n$-th moment of $X$ is

$$E(X^n) = \int_{-\infty}^{\infty} x^n f(x)dx;$$

the *variance* of $X$ is then $\text{var}(X) = E(X^2) - E(X)^2$. It follows, therefore, that the $n$-th moment of the random variable $X$ is given by

$$E(X^n) = (i)^n F^{(n)}(0).$$

If we have $N$ real-valued random variables $X_1, ..., X_N$ their *joint probability density function* is $f(x_1, ..., x_N) \geq 0$ having the property that, for any intervals $[a_1, b_1], ..., [a_N, b_N]$, the probability that $X_n$ takes its value within $[a_n, b_n]$, for each $n$, is given by the multiple integral

$$\int_{a_1}^{b_1} \cdots \int_{a_N}^{b_N} f(x_1, ..., x_N) dx_1 \cdots dx_N.$$

The joint moments are then

$$E(X_1^{m_1} \cdots X_N^{m_N}) = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} x_1^{m_1} \cdots x_N^{m_N} f(x_1, ..., x_N) dx_1 \cdots dx_N.$$

The joint moments can be calculated by evaluating at zero the partial derivatives of the characteristic function of the joint pdf.

The random variables are said to be *independent* if

$$f(x_1, ..., x_N) = f(x_1) \cdots f(x_N),$$

where, in keeping with the convention used in the probability literature, $f(x_n)$ denotes the pfd of the random variable $X_n$.

If $X$ and $Y$ are independent random variables with probability density functions $f(x)$ and $g(y)$ then the probability density function for the random variable $Z = X + Y$ is $(f * g)(z)$, the convolution of $f$ and $g$. To see this, we first calculate the cumulative distribution function

$$H(z) = \text{Prob}\,(X + Y \leq z),$$

which is

$$H(z) = \int_{x=-\infty}^{+\infty} \int_{y=-\infty}^{z-x} f(x) g(y) dy dx.$$

Using the change of variable $t = x + y$, we get

$$H(z) = \int_{x=-\infty}^{+\infty} \int_{t=-\infty}^{z} f(x) g(t - x) dt dx.$$

The pdf for the random variable $Z$ is $h(z) = H'(z)$, the derivative of $H(z)$. Differentiating the inner integral with respect to $z$ we obtain

$$h(z) = \int_{x=-\infty}^{+\infty} f(x) g(z - x) dx;$$

therefore $h(z) = (f * g)(z)$. It follows that the characteristic function for the random variable $Z = X + Y$ is the product of the characteristic functions for $X$ and $Y$.

# Chapter 24

# The Hilbert Transform

We encountered the Hilbert transform for sequences in our discussion of analytic functions and for functions in one of the exercises earlier. Now we take a closer look. In some contexts, such as harmonic analysis, the Hilbert transform is called the *conjugate function* [125]

### The Hilbert transform of periodic $f(t)$:

The *Hilbert transform* (HT) of the function $f(t) = \cos(\omega t)$ is the function $\sin(\omega t)$. The HT of $\sin(\omega t)$ is $-\cos(\omega t)$, so the HT can be viewed as performing integration; for this reason it is sometimes called a *quadrature filter*.

If $f(t)$ is a $2\pi$-periodic function with Fourier series expansion

$$f(t) = \sum_{n=-\infty}^{+\infty} a_n \exp(int),$$

then the HT of $f(t)$, denoted $HT_f(t)$, is formed by multiplying the coefficients $a_n$ by $-i$, for $n > 0$, by $i$ for $n < 0$ and by zero for $n = 0$. Therefore, we have

$$HT_f(t) = i \sum_{n=-\infty}^{-1} a_n \exp(int) - i \sum_{n=1}^{+\infty} a_n \exp(int).$$

Since

$$\cos(nt) = \frac{1}{2}\exp(-int) + \frac{1}{2}\exp(int)$$

we see that its Hilbert transform is

$$i\frac{1}{2}\exp(-int) - i\frac{1}{2}\exp(int),$$

which is $\sin(nt)$.

One way to motivate the HT is to connect the Fourier series representations with the Laurent series obtained by replacing $\exp(int)$ with $z^n$. The Fourier series for the function $g(t) = f(t) + iHT_f(t)$ has terms only for positive values of $n$. Therefore, when we replace $\exp(int)$ with $z^n$, we get only positive powers of the variable $z$, so the Laurent series becomes a Taylor series, so becomes analytic in a disk centered at zero. We can therefore connect the Fourier theory with the theory of analytic functions via the HT.

## The Hilbert transform for non-periodic $f(t)$:

For non-periodic functions $f(t)$ we can view the HT as operating on the Fourier transform of $f(t)$ instead of on its Fourier coefficients. Specificially, let $f(t)$ have Fourier transform $F(\omega)$. Then the HT of $f(t)$ has for its Fourier transform the function $G(\omega)$ that is equal to $-iF(\omega)$ for $\omega > 0$, to $iF(\omega)$ for $\omega < 0$ and equal to zero for $\omega = 0$. Recall that the function $\text{sgn}(\omega)$ is $+1$ for $\omega > 0$, $-1$ for $\omega < 0$ and zero for $\omega = 0$. Therefore, $HT_f(t)$, the HT of $f(t)$, has for its Fourier transform the function $G(\omega) = F(\omega)\text{sgn}(\omega)$. In the $t$ domain the HT is obtained by convolving $f(t)$ with the inverse Fourier transform of $\text{sgn}(\omega)$, which is the function $h(t) = \frac{1}{\pi t}$:

$$HT_f(t) = \frac{1}{\pi}\int_{-\infty}^{+\infty} \frac{f(\tau)}{t-\tau} d\tau.$$

So this is what the HT is; but what is it used for and how does it arise?

While the HT may seem to be a fairly obscure notion, the function $\text{sgn}(\omega)$ is quite common; the HT often arises in applications as a result of the use of the sgn function.

## The Hilbert transform of real-valued functions $f(t)$:

Suppose that $f(t)$ is a real-valued function. Then its Fourier transform $F(\omega)$ is conjugate-symmetric. Therefore, the values $F(\omega)$ for $\omega < 0$ are redundant and $f(t)$ is completely determined from the values of $F(\omega)$ for $\omega > 0$; we may therefore, wish to work solely with the positive $\omega$ values. Suppose we define $Z(\omega) = 0$ for $\omega \leq 0$ and

$$Z(\omega) = 2F(\omega)$$

for $\omega > 0$. Then since $Z(\omega)$ is not conjugate-symmetric, its inverse Fourier transform is not real. Its real part turns out to be the original $f(t)$ and its imaginary part is the HT of $f$.

Viewed another way, given a real-valued function $f(t)$ we seek a second real-valued function $g(t)$ so that the complex-valued function $z(t) = f(t) + ig(t)$ has Fourier transform $Z(\omega)$ that equals $2F(\omega)$ for $\omega > 0$ and is zero otherwise; then $g(t)$ is the HT of $f(t)$.

### The Hilbert transform of causal functions $f(t)$:

Another way in which the HT arises is in the context of *causal* functions. Say that complex-valued $f(t)$ is causal if $f(t) = 0$ for $t \leq 0$. Then the real and imaginary parts of its Fourier transform are $R(\omega)$ and $HT_R(\omega)$; that is, the imaginary part is the HT of the real part.

# Chapter 25

# The Fast Fourier Transform

A fundamental problem in signal processing is to estimate finitely many values of the function $F(\omega)$ from finitely many values of its (inverse) Fourier transform, $f(t)$. As we have seen, the DFT arises in several ways in that estimation effort. The *fast Fourier transform* (FFT), discovered in 1965 by Cooley and Tukey, is an important and efficient algorithm for calculating the vector DFT [74]. John Tukey has been quoted as saying that his main contribution to this discovery was the firm and often voiced belief that such an algorithm must exist.

To illustrate the main idea behind the FFT consider the problem of evaluating a real polynomial $P(x)$ at a point, say $x = c$: let the polynomial be

$$P(x) = a_0 + a_1 x + a_2 x^2 + ... + a_{2K} x^{2K},$$

where $a_{2K}$ might be zero. Performing the evaluation efficiently by Horner's method,

$$P(c) = (((a_{2K} c + a_{2K-1})c + a_{2K-2})c + a_{2K-3})c + ...,$$

requires $2K$ multiplications, so the complexity is on the order of the degree of the polynomial being evaluated. But suppose we also want $P(-c)$. We can write

$$P(x) = (a_0 + a_2 x^2 + ... + a_{2K} x^{2K}) + x(a_1 + a_3 x^2 + ... + a_{2K-1} x^{2K-2})$$

or

$$P(x) = Q(x^2) + xR(x^2).$$

Therefore we have $P(c) = Q(c^2) + cR(c^2)$ and $P(-c) = Q(c^2) - cR(c^2)$. If we evaluate $P(c)$ by evaluating $Q(c^2)$ and $R(c^2)$ separately, one more

multiplication gives us $P(-c)$ as well. The FFT is based on repeated use of this idea, which turns out to be more powerful when we are using complex exponentials, because of their periodicity.

Say the data are the samples are $\{f(n\Delta), n = 1, ..., N\}$, where $\Delta > 0$ is the sampling increment or sampling spacing.

The DFT estimate of $F(\omega)$ is the function $F_{DFT}(\omega)$, defined for $\omega$ in $[-\pi/\Delta, \pi/\Delta]$, and given by

$$F_{DFT}(\omega) = \Delta \sum_{n=1}^{N} f(n\Delta) e^{in\Delta\omega}.$$

The DFT estimate $F_{DFT}(\omega)$ is data consistent; its inverse Fourier transform value at $t = n\Delta$ is $f(n\Delta)$ for $n = 1, ..., N$. The DFT is sometimes used in a slightly more general context in which the coefficients are not necessarily viewed as samples of a function $f(t)$.

Given the complex $N$-dimensional column vector $\mathbf{f} = (f_0, f_1, ..., f_{N-1})^T$ define the $DFT$ of vector $\mathbf{f}$ to be the function $DFT_{\mathbf{f}}(\omega)$, defined for $\omega$ in $[0, 2\pi)$, given by

$$DFT_{\mathbf{f}}(\omega) = \sum_{n=0}^{N-1} f_n e^{in\omega}.$$

Let $\mathbf{F}$ be the complex $N$-dimensional vector $\mathbf{F} = (F_0, F_1, ..., F_{N-1})^T$, where $F_k = DFT_{\mathbf{f}}(2\pi k/N), k = 0, 1, ..., N-1$. So the vector $\mathbf{F}$ consists of $N$ values of the function $DFT_{\mathbf{f}}$, taken at $N$ equispaced points $2\pi/N$ apart in $[0, 2\pi)$.

From the formula for $DFT_{\mathbf{f}}$ we have, for $k = 0, 1, ..., N-1$,

$$F_k = F(2\pi k/N) = \sum_{n=0}^{N-1} f_n e^{2\pi ink/N}. \tag{25.1}$$

To calculate a single $F_k$ requires $N$ multiplications; it would seem that to calculate all $N$ of them would require $N^2$ multiplications. However, using the FFT algorithm we can calculate vector $\mathbf{F}$ in approximately $N \log_2(N)$ multiplications.

Suppose that $N = 2M$ is even. We can rewrite equation(25.1) as follows:

$$F_k = \sum_{m=0}^{M-1} f_{2m} e^{2\pi i(2m)k/N} + \sum_{m=0}^{M-1} f_{2m+1} e^{2\pi i(2m+1)k/N},$$

or, equivalently,

$$F_k = \sum_{m=0}^{M-1} f_{2m} e^{2\pi imk/M} + e^{2\pi ik/N} \sum_{m=0}^{M-1} f_{2m+1} e^{2\pi imk/M}. \tag{25.2}$$

Note that if $0 \leq k \leq M-1$ then

$$F_{k+M} = \sum_{m=0}^{M-1} f_{2m} e^{2\pi imk/M} - e^{2\pi ik/N} \sum_{m=0}^{M-1} f_{2m+1} e^{2\pi imk/M}, \qquad (25.3)$$

so there is no additional computational cost in calculating the second half of the entries of $\mathbf{F}$, once we have calculated the first half. The FFT is the algorithm that results when take full advantage of the savings obtainable by splitting a DFT calculating into two similar calculations of half the size.

We assume now that $N = 2^L$. Notice that if we use equations (25.2) and (25.3) to calculate vector $\mathbf{F}$, the problem reduces to the calculation of two similar DFT evaluations, both involving half as many entries, followed by one multiplication for each of the $k$ between 0 and $M-1$. We can split these in half as well. The FFT algorithm involves repeated splitting of the calculations of DFTs at each step into two similar DFTs, but with half the number of entries, followed by as many multiplications as there are entries in either one of these smaller DFTs. We use recursion to calculate the cost $C(N)$ of computing $\mathbf{F}$ using this FFT method. From equation (25.2) we see that $C(N) = 2C(N/2) + (N/2)$. Applying the same reasoning to get $C(N/2) = 2C(N/4) + (N/4)$, we obtain

$$C(N) = 2C(N/2) + (N/2) = 4C(N/4) + 2(N/2) = ...$$

$$= 2^L C(N/2^L) + L(N/2) = N + L(N/2).$$

Therefore the cost required to calculate $\mathbf{F}$ is approximately $N \log_2 N$.

From our earlier discussion of discrete linear filters and convolution we see that the FFT can be used to calculate the periodic convolution (or even the non-periodic convolution) of finite length vectors.

Finally, let's return to the original context of estimating the Fourier transform $F(\omega)$ of function $f(t)$ from finitely many samples of $f(t)$. If we have $N$ equispaced samples we can use them to form the vector $\mathbf{f}$ as above and perform the FFT algorithm to get vector $\mathbf{F}$ consisting of $N$ values of the DFT estimate of $F(\omega)$. It may happen that we wish to calculate more than $N$ values of the DFT estimate, perhaps to produce a smooth looking graph. We can still use the FFT, but we must trick it into thinking we have more data that the $N$ samples we really have. We do this by *zero-padding*. Instead of creating the $N$-dimensional vector $\mathbf{f}$, we make a longer vector by appending, say, $J$ zeros to the data, to make a vector that has dimension $N + J$. The DFT estimate is still the same function of $\omega$, since we have only included new zero coefficients as fake data. But the FFT thinks we have $N + J$ data values, so it returns $N + J$ values of the DFT, at $N + J$ equispaced values of $\omega$ in $[0, 2\pi)$.

# Chapter 26

# Two Problems in Fourier Transform Estimation

It is often the case in remote sensing that what we want and what we can measure are related by Fourier transformation. Frequently one of the two functions has bounded support, so that the other one is band-limited. If our measurements are samples of a function of bounded support we shall say that we are solving a problem of Type One, while if the sampled function is band-limited we say the problem is of Type Two. As we shall see, these two types of problems are distinct and different techniques are required to solve them.

Throughout this chapter we let $F(\omega)$ be defined for $\omega \in [0, 2\pi]$, with

$$f(x) = \frac{1}{2\pi} \int_0^{2\pi} F(\omega) e^{-ix\omega} d\omega. \tag{26.1}$$

In applications $F(\omega)$ usually represents some physical object of limited extent. In problems of Type Two remote sensing has provided (usually noisy) values of $f(x)$ for finitely many $x$.

When algorithms are being developed and tested one often works with simulations. If the $F(\omega)$ to be simulated is specified analytically we may be able to compute values of $f(x)$ by performing the integrals in equation (26.1). It may be the case, however, that the integrals cannot be performed exactly or even that $F(\omega)$ is represented by a finite vector of samples. Estimating values of $f(x)$ in such cases becomes a problem of Type One. In the hyperspectral imaging problem discussed in a later chapter problems of both types must be solved.

When discussing problems of Type One in this chapter we shall assume that we have the values $F_n = F(2\pi n/N)$, $n = 0, 1, ..., N-1$ and wish to estimate $f(x)$ for certain values of $x$. When discussing problems of Type

Two in this chapter we shall assume, at first, that we have the values $f(m)$, $m = 0, ..., M-1$ and wish to estimate values of $F(\omega)$ and then allow the data to be $f(x_m)$, $m = 1, ..., M$, where the $x_m$ are arbitrary.

For problems of Type One it is tempting to take as our estimate of $f(x)$ what is perhaps the obvious choice, the function

$$\hat{f}(x) = \frac{1}{N} \sum_{n=0}^{N-1} F_n e^{-2\pi n x/N}. \tag{26.2}$$

and for problems of Type Two the estimate

$$\hat{F}(\omega) = \sum_{m=0}^{M-1} f(m) e^{im\omega}. \tag{26.3}$$

If, in the first case, we decide to estimate $f(x)$ only for the integer values $j = 0, ..., N-1$ then we get

$$\hat{f}(j) = \frac{1}{N} \sum_{n=0}^{N-1} F_n e^{-2\pi n j/N}, \tag{26.4}$$

which can be calculated using the Fast Fourier Transform. Similarly, if, in the second case, we decide to estimate $F(\omega)$ only for the values $\omega = \omega_k = 2\pi k/M$, $k = 0, ..., M-1$, we get

$$\hat{F}(\omega_k) = \sum_{m=0}^{M-1} f(m) e^{2\pi k m/M}, \tag{26.5}$$

The main theme of this chapter is that while these estimates may be obvious, they are not necessarily good choices.

**Exercise 1:** Consider the function $F(\omega)$ defined on the interval $[0, 2\pi]$ by $F(\omega) = 1$ for $\frac{\pi}{2} \leq \omega \leq \frac{3\pi}{2}$ and $F(\omega) = 0$ elsewhere. The inverse Fourier transform of $F(\omega)$ is $f(x) = \frac{1}{2}(\sin(\frac{\pi}{2}x))/(\frac{\pi}{2}x)$. Let $N$ be a positive power of two and let $b_n = F(\frac{2\pi}{N}(n-1)$, for $n = 1, 2, ..., N$. The FFT of the vector $\mathbf{b}$ has the entries

$$fft(\mathbf{b})_k = \sum_{n=1}^{N} b_n \exp(-i(n-1)(k-1)\frac{2\pi}{N}),$$

for $k = 1, 2, ..., N$. Use MATLAB or some similar computer package to compute and compare the values $f(k-1)$ and $\frac{1}{N} fft(\mathbf{b})_k$ for $k = 1, ..., N$. Repeat this exercise for different values of $N$.

**Problems of Type One:** Let us assume that $F(\omega)$ is Riemann integrable. For each $x$ we can approximate the integral in equation (26.1) by the Riemann sum

$$rs(x;N) = \frac{1}{N}\sum_{n=0}^{N-1} F_n e^{-2\pi i n x/N}, \tag{26.6}$$

which is our estimate in equation (26.2). The problem is that how good an approximation $rs(x;N)$ is of $f(x)$ will depend on $x$; as $|x|$ gets large the integrand becomes ever more oscillatory and a larger value of $N$ will be needed to obtain a good approximation of the integral.

The basic idea here is to use the measured values $F(2\pi n/N)$, $n = 0, ..., N-1$ to find an approximation of the function $F(\omega)$ and then to take the inverse Fourier transform of this approximation as our estimate of the function $f(x)$. One particular approximation we study in detail is a step function, although other approximations can be considered. It is helpful to remember that the estimate in equation (26.2) is $N$-periodic and is based on the unrealistic approximation of $F(\omega)$ by finitely many delta functions supported on the points $2\pi n/N$, $n = 0, ..., N-1$.

Consider the step function approximation of $F(\omega)$ given by

$$S(\omega) = \sum_{n=0}^{N-1} F_n \chi_{\pi/N}(\omega - \frac{2n+1}{N}\pi) \tag{26.7}$$

with

$$s(x) = \frac{1}{2\pi}\int_0^{2\pi} S(\omega) e^{-2\pi i x\omega} d\omega. \tag{26.8}$$

Performing the integrations we find that

$$s(x) = e^{-ix\pi/N} rs(x;N) \frac{\sin(\pi x/N)}{\pi x/N}. \tag{26.9}$$

If $N$ is large enough for $S(\omega)$ to provide a reasonable approximation of $F(\omega)$ then $s(x)$ should be a good estimate of $f(x)$, at least for smaller values of $x$. Of course, since the rate of decay of $f(x)$ as $|x|$ approaches infinity depends on the smoothness of $F(\omega)$ we must not expect $s(x)$ to approximate $f(x)$ well for larger values of $x$. Before leaving our discussion of problems of Type One we want to investigate to what extent the function $rs(x;N)$ provides a good estimate of $f(x)$.

Notice that the first positive zero of $\sin(\pi x/N)$ occurs at $x = N$, which suggests that $rs(x;N)$ provides a reasonable estimate of $f(x)$ for $|x|$ not larger than, say, $N/2$; therefore we may use $f_k$ to estimate $f(k)$ for $0 \leq k \leq N/2$. To be safe, we may wish to use a smaller upper bound on $k$.

Note also that $rs(-x;N) = rs(-x+N;N)$, which means that we may use $f_{N-k}$ to approximate $f(-k)$ for $0 < k \leq N/2$.

To summarize, the $N$ samples of $F(\omega)$ provide useful estimates $rs(k;N)$ of $f(k)$ for $-N/2 < k \leq N/2$. For $N = 2K$ we have $-K < k \leq K$, so that the $N$ samples of $F(\omega)$ provide $2K = N$ useful estimates of $f(k)$ through the use of $rs(k;N)$.

There is yet another way to look at this problem. If $F(\omega)$ is twice continuously differentiable then

$$F(\omega) = \sum_{m=-\infty}^{\infty} f(m)e^{im\omega} \tag{26.10}$$

with uniform convergence of this Fourier series for $0 \leq \omega \leq 2\pi$. Therefore, for $M$ large enough, we can estimate $F(\omega)$ using the truncated Fourier series

$$T(\omega;M) = \sum_{m=-M}^{M} f(m)e^{im\omega}. \tag{26.11}$$

Let $N = 2M + 1$ now.

Substituting $\omega = 2\pi n/N$ into equation (26.11) we obtain

$$T(2\pi n/N;M) = \sum_{m=-M}^{M} f(m)e^{2\pi imn/N}. \tag{26.12}$$

For $j = -M, ..., M$ multiply both sides of equation (26.12) by $e^{-2\pi inj/N}$, sum over $n = 0, ..., N-1$ and use orthogonality to get $f(j)$ on the right side and

$$\frac{1}{N}\sum_{n=0}^{N-1} T(2\pi n/N;M)e^{-2\pi inj/N} \tag{26.13}$$

on the left. Viewing $T(2\pi n/N;M)$ as an estimate of $F(2\pi n/N)$ and replacing the former by the latter in equation (26.13), we conclude once again that $f(k)$ is well approximated by $f_k$ for $0 \leq k \leq M$ and $f(-k)$ by $f_{N-k}$ for $1 \leq k \leq M$.

**Exercise 2:** Show that if $N = 2M$ is even and the function $F(\omega)$ is real-valued then $f_{N-j} = \overline{f_j}$ for $j = 1, ..., M-1$, where $f_k$ is given by equation (**??**).

When $F(\omega)$ is real-valued $f(x)$ is conjugate-symmetric, that is, $f(-x) = \overline{f(x)}$ for each $x$. It follows from Exercise 2 that if we view $f_j$ as an estimate of $f(j)$ for $j = 1, ..., M$, then we should view $f_{N-j}$ as an estimate of $f(-j)$.

It does not make good sense to view $f_{N-j}$ as an estimate of $f(N-j)$ since there need be no relation between $f(j)$ and $f(N-j)$, while $f_j$ and $f_{N-j}$ are complex conjugates of each other.

**Problems of Type Two:** In problems of Type Two we want to estimate the function $F(\omega)$ having bounded support and have samples of its Fourier transform, $f(x)$. As we shall see, this type of problem presents difficulties that are quite different from those presented by problems of Type One.

According to Shannon's sampling theorem we can recover $F(\omega)$ completely from the infinite sequence of samples $\{f(k\Delta)\}$, where $k$ runs over all the integers, for any sampling rate $\Delta \leq 1$. Unfortunately, we do not have infinitely many samples. In most applications there is a bounded set of $x$ variables within which we select our sampling points. We may take as many sampling points as we desire, but must remain within the bounded set. We need not take the samples equispaced one unit apart; in fact, we may take irregularly spaced sample points. Let us assume now that we have the samples $\{f(x_m),\, m = 1, ..., M\}$, from which to estimate the function $F(\omega)$. We have several options now. One method, which we shall discuss at length in subsequent chapters is the PDFT (see [46, 47, 43, 44]), which estimates $F(\omega)$ for all $\omega$, using a certain finite parameter model. Only after this is done is the estimated function discretized. A second method, the one we shall present here, is closely related to the first method, but begins with a discretization of the function $F(\omega)$. It is the discrete PDFT (DPDFT) method .

We select $N > M$ and replace the function $F(\omega)$ with the vector $\mathbf{F} = (F_1, F_2, ..., F_N)^T$, where the entry $F_n$ can be viewed as $F_n = F(2\pi(n-1)/N)$. Our data is

$$f(x_m) = \frac{1}{2\pi}\int_0^{2\pi} F(\omega)e^{-ix_m\omega}d\omega,$$

for $m = 1, ..., M$. We approximate the integrals with finite sums, obtaining

$$f(x_m) = \frac{1}{2\pi}\sum_{n=1}^{N} F_n e^{-2\pi i x_m n}, \tag{26.14}$$

which we write in matrix form as $\mathbf{f} = A\mathbf{F}$, with $A$ the $M$ by $N$ matrix with entries $A_{mn} = \frac{1}{2\pi}\exp(-ix_m n)$. Since $M < N$ the systems of equations $A\mathbf{F} = \mathbf{f}$ will typically have infinitely many solutions. Our goal is to incorporate our prior knowledge of the function $F(\omega)$ in the choice of solution.

A common choice in such underdetermined problems is to select the *minimum norm* solution, given by

$$\mathbf{F}_{minnorm} = A^\dagger(AA^\dagger)^{-1}\mathbf{f},$$

where the superscript † indicates conjugate transpose and we assume, reasonably, that the matrix $AA^\dagger$ is invertible. However, suppose we have some prior information about the shape of the function $F(\omega)$, such as it is zero outside some interval $[a, b]$ contained within $[0, 2\pi]$, or, more generally, $|F(\omega)|$ can be approximated by some nonnegative function $P(\omega) \geq 0$. We then let $P_n = P(2\pi(n-1)/N)$ and $W_n = P_n^{-1/2}$ whenever $P_n > 0$; let $W_n = \alpha > 0$ for some small $\alpha > 0$ otherwise. Let $W$ be the diagonal matrix with entries $W_n$. The minimum weighted norm solution of $\mathbf{f} = A\mathbf{F}$ is

$$\mathbf{F}_{mwn} = W^{-1}A^\dagger(AW^{-1}A^\dagger)^{-1}\mathbf{f}.$$

This minimum weighted norm solution can be obtained from the minimum norm solution of a related system of linear equations. Let $B = AW^{-1/2}$ and $\mathbf{G} = W^{1/2}\mathbf{F}$. Then $\mathbf{f} = A\mathbf{F} = B\mathbf{G}$. The minimum norm solution of $\mathbf{f} = B\mathbf{G}$ is

$$\mathbf{G}_{minnorm} = B^\dagger(BB^\dagger)^{-1}\mathbf{f} = W^{-1/2}A^\dagger(AW^{-1}A^\dagger)^{-1}\mathbf{f}$$

and

$$\mathbf{F}_{mwn} = W^{-1/2}\mathbf{G}_{minnorm}.$$

We calculate $\mathbf{F}_{mwn}$ iteratively, either by applying the *algebraic reconstruction technique* (ART) directly to the system $\mathbf{f} = B\mathbf{G}$ or rewriting the ART iterative step for this system in terms of the original system $\mathbf{f} = A\mathbf{F}$.

When the data is noisy we often do not want an exact solution of $\mathbf{f} = A\mathbf{F}$. In that case we *regularize* by taking as our approximate solution the vector

$$\mathbf{F}_{rmwn} = W^{-1}A^\dagger(AW^{-1}A^\dagger + \epsilon^2 I)^{-1}\mathbf{f},$$

where $\epsilon > 0$ is small and $I$ is the identity matrix. This solution can also be found iteratively, using ART, without having to calculate the matrix $AW^{-1}A^\dagger$.

# Chapter 27

# A Brief Look at the ART

In applied mathematics it is often the case that the solution to our problem cannot be written in closed form, nor can it be calculated exactly in a finite number of steps. In such cases we are forced to find approximate solutions using iterative algorithms; the Newton-Raphson method for solving $f(x) = 0$ is an example of an iterative method. There are also situations in which, in theory, the solution can be found exactly, assuming infinitely precise calculations, but to do so would be impractical: solving large systems of linear equations is an example of such a problem. We know that, in theory, Gauss elimination will find the solution in a finite number of steps, if there is a unique solution. But, when there are thousands of equations in thousands of unknowns, as is commonly the case in image processing, Gauss elimination is not practical. The iterative *algebraic reconstruction technique* (ART) was devised to solve just such large systems of linear equations.

Finding a solution to the system of linear equations given in matrix form by $A\mathbf{x} = \mathbf{f}$ is equivalent to finding a vector $\mathbf{x}$ in $R^J$ that is in all of the sets

$$H_m = \{x | (A\mathbf{x})_m = f_m\},$$

for $m = 1, ..., M$. The sets $H_m$ are *hyperplanes* in $R^J$. One way to find such an $\mathbf{x}$ is to use the ART method.

In ART we begin with an arbitrary starting vector $\mathbf{x}^0$. We then let $\mathbf{x}^1$ be the vector in $H_1$ closest to $\mathbf{x}^0$, then $\mathbf{x}^2$ the vector in $H_2$ closest to $\mathbf{x}^1$, and so on. When we have found vector $\mathbf{x}^M$ in $H_M$ closest to $\mathbf{x}^{M-1}$, we then let $\mathbf{x}^{M+1}$ be the vector in $H_1$ closest to $\mathbf{x}^M$, etc.; that is, we cycle once again through each of the $M$ hyperplanes. This process is known to converge to the vector closest to $\mathbf{x}^0$ that is in all of the $H_m$.

Given any vector $\mathbf{x}$ and hyperplane $H_m$, the vector $\mathbf{z}$ in $H_m$ closest to

$\mathbf{x}$ can be written explicitly. We have

$$z_j = x_j + A_{mj}(f_m - (A\mathbf{x})_m)/\sum_{n=1}^{J} A_{mn}^2.$$

Therefore, the ART algorithm can be written explicitly as follows: for $k = 0, 1, ...$ and $m = k(\mathrm{mod}\, M) + 1$ we have

$$x_j^{k+1} = x_j^k + A_{mj}(f_m - (A\mathbf{x}^k)_m)/\sum_{n=1}^{J} A_{mn}^2.$$

It is known that the ART can be slow to converge if the equations that make up $A\mathbf{x} = \mathbf{f}$ are ordered so that successive rows of $A$ are not significantly different. To avoid this it is highly recommended that the equations be reordered according to some random selection prior to using ART.

In a later chapter we shall examine the ART and related algorithms, such as the *multiplicative* ART (MART), in the context of block-iterative methods.

# Chapter 28

# Bandlimited Extrapolation

Let $f(x)$ and $F(\omega)$ be a Fourier transform pair. We know from the formulas in equations (13.1) and (13.2) that we can determine $F$ from $f$ and vice versa. But what happens if we have some, but not all, of the values $f(x)$? Can we still find $F(\omega)$ for all $\omega$? If we can, then we can also recover the missing values of $f$, which says that there must be considerable redundancy in the way $f$ stores information. We shall investigate this matter further now for the important case in which $F$ has bounded support; that is, there is some $\Omega > 0$ such that $F(\omega) = 0$, for $|\omega| > \Omega$. The function $f(x)$ is then said to be $\Omega$-*bandlimited*.

We shall assume throughout this chapter that $f$ is $\Omega$-bandlimited and ask how much we need to know about $f$ to recover $F(\omega)$ for all $\omega$. Because recovering $F(\omega)$ for all $\omega$ is equivalent to finding $f(x)$ for all $x$, this problem is called the *bandlimited extrapolation problem*.

We have already encountered one result along these lines. According to Shannon's sampling theorem, if we have the values $\{f(n\Delta),\ -\infty < n < \infty\}$, for some $\Delta \in (0, \frac{\pi}{\Omega}]$, then we can recover $F(\omega)$ for all $\omega$ and thereby $f(x)$ for all $x$. Therefore, these infinite sequences of samples of $f$ contain complete information about $f$. Other results of this sort have quite a different flavor.

Since $F(\omega) = 0$ outside its interval of support $[-\Omega, \Omega]$ the extension of $f(x)$ to complex $z$, given by the *Fourier-Laplace transform*

$$f(z) = \int_{-\infty}^{\infty} F(\omega) e^{-iz\omega} d\omega/2\pi, \tag{28.1}$$

can be differentiated under the integral sign since the limits of integration are now finite. In fact, the function $f(z)$ is a complex-valued function that

is analytic throughout the complex plane. Such functions have power series expansions that converge for all $z$.

**Exercise 1:** Show that there can be no Fourier transform pair $f, F$ for which positive constants $a$ and $b$ exist such that $f(x) = 0$ for $|x| > a$ and $F(\omega) = 0$ for $|\omega| > b$. Thus it is not possible for both $f$ and $F$ to be band-limited.

**Hint:** Use the analyticity of the function $f(z)$.

The coefficients needed for such a power series expansion are determined by the derivatives of $f(z)$ at a single point, say $z = 0$. Therefore, if we have the values of $f(z)$ for $z$ in some small disc around $z = 0$ we have all the information we need. Actually, even this amount of knowledge about $f$ is too much; to calculate the derivatives at $z = 0$ we need only know $f(x_n)$ for some sequence $\{x_n\}$ of real numbers converging to $z = 0$.

This is fine in theory, but, of course, we cannot hope to calculate all the derivatives of $f$ at $z = 0$. Even calculating a few derivatives in the presence of noisy measurements of $f$ is hopeless. In [152] Papoulis presents an iterative scheme for determining $F(\omega)$ from knowledge of $f(x)$ for $x$ within an interval $A = [a, b]$ of the real line. This is not a practical technique, since it uses infinitely many samples of $f(x)$, but can be modified to provide useful algorithms, as we shall see. The iterative and non-iterative methods we describe below are usually called *super-resolution techniques* in the signal processing literature. Similar methods applied in sonar and radar array processing are called *super-directive* methods [75].

**Papoulis' iterative method:** Let $g^0(x) = \chi_A(x)f(x)$. Having found $g^k(x)$ let $G^k(\omega)$ be the FT of $g^k$, $H^k(\omega) = \chi_\Omega(\omega)G^k(\omega)$ and $h^k(x)$ the inverse FT of $H^k(\omega)$. Then take $g^{k+1}(x) = f(x)$ for $x \in A$ and $g^{k+1}(x) = h^k(x)$ otherwise. The sequence $\{h^k(x)\}$ converges to $f(x)$ for all $x$ and the sequence $\{H^k\}$ converges in the mean square sense to $F$.

In practice we have only finitely many values of $f(x)$. This is not, of course, enough information to determine $F(\omega)$. We seek an estimate of $F$, or, equivalently, an approximate extrapolation of the data. We consider now several practical variants of Papoulis' iterative method.

**Gerchberg-Papoulis iteration (I):** The algorithm discussed in this section is called the *Gerchberg-Papoulis* (GP) bandlimited iteration method [100], [151]. For notational convenience we shall assume that $\Omega < \pi$ and that we have the finite data $f(n)$, $n = 0, 1, ..., M-1$. We seek to estimate the values $f(n)$, $n = M, M+1, ..., N$ for some choice of $N > M$. We begin with $g^0$ the $N$-dimensional vector with entries $g^0(n) = f(n)$ for

$n = 0, 1, ..., M-1$ and $g^0(n) = 0$ for $n = M, M+1, ..., N-1$. Then having found the vector $g^k$ we let

$$G_m^k = \sum_{n=0}^{N-1} g^k(n) \exp(2\pi imn/N),$$

for $m = 0, 1, ..., N-1$. We interpret these values as samples of a function $G^k(\omega)$ defined on $[-\pi, \pi]$; specifically, we take

$$G_m^k = G^k(2\pi m/N)$$

for $m = 0, 1, ..., \frac{N}{2}$ and

$$G_m^k = G^k(-2\pi + 2\pi m/N)$$

for $m = \frac{N}{2} + 1, ..., N-1$; for convenience we assume that $N$ is even. Mimicking the definition of $H^k(\omega)$, we define $H_m^k$ to be $G_m^k$ for those $m = 0, 1, ..., \frac{N}{2}$ such that $2\pi m/N \leq \Omega$ and for those $m = \frac{N}{2} + 1, ..., N-1$ for which $-2\pi + 2\pi m/N \geq -\Omega$. For all other values of $m$ we set $H_m^k = 0$. Now calculate

$$h_n^k = \frac{1}{N} \sum_{m=0}^{N-1} H_m^k \exp(-2\pi imn/N),$$

for $n = 0, 1, ..., N-1$. Finally, set $g_n^{k+1} = f(n)$, for $n = 0, 1, ..., M-1$ and $g_n^{k+1} = h_n^k$ for $n = M, M+1, ..., N-1$. The limit vector $g^\infty$ has $g_n^\infty = f(n)$ for $n = 0, 1, ..., M-1$, but in order to have $G_m^\infty = 0$ for those $m$ corresponding to frequencies outside $[-\Omega, \Omega]$ we need to take $N \geq M\pi/\Omega$. The values $g_n^\infty$ for $n = M, M+1, ..., N-1$ are then our extrapolated values of $f$.

The advantages of this approach are that only finite data is used and the calculations can be performed using the fast Fourier transform. The vectors obtained are optimal in some sense [53], [54]. Obviously, one drawback is that we do not extrapolate $f(n)$ for all integers $n$, but only for a finite subset. Also, we do not obtain a function $G^\infty(\omega)$ of the continuous variable $\omega$ that is equal to zero for all $\omega$ outside the band $[-\Omega, \Omega]$ and whose corresponding $g^\infty(x)$ is consistent with the finite data. To remedy this we consider another variant of the GP algorithm.

**Gerchberg-Papoulis iteration (II):** We shall assume again that $\Omega < \pi$ and that we have the finite data $f(n)$, $n = 0, 1, ..., M-1$. Since

$$F(\omega) = \sum_{n=-\infty}^{\infty} f(n) \exp(in\omega)$$

for $\omega \in [-\pi, \pi]$, we seek to extrapolate $f(n)$ for $n$ not in the set $\{0, 1, ..., M-1\}$.

Mimicking the algorithm in the previous section, we begin with the infinite sequence $g^0 = \{g_n^0, -\infty < n < \infty\}$ where $g_n^0 = f(n)$ for $n = 0, 1, ..., M-1$ and $g_n^0 = 0$ otherwise. Having found the infinite sequence $g^k$ we define

$$G^k(\omega) = \sum_{n=-\infty}^{\infty} g_n^k \exp(in\omega)$$

for $\omega \in [-\pi, \pi]$. Then we set

$$H^k(\omega) = \chi_\Omega(\omega) G^k(\omega)$$

and

$$h_n^k = \frac{1}{2\pi} \int_{-\pi}^{\pi} H^k(\omega) \exp(-in\omega) d\omega.$$

Then let $g_n^{k+1} = f(n)$ for $n = 0, 1, ..., M-1$ and $g_n^{k+1} = h_n^k$ otherwise.

It would appear that this iterative scheme cannot actually be performed because it requires calculating $g_n^{k+1}$ for all integers $n$. Fortunately, there is a way out.

**Non-iterative bandlimited extrapolation:** Note that $G^{k+1}(\omega)$ can be written as

$$G^{k+1}(\omega) = H^k(\omega) + G^0(\omega) - \sum_{n=0}^{N-1} h_n^k \exp(in\omega),$$

so that

$$H^{k+1}(\omega) - H^k(\omega) = \chi_\Omega(\omega) \sum_{n=0}^{N-1} a_n^k \exp(in\omega) \tag{28.2}$$

for some $a_0^k, ..., a_{N-1}^k$. If we wish we can implement the GP iterative method by iteratively updating these constants. There is a better way to proceed, however.

It follows from equation (28.2) and the definition of $H^0$ that the limit $H^\infty(\omega)$ has the form

$$H^\infty(\omega) = \chi_\Omega(\omega) \sum_{n=0}^{N-1} a_n \exp(in\omega) \tag{28.3}$$

for some constants $a_0, ..., a_{N-1}$. We then solve for these coefficients using our data. Taking the inverse Fourier transform of both sides of equation (28.3)and forcing data consistency, we obtain the system of equations

$$f(m) = \sum_{n=0}^{N-1} a_n \frac{\sin \Omega(m-n)}{\pi(m-n)}, \tag{28.4}$$

$m = 0, ..., N-1$, which we solve to find the coefficients. Once we have the coefficients we insert them into the expression for $H^\infty(\omega)$ to obtain a function supported on the interval $[-\Omega, \Omega]$ whose associated $h^\infty(x)$ is consistent with the data. The extrapolated sequence is then $\{h^\infty(n)\}$ for integers $n$ not between 0 and $M-1$. This noniterative implementation of the GP extrapolation is not new; it was presented in [45], and has been rediscovered several times since then (see p. 209 of [170]).

Because our data usually contains noise we need to exercise some care in solving the system in equation (28.4). The matrix $S$ whose entries are

$$S_{mn} = \frac{\sin \Omega(m-n)}{\pi(m-n)}$$

is typically ill-conditioned, particularly when $\Omega$ is much smaller than $\pi$. To reduce sensitivity to noise we can *regularize*; one way is to multiply the entries on the main diagonal of $S$ by, say, 1.0001. This increases the eigenvalues of $S$, thereby decreasing the eigenvalues of $S^{-1}$ and making the computed solution less sensitive to the noise.

The finite data we have tells us nothing about the values $f(n)$ we have not measured, in the sense that we can define $f(M)$ any way we wish and still construct an $\Omega$-bandlimited function consistent with the data and with this chosen value of $f(M)$. In a similar sense our finite data also tells us nothing about the value of $\Omega$; we can select any interval $[a, b]$ and find a function $H(\omega)$ supported on $[a, b]$ whose $h(x)$ is consistent with the data. But this is not quite the whole story; finite data cannot rule out anything, but it can suggest strongly that certain things are false. For example, if we select the interval $[a, b]$ disjoint from $[-\Omega, \Omega]$ the function $H(\omega)$ will probably have large energy; that is, the integral

$$\int_a^b |H(\omega)|^2 d\omega$$

will be much larger than

$$\int_{-\Omega}^{\Omega} |H^\infty(\omega)|^2 d\omega.$$

We can use this fact to help us decide if we have chosen a good value for $\Omega$. In [43] this same idea was used to obtain an iterative algorithm for solving the phase retrieval problem discussed in a later chapter.

When the data set is large, as usually happens in multi-dimensional problems such as image reconstruction, solving the equations (28.4) is sometimes performed iteratively. Nevertheless, the algorithm still differs from the first GP method in that we are still extrapolating infinitely many values of $f(n)$; we are just doing it using a finite parameter model.

The non-iterative implementation of the Gerchberg-Papoulis bandlimited extrapolation method can be extended in several ways to solve Fourier transform estimation problems. The *modified* DFT (MDFT) estimator generalizes the non-iterative GP method to accomodate non-equispaced sampling. More generally, the PDFT method permits us to include other prior information about the shape of $F(\omega)$ beyond knowledge of its support; it also applies to multi-dimensional problems. Constructing the matrix used in the system of equations can be difficult when the data sets are large; an iterative discrete implementation of the PDFT, the DPDFT, allows us to avoid dealing with this large matrix. There is also a nonlinear version of the PDFT, the *indirect* PDFT (IPDFT), that extends the maximum entropy method for extrapolating autocorrelation data.

# Chapter 29

# Fourier Transform Estimation

The basic problem we want to solve is the reconstruction of an object function $F(\omega)$ from finitely many values of its inverse Fourier transform

$$f(x) = \int F(\omega) \exp(-ix\omega) d\omega/2\pi, \tag{29.1}$$

where, for notational convenience, we use single letters $x$ and $\omega$ to denote possibly multi-dimensional variables. We assume that the formula

$$F(\omega) = \int f(x) \exp(ix\omega) dx$$

also holds.

Let the data be $f(x_m)$, $m = 1, ..., M$. Given this data, we want to estimate $F(\omega)$. Notice that any estimate of $F(\omega)$, which we denote as $\hat{F}(\omega)$, corresponds to an estimate of $f(x)$ by inserting $\hat{F}(\omega)$ into equation (29.1); that is

$$\hat{f}(x) = \int \hat{F}(\omega) \exp(-ix\omega) d\omega/2\pi. \tag{29.2}$$

We shall say that the estimate $\hat{F}(\omega)$ is *data consistent* if

$$\hat{f}(x_m) = f(x_m),\ m = 1, ..., M.$$

**A first estimate for $F(\omega)$:** It seems reasonable to take as our first attempt the estimate

$$\hat{F}(\omega) = \sum_{m=1}^{M} f(x_m) \exp(ix_m\omega). \tag{29.3}$$

Is this estimate data consistent? Let's calculate $\hat{f}(x)$ and see. Inserting $\hat{F}(\omega)$ in equation (29.3) into equation (29.2) we get

$$\hat{f}(x) = \sum_{m=1}^{M} f(x_m)\delta(x - x_m),$$

where $\delta(x - a)$ denotes the Dirac delta function supported at the point $a$. The estimate is not data consistent, since what we measured at $x = x_m$ was not the top of a delta function, but just a number, $f(x_m)$. Does our estimate seem reasonable now? Is it reasonable that the estimate of the function $f(x)$ just happens to have delta function components located at precisely the places we chose to sample and is zero everywhere else? Perhaps we can do better.

We go beyond our first estimation attempt by incorporating some prior knowledge in our estimate, or, at least, making reasonable assumptions about the function $F(\omega)$ being estimated. The first type of assumption we make concerns the support of $F(\omega)$, that is, the region in $\omega$-space outside of which $F(\omega)$ is identically equal to zero.

**Including a support constraint:** Let $\Omega > 0$ and suppose that the function $F(\omega) = 0$ for $|\omega| > \Omega$. Let $\chi_\Omega(\omega)$ be the function that is one for $|\omega| \leq \Omega$ and zero otherwise. Building on our first attempt, we try the estimate

$$\hat{F}(\omega) = \chi_\Omega(\omega) \sum_{m=1}^{M} f(x_m) \exp(ix_m\omega). \tag{29.4}$$

Is this estimate data consistent? Inserting $\hat{F}(\omega)$ in equation (29.4) into equation (29.2) we get

$$\hat{f}(x) = \sum_{m=1}^{M} f(x_m) \frac{\sin \Omega(x - x_m)}{\pi(x - x_m)}. \tag{29.5}$$

Now we ask if it is true that

$$f(x_n) = \sum_{m=1}^{M} f(x_m) \frac{\sin \Omega(x_n - x_m)}{\pi(x_n - x_m)} \tag{29.6}$$

for $n = 1, ..., M$. The answer is, generally, no, although in special cases, the answer is yes, or almost yes.

**The Nyquist case:** Suppose that $\Omega = \pi$, $F(\omega)$ is zero for $|\omega| > \pi$ and the data is $f(m)$, $m = 1, ..., M$. Then the estimate

$$\hat{F}(\omega) = \chi_\pi(\omega) \sum_{m=1}^{M} f(m) \exp(im\omega)$$

is data consistent; it is then what is often called the *discrete Fourier transform* (DFT) of the data, defined for $\omega$ in the interval $[-\pi, \pi]$. For this reason we write the estimate as $F_{DFT}(\omega)$. The inversion formula gives

$$\hat{f}(x) = \sum_{m=1}^{M} f(m) \frac{\sin \pi(x-m)}{\pi(x-m)}$$

and

$$\hat{f}(n) = \sum_{m=1}^{M} f(m) \frac{\sin \pi(n-m)}{\pi(n-m)}$$

holds for each $n = 1, ..., M$, since the matrix becomes the identity matrix.

Suppose, more generally, that $\Omega = \frac{\pi}{\Delta}$ for some $\Delta > 0$, $F(\omega)$ is zero for $|\omega| > \frac{\pi}{\Delta}$ and the data is $f(m\Delta)$, $m = 1, ..., M$. Then the estimate

$$\hat{F}(\omega) = \chi_{\frac{\pi}{\Delta}}(\omega) \sum_{m=1}^{M} f(m\Delta) \exp(im\Delta\omega)$$

is almost data consistent. The inversion formula gives

$$\hat{f}(x) = \sum_{m=1}^{M} f(m\Delta) \frac{\sin \frac{\pi}{\Delta}(x-m\Delta)}{\pi(x-m\Delta)}$$

and so

$$\hat{f}(n\Delta) = \frac{1}{\Delta} \sum_{m=1}^{M} f(m\Delta) \frac{\sin \pi(n-m)}{\pi(n-m)} = \frac{1}{\Delta} f(n\Delta)$$

holds for each $n = 1, ..., M$. To get data consistency we multiply our estimate by $\Delta$; that is, we take

$$\hat{F}(\omega) = \Delta \chi_{\frac{\pi}{\Delta}}(\omega) \sum_{m=1}^{M} f(m\Delta) \exp(im\Delta\omega).$$

Now this estimate is both data consistent and supported on the interval $[-\frac{\pi}{\Delta}, \frac{\pi}{\Delta}]$. This estimate may also be called the DFT, ignoring the $\Delta$ multiplier or redefining variables to make $\Delta = 1$.

**Exercise 1:** Use the orthogonality principle to show that the DFT minimizes the distance

$$\int_{-\pi}^{\pi} |F(\omega) - \sum\nolimits_{m=1}^{M} a_m e^{im\omega}|^2 d\omega.$$

When the data is $f(m\Delta)$, so is equispaced, we assume that $F(\omega) = 0$ for $|\omega| > \frac{\pi}{\Delta}$; that is, we assume that our sample spacing $\Delta$ is small enough to

avoid aliasing. What happens when we *oversample*; that is, when $F(\omega) = 0$ for $|\omega| > \Omega$, where $\Omega < \frac{\pi}{\Delta}$?

**The general case:** Even for integer spaced data $f(m)$, $m = 1, ..., M$, the estimate

$$\hat{F}(\omega) = \chi_\Omega(\omega) \sum_{m=1}^{M} f(m) \exp(im\omega)$$

will not be data consistent if $\Omega < \pi$. For more generally spaced data $f(x_m)$, $m = 1, ..., M$ the estimate

$$\hat{F}(\omega) = \chi_\Omega(\omega) \sum_{m=1}^{M} f(x_m) \exp(ix_m\omega)$$

will not be data consistent. The approach we take is to retain the algebraic form of these estimators, but to allow the coefficients to be determined by data consistency.

Take as the estimate of $F(\omega)$ the function

$$F_\Omega(\omega) = \chi_\Omega(\omega) \sum_{m=1}^{M} a_m \exp(ix_m\omega), \tag{29.7}$$

with the coefficients $a_m$ chosen to give data consistency. This means we must select the $a_m$ to satisfy the equations

$$f(x_n) = \sum_{m=1}^{M} a_m \frac{\sin \Omega(x_n - x_m)}{\pi(x_n - x_m)}$$

for $n = 1, ..., M$. The resulting estimate $F_\Omega(\omega)$ is both data consistent and supported on the interval $[-\Omega, \Omega]$. This *non-iterative bandlimited extrapolation method* was called the *modified DFT* (MDFT) in [45]. Figure 29.1 below shows the advantage of the MDFT, in the top frame, over the DFT below. The true object to be reconstructed is the solid figure. The sampling spacing is $\Delta = 1$, but $\Omega = \pi/30$, so the 129 data points are thirty times oversampled.

**A paradox:** It follows from what we just did that for any finite data and any $\alpha < \beta$ there is a function $\hat{F}(\omega)$ supported on the interval $[\alpha, \beta]$ and consistent with the data. Does the data contain no information about the actual support of $F(\omega)$? This would seem to say that the data we have measured contains essentially no information, since we can generate thousands of additional data points, select any $\alpha$ and $\beta$ and still find a data consistent estimate of $F(\omega)$. How can this be true when, at the same time,

we have plenty of simulation cases in which we are able to generate fairly accurate estimates of the correct answer using these techniques?

The answer is that while the data we have does not eliminate any possible support for the function $F(\omega)$ it is capable of indicating preferences. When we use equation (29.7) we do get an estimate that is data consistent, but if the support $[-\Omega, \Omega]$ is a poor choice we usually have an indication of that in the norm of the estimate. The norm of $F_\Omega(\omega)$ is

$$||F_\Omega|| = \sqrt{\int_{-\Omega}^{\Omega} |F_\Omega(\omega)|^2 d\omega}$$

and can be quite large if the data and the $\Omega$ are poorly matched. Usually, the true $F(\omega)$ is a physically meaningful function that does not have unusually large norm, so any estimate $F_\Omega(x)$ with a large norm is probably incorrect and a better $\Omega$ should be sought.

**Properties of the estimate $F_\Omega(\omega)$:** In addition to being data consistent and having for its support the interval $[-\Omega, \Omega]$ the estimate $F_\Omega(\omega)$ given by equation (29.7) has two additional properties that are worth mentioning. The choice $G(\omega) = F_\Omega(\omega)$ minimizes the integral

$$\int_{-\Omega}^{\Omega} |G(\omega)|^2 dx$$

over all estimates $G(\omega)$ that are data consistent. It also minimizes the approximation error

$$\int_{-\Omega}^{\Omega} |F(\omega) - \sum_{m=1}^{M} a_m \exp(ix_m\omega)|^2 d\omega \tag{29.8}$$

over all choices of coefficients $a_m$. So in this sense it is the best approximation of the truth that we can find that has its particular algebraic form, provided, of course, that $F(\omega)$ is supported on $[-\Omega, \Omega]$.

**Exercise 2:** Suppose that $0 < \Omega$ and $F(\omega) = 0$ for $|\omega| > \Omega$. Let $f(x)$ be the inverse Fourier transform of $F(\omega)$ and suppose that the data is $f(x_m)$, $m = 1, ..., M$. Use the orthogonality principle to find the coefficients $a_m$ that minimize the error given by equation (29.8). Show that the resulting estimate of $F(\omega)$ is consistent with the data.

The choice of $\Omega$ is left up to us. Suppose that our choice is too big. Then the estimate in equation (29.7) gives the best estimate of its algebraic form over the interval $[-\Omega, \Omega]$, but since $F(\omega)$ is zero on a portion of this interval, the estimate spends some effort estimating the value zero. If we

can get a more accurate estimate of the true support of $F(\omega)$ then we can modify the $\Omega$ and get a better estimate of $F(\omega)$.

Once we have calculated the estimate $F_\Omega(\omega)$ we obtain a procedure for extrapolating the data by computing its inverse Fourier transform:

$$f_\Omega(x) = \sum_{m=1}^{M} a_m \frac{\sin \Omega(x - x_m)}{\pi(x - x_m)}$$

estimates the values $f(x)$ we did not measure. This procedure extends the Gerchberg-Papoulis (GP) method for bandlimited extrapolation that we saw in the previous chapter.

**The PDFT:** The estimate $F_\Omega(\omega)$ is the product of two terms: the first is $\chi_\Omega(\omega)$, which incorporates prior knowledge about the function $F(\omega)$, and the second is the sum, whose coefficients are calculated to insure data consistency. We obtain a more flexible class of estimators by replacing the first term, $\chi_\Omega(\omega)$, with $P(\omega) \geq 0$, a prior estimate of the magnitude of $F(\omega)$. The resulting estimate, called the PDFT, is the subject of the next chapter.

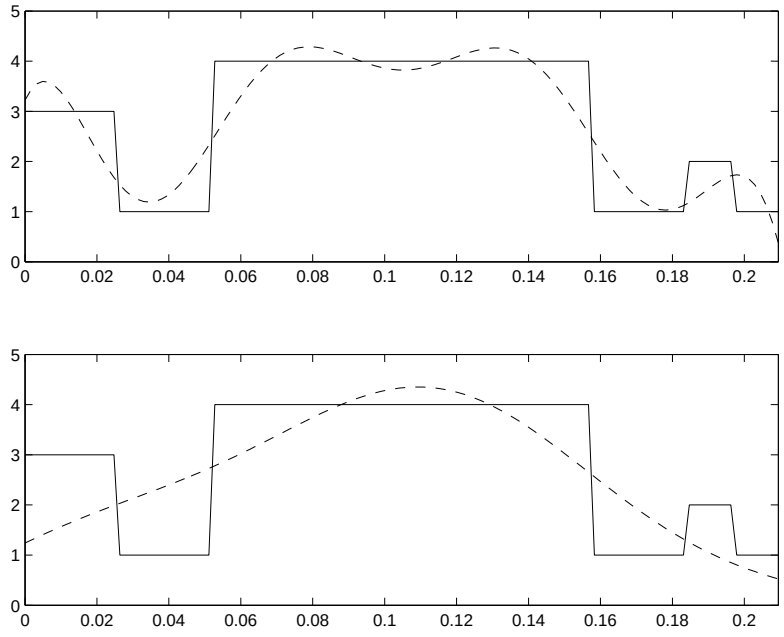



Figure 29.1: The non-iterative bandlimited extrapolation method (MDFT) (top) and the DFT (below) for $M = 129$, $\Delta = 1$ and $\Omega = \pi/30$.

# Chapter 30

# The PDFT

Most of the time the data we have is noisy, the data we have isn't really the data we want, the locations where we measured the data were the ones available, not the ones we wanted to use, the physical model we are using to interpret the data is not quite right, but is the best we can do, and we don't have enough data. All these difficulties are important and we shall deal with each one of them in one way or another. Beginning with the discussion of bandlimited extrapolation and continuing through this chapter, we focus on the last problem, the limited data problem.

In many estimation and reconstruction problems we have a limited amount of data that is not sufficient, by itself, to provide a useful result; additional information is needed. In the bandlimited extrapolation problem just discussed we were able to use the information about the support of the Fourier transform function $F(\omega)$ to improve our estimate. We may, at times, have some prior estimate not only of the support, but of its overall shape; such prior profile information can be useful in estimating $F(\omega)$. The PDFT [46], [47] is a generalization of the MDFT in equation (29.7), designed to permit the use of such prior profile estimates.

Suppose now that the data is $f(x_m)$, $m = 1, ..., M$. Suppose also that we have some prior estimate of the magnitude of $F(\omega)$ for each real $\omega$, in the form of a function $P(\omega) \geq 0$. In the previous chapter $P(\omega)$ appeared as $\chi_\pi(\omega)$ and $\chi_\Omega(\omega)$. We take as our estimate of $F$ the function of the form

$$F_{PDFT}(\omega) = P(\omega) \sum_{m=1}^{M} c_m \exp(ix_m\omega), \tag{30.1}$$

where the $c_m$ are chosen to give data consistency.

**Exercise 1:** Show that the $c_m$ must satisfy the equations

$$f(x_n) = \sum_{m=1}^{M} c_m p(x_n - x_m),\ n = 1, ..., M, \tag{30.2}$$

where $p(x)$ is the inverse Fourier transform of $P(\omega)$. Note that for $P(\omega) = \chi_\Omega(\omega)$ we have $p(x) = \frac{\sin(\Omega x)}{\pi x}$.

Both of the estimates $F_{DFT}(\omega)$ and $F_\Omega(\omega)$ provide a best approximation of its form and support for $F(\omega)$. The same is true of the PDFT.

**Exercise 2:** Show that the estimate $F_{PDFT}(\omega)$ minimizes the distance

$$\int |F(\omega) - P(\omega) \sum\nolimits_{m=1}^{M} a_m \exp(ix_m\omega)|^2 P(\omega)^{-1} d\omega$$

over all choices of the coefficients $a_m$.

Both of the estimates $F_{DFT}(\omega)$ and $F_\Omega(\omega)$ minimize an energy, subject to data consistency. Something similar happens with the PDFT; the PDFT minimizes the weighted energy

$$\int_{-\pi}^{\pi} |F_{PDFT}(\omega)|^2 P(\omega)^{-1} d\omega, \tag{30.3}$$

subject to data consistency, with the understanding that $P(\omega)^{-1} = 0$ if $P(\omega) = 0$. That the PDFT is a minimum weighted energy solution will be important later when we turn to the discrete PDFT.

For relatively small $M$ the PDFT is easily calculated. The difficult part is constructing the matrix $P$ having the entries $P_{m,n} = p(x_m - x_n)$, which requires the calculation of the inverse Fourier transform of $P(\omega)$ at the irregularly spaced points $x_m - x_n$. In addition, the matrix $P$ is often ill-conditioned, meaning that some of its (necessarily positive) eigenvalues are near zero. Noise in the data $f(x_m)$ can lead to unreasonably large values of $c_m$ and to a PDFT estimate that is useless. To combat this problem we can multiply the terms $P_{n,n}$ on the main diagonal of $P$ by (say) 1.001. This prevents the eigenvalues from becoming too small.

For large data sets it is more difficult to work with the PDFT as formulated. The matrix $P$ is very large, its entries difficult to compute, storage becomes a problem and solving the resulting system of equations is expensive. To avoid all these problems and to have a formulation of the PDFT that is conceptually easier to use we turn to a discrete formulation, which we call the DPDFT.

In a recent article [157] Poggio and Smale discuss the use of positive-definite kernels for interpolation, in the context of artificial intelligence and supervised learning.

Figure 30.1 below illustrates the DFT, MDFT and the PDFT; Figure 30.2 zooms in on the smaller peak. The original object is in the upper left. Its support is contained within the interval $[0, 128]$. The data are the

Fourier transform values $f(\frac{2\pi n}{4096}), |n| \leq 500$; therefore the data is thirty-two times oversampled. The MDFT uses as the object support the interval $[13, 117]$ and the PDFT uses the main lobe of the original as the prior; the matrix in both cases is regularized. By incorporating prior information about the object to be reconstructed in the first factor $P(\omega)$ the PDFT allows the trigonometric polynomial that is the second factor to describe only those parts of the object not already accounted for by the prior. Figure 30.3 shows only the polynomial factors in each estimate.

The usefulness of the PDFT in image processing is illustrated in Figure 30.4. The original is a simulated head slice. The data are low spatial frequency values. The DFT does show us that the object is round and appears to have a skull-type outer layer. Beyond that, it tells us nothing of use about the interior. From the DFT image or from prior knowledge of the problem at hand, we take as our prior estimate of the image the skull shape, with a uniform interior. Using this prior and the same low-pass data the PDFT can recover the original with only slight blurring.

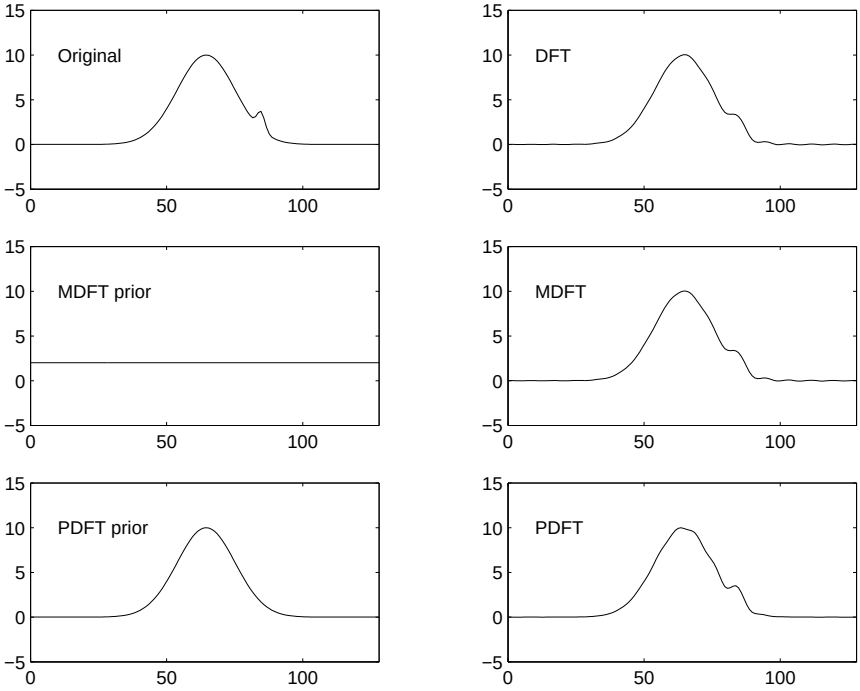



Figure 30.1: The DFT, MDFT and PDFT

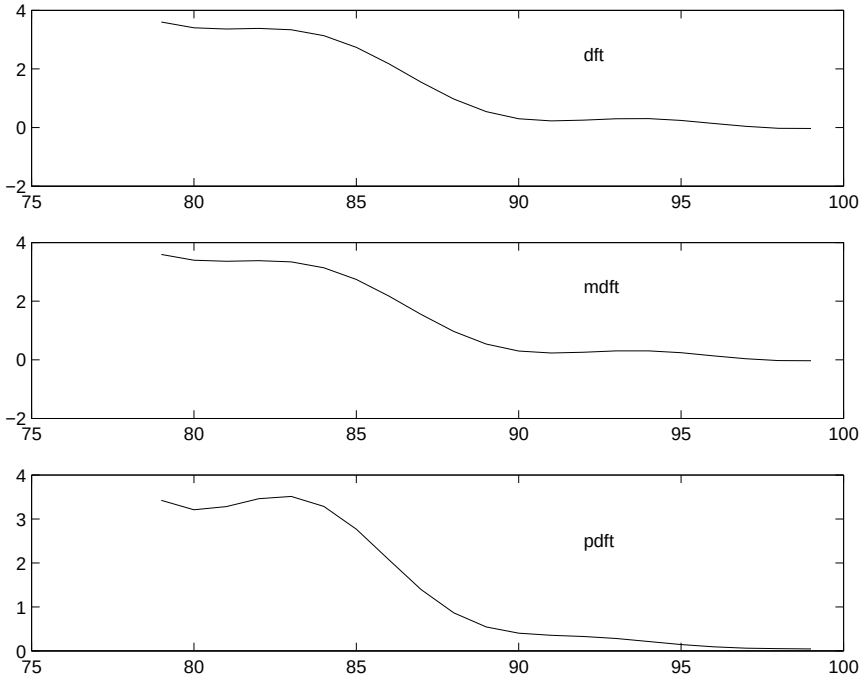



Figure 30.2: The DFT, MDFT and PDFT up close

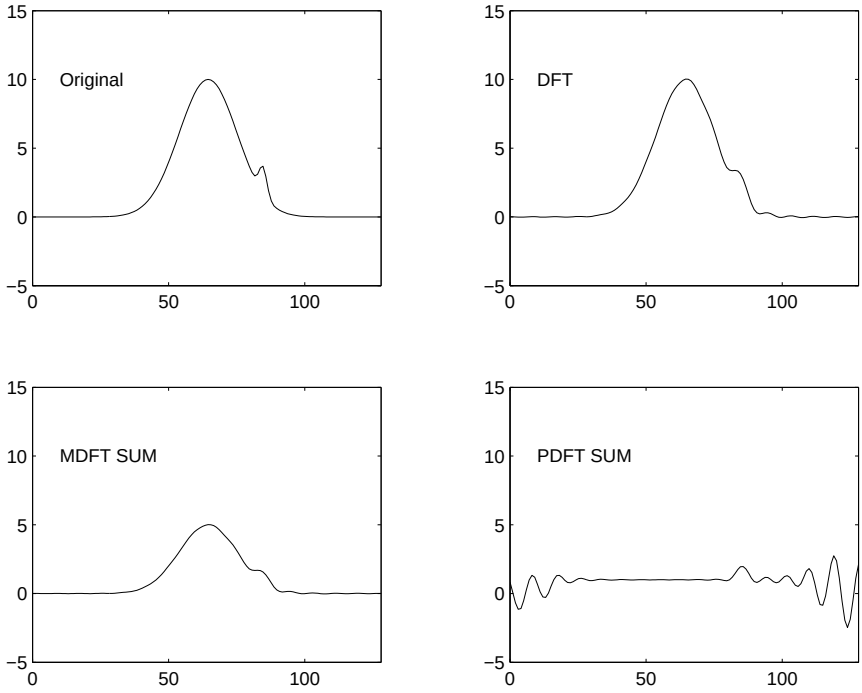



Figure 30.3: The polynomial terms in the DFT, MDFT and PDFT

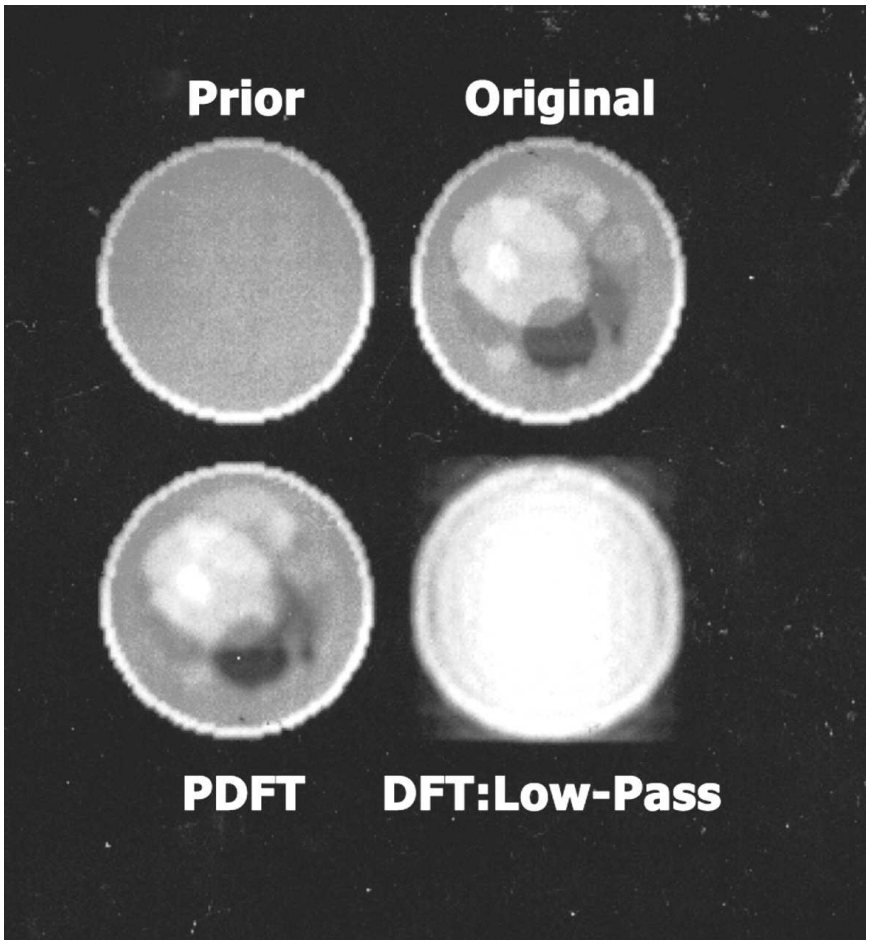



Figure 30.4: The PDFT in image reconstruction

# Chapter 31

# More on Bandlimited Extrapolation

Let our data be $f(x_m)$, $m = 1, ..., M$, where the $x_m$ are arbitrary values of the variable $x$. If $F(\omega)$ is zero outside $[-\Omega, \Omega]$, then minimizing the energy over $[-\Omega, \Omega]$ subject to data consistency produces an estimate of the form

$$F_\Omega(\omega) = \chi_\Omega(\omega) \sum\nolimits_{m=1}^{M} b_m \exp(ix_m\omega),$$

with the $b_m$ satisfying the equations

$$f(x_n) = \sum\nolimits_{m=1}^{M} b_m \frac{\sin(\Omega(x_m - x_n))}{\pi(x_m - x_n)},$$

for $n = 1, ..., M$. The matrix $S_\Omega$ with entries $\frac{\sin(\Omega(x_m - x_n))}{\pi(x_m - x_n)}$ we call a *sinc* matrix.

Although it seems reasonable that incorporating the additional information about the support of $F(\omega)$ should improve the estimation, it would be more convincing if we had a more mathematical argument to make. For that we turn to an analysis of the eigenvectors of the sinc matrix.

**Exercise 1:** The purpose of this exercise is to show that, for an Hermitian nonnegative-definite $M$ by $M$ matrix $Q$, a norm-one eigenvector $\mathbf{u}^1$ of $Q$ associated with its largest eigenvalue, $\lambda_1$, maximizes the quadratic form $\mathbf{a}^\dagger Q\mathbf{a}$ over all vectors $\mathbf{a}$ with norm one. Let $Q = ULU^\dagger$ be the eigenvector decomposition of $Q$, where the columns of $U$ are mutually orthogonal eigenvectors $\mathbf{u}^n$ with norms equal to one, so that $U^\dagger U = I$, and $L = diag\{\lambda_1, ..., \lambda_M\}$ is the diagonal matrix with the eigenvalues of $Q$ as its entries along the main

diagonal. Assume that $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_M$. Then maximize

$$\mathbf{a}^\dagger Q\mathbf{a} = \sum_{n=1}^{M} \lambda_n \, |\mathbf{a}^\dagger \mathbf{u}^n|^2,$$

subject to the constraint

$$\mathbf{a}^\dagger \mathbf{a} = \mathbf{a}^\dagger U^\dagger U\mathbf{a} = \sum_{n=1}^{M} |\mathbf{a}^\dagger \mathbf{u}^n|^2 = 1.$$

Hint: Show $\mathbf{a}^\dagger Q\mathbf{a}$ is a convex combination of the eigenvalues of $Q$.

**Exercise 2:** Show that for the sinc matrix $Q = S_\Omega$ the quadratic form $\mathbf{a}^\dagger Q\mathbf{a}$ in the previous exercise becomes

$$\mathbf{a}^\dagger S_\Omega \mathbf{a} = \frac{1}{2\pi} \int_{-\Omega}^{\Omega} |\sum\nolimits_{n=1}^{M} a_n e^{in\omega}|^2 d\omega.$$

Show that the norm of the vector $\mathbf{a}$ is the integral

$$\frac{1}{2\pi} \int_{-\pi}^{\pi} |\sum\nolimits_{n=1}^{M} a_n e^{in\omega}|^2 d\omega.$$

**Exercise 3:** For $M = 30$ compute the eigenvalues of the matrix $S_\Omega$ for various choices of $\Omega$, such as $\Omega = \frac{\pi}{k}$, for $k = 2, 3, ..., 10$. For each $k$ arrange the set of eigenvalues in decreasing order and note the proportion of them that are not near zero. The set of eigenvalues of a matrix is sometimes called its *eigenspectrum* and the nonnegative function $\chi_\Omega(\omega)$ is a power spectrum; here is one time in which different notions of a *spectrum* are related.

Suppose that the vector $\mathbf{u}^1 = (u_1^1, ..., u_M^1)^T$ is an eigenvector of $S_\Omega$ corresponding to the largest eigenvalue, $\lambda_1$. Associate with $\mathbf{u}^1$ the function

$$U^1(\omega) = \sum\nolimits_{n=1}^{M} u_n^1 e^{in\omega}.$$

Then

$$\lambda_1 = \int_{-\Omega}^{\Omega} |U^1(\omega)|^2 d\omega / \int_{-\pi}^{\pi} |U^1(\omega)|^2 d\omega$$

and $U^1(\omega)$ is the function of its form that is most concentrated within the interval $[-\Omega, \Omega]$.

Similarly, if $\mathbf{u}^M$ is an eigenvector of $S_\Omega$ associated with the smallest eigenvalue $\lambda_M$, then the corrsponding function $U^M(\omega)$ is the function of its form least concentrated in the interval $[-\Omega, \Omega]$.

**Exercise 4:** Plot for $|\omega| \leq \pi$ the functions $|U^m(\omega)|$ corresponding to each of the eigenvectors of the sinc matrix $S_\Omega$. Pay particular attention to the places where each of these functions is zero.

The eigenvectors of $S_\Omega$ corresponding to different eigenvalues are orthogonal, that is $(\mathbf{u}^m)^\dagger \mathbf{u}^n = 0$ if $m$ is not $n$. We can write this in terms of integrals:

$$\int_{-\pi}^{\pi} U^n(\omega)\overline{U^m(\omega)}d\omega = 0$$

if $m$ is not $n$. The mutual orthogonality of these functions is related to the locations of their roots, which were studied in the previous exercise.

Any Hermitian matrix $Q$ is invertible if and only if none of its eigenvalues is zero. With $\lambda_m$ and $\mathbf{u}^m$, $m = 1, ..., M$ the eigenvalues and eigenvectors of $Q$ the inverse of $Q$ can then be written as

$$Q^{-1} = (1/\lambda_1)\mathbf{u}^1(\mathbf{u}^1)^\dagger + ... + (1/\lambda_M)\mathbf{u}^M(\mathbf{u}^M)^\dagger.$$

**Exercise 5:** Show that the MDFT estimator (29.7) $F_\Omega(\omega)$ can be written as

$$F_\Omega(\omega) = \chi_\Omega(\omega) \sum\nolimits_{m=1}^{M} \frac{1}{\lambda_m}(\mathbf{u}^m)^\dagger \mathbf{d}\, U^m(\omega),$$

where $\mathbf{d}$ is the data vector.

**Exercise 6:** Show that the DFT estimate of $F(\omega)$, restricted to the interval $[-\Omega, \Omega]$, is

$$F_{DFT}(\omega) = \chi_\Omega(\omega) \sum\nolimits_{m=1}^{M} (\mathbf{u}^m)^\dagger \mathbf{d}\, U^m(\omega).$$

From these two exercises we can learn why it is that the estimate $F_\Omega(\omega)$ resolves better than the DFT. The former makes more use of the functions $U^m(\omega)$ for higher values of $m$, since these are the ones for which $\lambda_m$ is closer to zero. Since those functions are the ones having most of their roots within the interval $[-\Omega, \Omega]$, they have the most flexibility within that region and are better able to describe those features in $F(\omega)$ that are not resolved by the DFT.

# Chapter 32

# The Phase Problem

In optical image processing and elsewhere we find that we are unable to measure the complex values of the inverse Fourier transform $f(x_m)$, but only the magnitudes $|f(x_m)|$. Estimating $F(\omega)$ from these magnitude-only values is called the *phase problem* [92], [79], [94], [131], [57]. Such problems can arise in optical imaging through turbulent atmosphere, for example [93]. One solution to the phase problem in crystallography led to a Nobel Prize in the early 1980's for Jerome Karle.

Assume throughout this chapter that $F(\omega) = 0$ for $|\omega| > \Omega$. We can select an arbitrary collection of phases $\theta_m$ to combine with the magnitudes, to form the complex *pseudo data* $|f(x_m)|e^{i\theta_m}$. If we have some idea of the proper choice of $\Omega$ we calculate the estimate $F_\Omega(\omega)$ corresponding to the pseudo-data and again monitor the energy integral. For good choices of the phases the energy should not be too large, while for inappropriate choices the energy should be much larger, particularly if the data is oversampled. In Figure 32.1 we see the MDFT energy as a function of $D$, where the object is the original in Figure 30.1. The data is $r(n)$, $|n| \leq 25$ and the perturbed data is $r(n)\exp(iDu(n))$ for $u(n)$ random in $[0, 1]$ and $D$ in $[0, 1]$. The reconstruction process can be implemented as an iterative optimization procedure, in which we select a new collection of phases at each step in such a way as to reduce the energy in the bandlimited extrapolation that results. In [43] we show how to do this in an efficient manner. When the extrapolation energy is sufficiently small, the resulting estimate is typically acceptable, particularly when the data is oversampled.

When we have only magnitude measurements we can at least be sure that if $|f(x_m)| = 0$ then $f(x_m) = 0$. This suggests that we might try to estimate the function $F(\omega)$ from the zeros of its inverse Fourier transform. In [138] we showed that this approach has some promise for solving the phase problem.

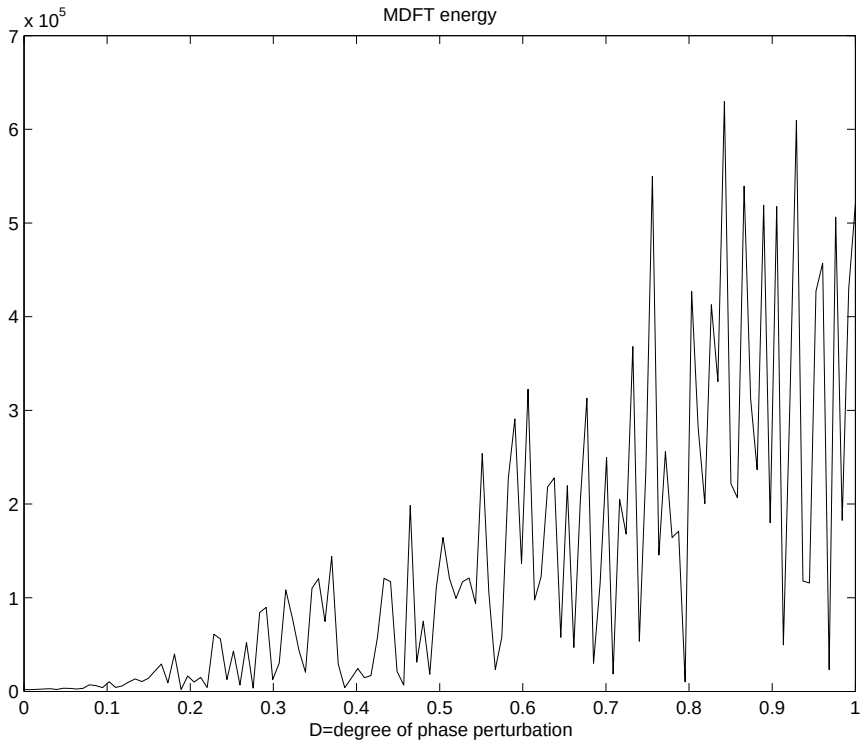



Figure 32.1: MDFT energy as a function of $D$

# Chapter 33

# A Little Matrix Theory

The 2 by 2 matrix $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ has an inverse

$$A^{-1} = \frac{1}{ad - bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

whenever the *determinant* of $A$, $\det(A) = ad - bc \neq 0$. More generally, associated with every complex square matrix is the complex number called its determinant, which is obtained from the entries of the matrix using formulas that can be found in any text on linear algebra. The significance of the determinant is that the matrix is invertible if and only if its determinant is not zero. This is of more theoretical than practical importance, since no computer can tell when a number is precisely zero.

Given $N$ by $N$ complex matrix $A$, we say that a complex number $\lambda$ is an *eigenvalue* of $A$ if there is a nonzero vector $\mathbf{u}$ with $A\mathbf{u} = \lambda\mathbf{u}$. The column vector $\mathbf{u}$ is then called an *eigenvector* of $A$ associated with eigenvalue $\lambda$; clearly, if $\mathbf{u}$ is an eigenvector of $A$, then so is $c\mathbf{u}$, for any constant $c \neq 0$. If $\lambda$ is an eigenvalue of $A$ then the matrix $A - \lambda I$ fails to have an inverse, since $(A - \lambda I)\mathbf{u} = \mathbf{0}$ but $\mathbf{u} \neq \mathbf{0}$. If we treat $\lambda$ as a variable and compute the determinant of $A - \lambda I$ we obtain a polynomial of degree $N$ in $\lambda$. Its roots $\lambda_1, ..., \lambda_N$ are then the eigenvalues of $A$. If $||\mathbf{u}||^2 = \mathbf{u}^\dagger\mathbf{u} = 1$ then $\mathbf{u}^\dagger A\mathbf{u} = \lambda\mathbf{u}^\dagger\mathbf{u} = \lambda$.

Suppose that $A\mathbf{x} = \mathbf{b}$ is a consistent linear system of $M$ equations in $N$ unknowns, where $M < N$. Then there are infinitely many solutions. A standard procedure in such cases is to find that solution $\mathbf{x}$ having the smallest norm

$$||\mathbf{x}|| = \sqrt{\sum_{n=1}^{N} |x_n|^2}.$$

As we shall see shortly, the *minimum norm* solution of $A\mathbf{x} = \mathbf{b}$ is a vector of the form $\mathbf{x} = A^\dagger\mathbf{z}$, where $A^\dagger$ denotes the conjugate transpose of the matrix

$A$. Then $A\mathbf{x} = \mathbf{b}$ becomes $AA^\dagger\mathbf{z} = \mathbf{b}$. Typically $(AA^\dagger)^{-1}$ will exist and we get $\mathbf{z} = (AA^\dagger)^{-1}\mathbf{b}$, from which it follows that the minimum norm solution is $\mathbf{x} = A^\dagger(AA^\dagger)^{-1}\mathbf{b}$. When $M$ and $N$ are not too large forming the matrix $AA^\dagger$ and solving for $\mathbf{z}$ is not prohibitively expensive and time-consuming. However, in image processing the vector $\mathbf{x}$ is often a vectorization of a two-dimensional (or even three-dimensional) image and $M$ and $N$ can be on the order of tens of thousands or more. The ART algorithm gives us a fast method for finding the minimum norm solution without computing $AA^\dagger$.

We begin by proving that the minimum norm solution of $A\mathbf{x} = \mathbf{b}$ has the form $\mathbf{x} = A^\dagger\mathbf{z}$ for some $M$-dimensional complex vector $\mathbf{z}$.

Let the *null space* of the matrix $A$ be all $N$-dimensional complex vectors $\mathbf{w}$ with $A\mathbf{w} = \mathbf{0}$. If $A\mathbf{x} = \mathbf{b}$ then $A(\mathbf{x} + \mathbf{w}) = \mathbf{b}$ for all $\mathbf{w}$ in the null space of $A$. If $\mathbf{x} = A^\dagger\mathbf{z}$ and $\mathbf{w}$ is in the null space of $A$ then

$$\begin{aligned} ||\mathbf{x} + \mathbf{w}||^2 = ||A^\dagger\mathbf{z} + \mathbf{w}||^2 = (A^\dagger\mathbf{z} + \mathbf{w})^\dagger(A^\dagger\mathbf{z} + \mathbf{w}) \\ = (A^\dagger\mathbf{z})^\dagger(A^\dagger\mathbf{z}) + (A^\dagger\mathbf{z})^\dagger\mathbf{w} + \mathbf{w}^\dagger(A^\dagger\mathbf{z}) + \mathbf{w}^\dagger\mathbf{w} \\ = ||A^\dagger\mathbf{z}||^2 + (A^\dagger\mathbf{z})^\dagger\mathbf{w} + \mathbf{w}^\dagger(A^\dagger\mathbf{z}) + ||\mathbf{w}||^2 \\ = ||A^\dagger\mathbf{z}||^2 + ||\mathbf{w}||^2, \end{aligned}$$

since

$$\mathbf{w}^\dagger(A^\dagger\mathbf{z}) = (A\mathbf{w})^\dagger\mathbf{z} = \mathbf{0}^\dagger\mathbf{z} = 0$$

and

$$(A^\dagger\mathbf{z})^\dagger\mathbf{w} = \mathbf{z}^\dagger A\mathbf{w} = \mathbf{z}^\dagger\mathbf{0} = 0.$$

Therefore $||\mathbf{x} + \mathbf{w}|| = ||A^\dagger\mathbf{z} + \mathbf{w}|| > ||A^\dagger\mathbf{z}|| = ||\mathbf{x}||$ unless $\mathbf{w} = \mathbf{0}$. This completes the proof.

**Exercise 1:** Show that if $\mathbf{z} = (z_1, ..., z_N)^T$ is a column vector with complex entries and $H = H^\dagger$ is an $N$ by $N$ Hermitian matrix with complex entries then the quadratic form $\mathbf{z}^\dagger H\mathbf{z}$ is a real number. Show that the quadratic form $\mathbf{z}^\dagger H\mathbf{z}$ can be calculated using only real numbers. Let $\mathbf{z} = \mathbf{x} + i\mathbf{y}$, with $\mathbf{x}$ and $\mathbf{y}$ real vectors and let $H = A + iB$, where $A$ and $B$ are real matrices. Then show that $A^T = A$, $B^T = -B$, $\mathbf{x}^T B\mathbf{x} = 0$ and finally,

$$\mathbf{z}^\dagger H\mathbf{z} = [\,\mathbf{x}^T \quad \mathbf{y}^T\,] \begin{bmatrix} A & -B \\ B & A \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix}.$$

Use the fact that $\mathbf{z}^\dagger H\mathbf{z}$ is real for every vector $\mathbf{z}$ to conclude that the eigenvalues of $H$ are real.

It can be shown that it is possible to find a set of $N$ mutually orthogonal eigenvectors of the Hermitian matrix $H$; call them $\{\mathbf{u}^1, ..., \mathbf{u}^N\}$. The matrix $H$ can then be written as

$$H = \sum\nolimits_{n=1}^{N} \lambda_n \mathbf{u}^n (\mathbf{u}^n)^\dagger,$$

a linear superposition of the *dyad* matrices $\mathbf{u}^n(\mathbf{u}^n)^\dagger$. We can also write $H = ULU^\dagger$, where $U$ is the matrix whose $n$-th column is the column vector $\mathbf{u}^n$ and $L$ is the diagonal matrix with the eigenvalues down the main diagonal and zero elsewhere.

The matrix $H$ is invertible if and only if none of the $\lambda$ are zero and its inverse is

$$H^{-1} = \sum_{n=1}^{N} \lambda_n^{-1} \mathbf{u}^n (\mathbf{u}^n)^\dagger.$$

We also have $H^{-1} = UL^{-1}U^\dagger$.

A Hermitian matrix $Q$ is said to be nonnegative- (positive-)definite if all the eigenvalues of $Q$ are nonnegative (positive). The matrix $Q$ is a nonnegative-definite matrix if and only if there is another matrix $C$ such that $Q = C^\dagger C$. Since the eigenvalues of $Q$ are nonnegative, the diagonal matrix $L$ has a square root, $\sqrt{L}$. Using the fact that $U^\dagger U = I$ we have

$$Q = ULU^\dagger = U\sqrt{L}U^\dagger U\sqrt{L}U^\dagger;$$

we then take $C = U\sqrt{L}U^\dagger$, so $C^\dagger = C$. Then $\mathbf{z}^\dagger Q\mathbf{z} = \mathbf{z}^\dagger C^\dagger C\mathbf{z} = ||C\mathbf{z}||^2$, so that $Q$ is positive-definite if and only if $C$ is invertible.

**Exercise 2:** Let $A$ be an $M$ by $N$ matrix with complex entries. View $A$ as a linear function with domain $C^N$, the space of all $N$-dimensional complex column vectors, and range contained within $C^M$, via the expression $A(\mathbf{x}) = A\mathbf{x}$. Suppose that $M > N$. The range of $A$, denoted $R(A)$, cannot be all of $C^M$. Show that every vector $\mathbf{z}$ in $C^M$ can be written uniquely in the form $\mathbf{z} = A\mathbf{x} + \mathbf{w}$, where $A^\dagger\mathbf{w} = \mathbf{0}$. Show that $\|\mathbf{z}\|^2 = \|A\mathbf{x}\|^2 + \|\mathbf{w}\|^2$, where $\|\mathbf{z}\|^2$ denotes the square of the norm of $\mathbf{z}$.

**Hint:** If $\mathbf{z} = A\mathbf{x} + \mathbf{w}$ then consider $A^\dagger\mathbf{z}$. Assume $A^\dagger A$ is invertible.

**Exercise 3:** When the complex $M$ by $N$ matrix $A$ is stored in the computer it is usually *vectorized*; that is, the matrix

$$A = \begin{bmatrix} A_{11} & A_{12} & \dots & A_{1N} \\ A_{21} & A_{22} & \dots & A_{2N} \\ \cdot & & & \\ \cdot & & & \\ \cdot & & & \\ A_{M1} & A_{M2} & \dots & A_{MN} \end{bmatrix}$$

becomes

$$\mathbf{vec}(A) = (A_{11}, A_{21}, ..., A_{M1}, A_{12}, A_{22}, ..., A_{M2}, ..., A_{MN})^T.$$

**a:** Show that the complex dot product $\mathbf{vec}(A)\cdot\mathbf{vec}(B) = \mathbf{vec}(B)^\dagger\mathbf{vec}(A)$ can be obtained by

$$\mathbf{vec}(A)\cdot\mathbf{vec}(B) = \text{trace}\,(AB^\dagger) = tr(AB^\dagger),$$

where, for a square matrix $C$, trace $(C)$ means the sum of the entries along the main diagonal of $C$. We can therefore use the trace to define an inner product between matrices: $< A, B >= \text{trace}\,(AB^\dagger)$.

**b:** Show that $\text{trace}\,(AA^\dagger) \geq 0$ for all $A$, so that we can use the trace to define a norm on matrices: $||A||^2 = \text{trace}\,(AA^\dagger)$.

**Exercise 4:** Let $B = ULD^\dagger$ be an $M$ by $N$ matrix in diagonalized form; that is, $L$ is an $M$ by $N$ diagonal matrix with entries $\lambda_1, ..., \lambda_K$ on its main diagonal, where $K = \min(M, N)$, and $U$ and $V$ are square matrices. Let the $n$th column of $U$ be denoted $\mathbf{u^n}$ and similarly for the columns of $V$. Such a diagonal decomposition occurs in the *singular value decomposition* (SVD). Show that we can write

$$B = \lambda_1\mathbf{u^1}(\mathbf{v^1})^\dagger + ... + \lambda_K\mathbf{u^K}(\mathbf{v^K})^\dagger.$$

If $B$ is an $N$ by $N$ Hermitian matrix then we can take $U = V$ and $K = M = N$, with the columns of $U$ the eigenvectors of $B$, normalized to have Euclidean norm equal to one, and the $\lambda_n$ to be the eigenvalues of $B$. In this case we may also assume that $U$ is a *unitary* matrix, that is, $UU^\dagger = U^\dagger U = I$, where $I$ denotes the identity matrix.

## Regularization of linear systems of equations:

A consistent linear system of equations $A\mathbf{x} = \mathbf{b}$ is *ill-conditioned* if small changes in the entries of vector $\mathbf{b}$ can result in large changes in the solution. Such situations are common in signal processing and are usually dealt with by regularization. We consider regularization in this subsection.

We assume, throughout this subsection, that $A$ is a real $M$ by $N$ matrix with full rank; then either $AA^T$ or $A^TA$ is invertible, whichever one has the smaller size.

**Exercise 5:** Show that the vector $\mathbf{x} = (x_1, ..., x_N)^T$ minimizes the mean squared error

$$\|A\mathbf{x} - \mathbf{b}\|^2 = \sum_{m=1}^{N}(Ax_m - b_m)^2,$$

if and only if $\mathbf{x}$ satisfies the system of linear equations $A^T(A\mathbf{x} - \mathbf{b}) = \mathbf{0}$, where $Ax_m = (A\mathbf{x})_m = \sum_{n=1}^{N} A_{mn}x_n$.

**Hint:** Calculate the partial derivatives of $\|A\mathbf{x} - \mathbf{b}\|^2$ with respect to each $x_n$.

**Exercise 6:** Let $\epsilon$ be in $(0, 1)$ and let $I$ be the identity matrix whose dimensions are understood from the context. Show that

$$((1-\epsilon)AA^T + \epsilon I)^{-1}A = A((1-\epsilon)A^TA + \epsilon I)^{-1},$$

and, taking transposes,

$$A^T((1-\epsilon)AA^T + \epsilon I)^{-1} = ((1-\epsilon)A^TA + \epsilon I)^{-1}A^T.$$

**Hint:** use the identity

$$A((1-\epsilon)A^TA + \epsilon I) = ((1-\epsilon)AA^T + \epsilon I)A.$$

**Exercise 7:** Show that any vector $\mathbf{p}$ in $R^N$ can be written as $\mathbf{p} = A^T\mathbf{q} + \mathbf{r}$, where $A\mathbf{r} = 0$.

We want to solve $A\mathbf{x} = \mathbf{b}$, at least in some approximate sense. Of course, there may be no solution, a unique solution or even multiple solutions. It often happens in applications that, even when there is an exact solution of $A\mathbf{x} = \mathbf{b}$, noise in the vector $\mathbf{b}$ makes such as exact solution undesirable; in such cases a *regularized solution* is usually used instead. Let $\epsilon > 0$ and define

$$F_\epsilon(x) = (1-\epsilon)\|A\mathbf{x} - \mathbf{b}\|^2 + \epsilon\|\mathbf{x} - \mathbf{p}\|^2.$$

**Exercise 8:** Show that $F_\epsilon$ always has a unique minimizer $\hat{\mathbf{x}}_\epsilon$ given by

$$\hat{\mathbf{x}}_\epsilon = ((1-\epsilon)A^TA + \epsilon I)^{-1}((1-\epsilon)A^T\mathbf{b} + \epsilon\mathbf{p});$$

this is a regularized solution of $A\mathbf{x} = \mathbf{b}$. Here $\mathbf{p}$ is a prior estimate of the desired solution. Note that the inverse above always exists.

What happens to $\hat{\mathbf{x}}_\epsilon$ as $\epsilon$ goes to zero? This will depend on which case we are in:

Case 1: $N \leq M$, $A^TA$ invertible; or

Case 2: $N > M$, $AA^T$ invertible.

**Exercise 9:** Show that, in Case 1, taking limits as $\epsilon \to 0$ on both sides of the expression for $\hat{\mathbf{x}}_\epsilon$ gives $\hat{\mathbf{x}}_\epsilon \to (A^TA)^{-1}A^T\mathbf{b}$, the least squares solution of $A\mathbf{x} = \mathbf{b}$.

We consider Case 2 now. Write $\mathbf{p} = A^T\mathbf{q} + \mathbf{r}$, with $A\mathbf{r} = \mathbf{0}$. Then

$$\hat{\mathbf{x}}_\epsilon = A^T((1-\epsilon)AA^T + \epsilon I)^{-1}((1-\epsilon)\mathbf{b} + \epsilon\mathbf{q}) + ((1-\epsilon)A^TA + \epsilon I)^{-1}(\epsilon\mathbf{r}).$$

**Exercise 10:** (a): Show that

$$((1-\epsilon)A^TA + \epsilon I)^{-1}(\epsilon\mathbf{r}) = \mathbf{r}, \forall\epsilon.$$

**Hint:** let

$$\mathbf{t}_\epsilon = ((1-\epsilon)A^TA + \epsilon I)^{-1}(\epsilon\mathbf{r}).$$

Then multiplying by $A$ gives

$$A\mathbf{t}_\epsilon = A((1-\epsilon)A^TA + \epsilon I)^{-1}(\epsilon\mathbf{r}).$$

Now show that $A\mathbf{t}_\epsilon = 0$.
(b): Now take the limit of $\hat{\mathbf{x}}_\epsilon$, as $\epsilon \to 0$, to get $\hat{\mathbf{x}}_\epsilon \to A^T(AA^T)^{-1}\mathbf{b} + \mathbf{r}$. Show that this is the solution of $A\mathbf{x} = \mathbf{b}$ closest to $\mathbf{p}$.
**Hint:** Draw a diagram for the case of one equation in two unknowns.

**Some useful matrix identities:** In the exercise that follows we consider several matrix identities that are useful in developing the Kalman filter.

**Exercise 11:** Establish the following identities, assuming that all the products and inverses involved are defined:

$$CDA^{-1}B(C^{-1} - DA^{-1}B)^{-1} = (C^{-1} - DA^{-1}B)^{-1} - C; \tag{33.1}$$

$$(A - BCD)^{-1} = A^{-1} + A^{-1}B(C^{-1} - DA^{-1}B)^{-1}DA^{-1}; \tag{33.2}$$

$$A^{-1}B(C^{-1} - DA^{-1}B)^{-1} = (A - BCD)^{-1}BC; \tag{33.3}$$

$$(A - BCD)^{-1} = (I + GD)A^{-1}, \tag{33.4}$$

for

$$G = A^{-1}B(C^{-1} - DA^{-1}B)^{-1}.$$

**Hints:** To get equation (33.1) use

$$C(C^{-1} - DA^{-1}B) = I - CDA^{-1}B.$$

For the second identity, multiply both sides of equation (33.2) on the left by $A - BCD$ and at the appropriate step use the identity (33.1). For (33.3) show that

$$BC(C^{-1} - DA^{-1}B) = B - BCDA^{-1}B = (A - BCD)A^{-1}B.$$

For (33.4), substitute what $G$ is and use (33.2).

# Chapter 34

# Matrix and Vector Calculus

As we saw in the previous chapter, the least squares approximate solution of $A\mathbf{x} = \mathbf{b}$ is a vector $\hat{\mathbf{x}}$ that minimizes the function $||A\mathbf{x} - \mathbf{b}||$. In our discussion of bandlimited extrapolation we showed that, for any nonnegative definite matrix $Q$, the vector having norm one that maximizes the quadratic form $\mathbf{x}^\dagger Q\mathbf{x}$ is an eigenvector of $Q$ associated with the largest eigenvalue. In the chapter on best linear unbiased optimization we seek a matrix that minimizes a certain function. All of these examples involve what we can call *matrix-vector calculus*; that is, the differentiation of a function with respect to a matrix or a vector. The gradient of a function of several variables is a well known example and we begin there. Since there is some possibility of confusion, for the rest of this chapter we follow the notational convention that $\mathbf{x}$ is a column vector and $x$ is a scalar.

### Differentiation with respect to a vector:

Let $\mathbf{x} = (x_1, ..., x_N)^T$ be an $N$-dimensional real column vector. Let $z = f(\mathbf{x})$ be a real-valued function of the entries of $\mathbf{x}$. The derivative of $z$ with respect to $\mathbf{x}$, also called the *gradient* of $z$, is the column vector

$$\frac{\partial z}{\partial \mathbf{x}} = \mathbf{a} = (a_1, ..., a_N)^T$$

with entries

$$a_n = \frac{\partial z}{\partial x_n}.$$

**Exercise 1:** Let $\mathbf{y}$ be a fixed real column vector and $z = f(\mathbf{x}) = \mathbf{y}^T\mathbf{x}$.

Show that

$$\frac{\partial z}{\partial \mathbf{x}} = \mathbf{y}.$$

**Exercise 2:** Let $Q$ be a real symmetric nonnegative definite matrix and let $z = f(\mathbf{x}) = \mathbf{x}^T Q\mathbf{x}$. Show that the gradient of this quadratic form is

$$\frac{\partial z}{\partial \mathbf{x}} = 2Q\mathbf{x}.$$

**Hint:** Write $Q$ as a linear combination of dyads involving the eigenvectors.

**Exercise 3:** Let $z = ||A\mathbf{x} - \mathbf{b}||^2$. Show that

$$\frac{\partial z}{\partial \mathbf{x}} = 2A^T A\mathbf{x} - 2A^T\mathbf{b}.$$

**Hint:** Use $z = (A\mathbf{x} - \mathbf{b})^T(A\mathbf{x} - \mathbf{b})$.

We can also consider the second derivative of $z = f(\mathbf{x})$, which is the *Hessian matrix* of $z$

$$\frac{\partial^2 z}{\partial \mathbf{x}^2} = A$$

with entries

$$A_{mn} = \frac{\partial^2 z}{\partial x_m \partial x_n}.$$

If the entries of the vector $\mathbf{z} = (z_1, ..., z_M)^T$ are real-valued functions of the vector $\mathbf{x}$ the derivative of $\mathbf{z}$ is the matrix whose $m$-th column is the derivative of the real-valued function $z_m$. This matrix is usually called the *Jacobian matrix* of $\mathbf{z}$. If $M = N$ the determinant of the Jacobian matrix is the *Jacobian*.

**Exercise 4:** Suppose $(u, v) = (u(x, y), v(x, y))$ is a change of variables from the Cartesian $(x, y)$ coordinate system to some other $(u, v)$ coordinate system. Let $\mathbf{x} = (x, y)^T$ and $\mathbf{z} = (u(\mathbf{x}), v(\mathbf{x}))^T$.

**a:** Calculate the Jacobian for the rectangular coordinate system obtained by rotating the $(x, y)$ system through an angle of $\theta$.

**b:** Calculate the Jacobian for the transformation from the $(x, y)$ system to polar coordinates.

## Differentiation with respect to a matrix:

Now we consider real-valued functions $z = f(A)$ of a real matrix $A$. As an example, for square matrices $A$ we have

$$z = f(A) = \text{trace}\,(A) = \sum_{n=1}^{N} A_{nn},$$

the sum of the entries along the main diagonal of $A$.

The derivative of $z = f(A)$ is the matrix

$$\frac{\partial z}{\partial A} = B$$

whose entries are

$$B_{mn} = \frac{\partial z}{\partial A_{mn}}.$$

**Exercise 5:** Show that the derivative of trace $(A)$ is $B = I$, the identity matrix.

**Exercise 6:** Show that the derivative of $z = \text{trace}\,(DAC)$ with respect to $A$ is

$$\frac{\partial z}{\partial A} = D^T C^T. \tag{34.1}$$

We note in passing that the derivative of $\det(DAC)$ with respect to $A$ is the matrix $\det(DAC)(A^{-1})^T$.

Although the trace is not independent of the order of the matrices in a product, it is independent of cyclic permutation of the factors:

$$\text{trace}\,(ABC) = \text{trace}\,(CAB) = \text{trace}\,(BCA).$$

Therefore the trace is independent of the order for the product of two matrices:

$$\text{trace}\,(AB) = \text{trace}\,(BA).$$

From this fact we conclude that

$$\mathbf{x}^T\mathbf{x} = \text{trace}\,(\mathbf{x}^T\mathbf{x}) = \text{trace}\,(\mathbf{x}\mathbf{x}^T).$$

If $\mathbf{x}$ is a random vector with correlation matrix

$$R = E(\mathbf{x}\mathbf{x}^T)$$

then

$$E(\mathbf{x}^T\mathbf{x}) = E(\text{trace}\,(\mathbf{x}\mathbf{x}^T)) = \text{trace}\,(E(\mathbf{x}\mathbf{x}^T)) = \text{trace}\,(R).$$

We shall use this trick in the chapter on detection.

**Exercise 7:** Let $z = \text{trace}\,(A^T C A)$. Show that the derivative of $z$ with respect to the matrix $A$ is

$$\frac{\partial z}{\partial A} = CA + C^T A. \tag{34.2}$$

Therefore, if $C = Q$ is symmetric, then the derivative is $2QA$.

We have restricted the discussion here to real matrices and vectors. It often happens that we want to optimize a real quantity with respect to a complex vector. We can rewrite such quantities in terms of the real and imaginary parts of the complex values involved, to reduce everything to the real case just considered. For example, let $Q$ be a hermitian matrix; then the quadratic form $\mathbf{k}^\dagger Q\mathbf{k}$ is real, for any complex vector $\mathbf{k}$. As we saw in an earlier exercise, we can write the quadratic form entirely in terms of real matrices and vectors.

If $w = u + iv$ is a complex number with real part $u$ and imaginary part $v$ the function $z = f(w) = |w|^2$ is real-valued. The derivative of $z = f(w)$ with respect to the complex variable $w$ does not exist. When we write $z = u^2 + v^2$ we consider $z$ as a function of the real vector $\mathbf{x} = (u, v)^T$. The derivative of $z$ with respect to $\mathbf{x}$ is the vector $(2u, 2v)^T$.

Similarly, when we consider the real quadratic form $\mathbf{k}^\dagger Q\mathbf{k}$, we view each of the complex entries of the $N$ by 1 vector $\mathbf{k}$ as two real numbers forming a two-dimensional real vector. We then differentiate the quadratic form with respect to the $2N$ by 1 real vector formed from these real and imaginary parts. If we turn the resulting $2N$ by 1 real vector back into an $N$ by 1 complex vector, we get $2Q\mathbf{k}$ as the derivative; so it appears as if the formula for differentiating in real case carries over to the complex case.

# Chapter 35

# The Singular Value Decomposition

We saw earlier that an $N$ by $N$ Hermitian matrix $H$ can be written in terms of its eigenvalues and eigenvectors as $H = ULU^\dagger$ or as

$$H = \sum_{n=1}^{N} \lambda_n \mathbf{u^n} (\mathbf{u^n})^\dagger.$$

The *singular value decomposition* (SVD) is a similar result that applies to any rectangular matrix. It is an important tool in image compression and pseudo-inversion.

Let $C$ be any $N$ by $K$ complex matrix. In presenting the SVD of $C$ we shall assume that $K \geq N$; the SVD of $C^\dagger$ will come from that of $C$. Let $A = C^\dagger C$ and $B = CC^\dagger$; we assume, reasonably, that $B$, the smaller of the two matrices, is invertible, so all the eigenvalues $\lambda_1, ..., \lambda_N$ of $B$ are positive. Then write the eigenvalue/eigenvector decomposition of $B$ as $B = ULU^\dagger$.

**Exercise 1:** Show that the nonzero eigenvalues of $A$ and $B$ are the same.

Let $V$ be the $K$ by $K$ matrix whose first $N$ columns are those of the matrix $C^\dagger U L^{-1/2}$ and whose remaining $K - N$ columns are any mutually orthogonal norm-one vectors that are all orthogonal to each of the first $N$ columns. Let $M$ be the $N$ by $K$ matrix with diagonal entries $M_{nn} = \sqrt{\lambda_n}$ for $n = 1, ..., N$ and whose remaining entries are zero. The nonzero entries of $M$, $\sqrt{\lambda_n}$, are called the *singular values* of $C$. The *singular value decomposition* (SVD) of $C$ is $C = UMV^\dagger$. The SVD of $C^\dagger$ is $C^\dagger = VM^T U^\dagger$.

**Exercise 2:** Show that $UMV^\dagger$ equals $C$.

Using the SVD of $C$ we can write

$$C = \sum_{n=1}^{N} \sqrt{\lambda_n} \mathbf{u^n} (\mathbf{v^n})^\dagger,$$

where $\mathbf{v^n}$ denotes the $n$-th column of the matrix $V$.

In image processing matrices such as $C$ are used to represent discrete two-dimensional images, with the entries of $C$ corresponding to the grey level or color at each pixel. It is common to find that most of the $N$ singular values of $C$ are nearly zero, so that $C$ can be written approximately as a sum of far fewer than $N$ dyads; this is SVD image compression.

If $N \neq K$ then $C$ cannot have an inverse; it does, however, have a *pseudo-inverse*, $C^* = VM^*U^\dagger$, where $M^*$ is the matrix obtained from $M$ by taking the inverse of each of its nonzero entries and leaving the remaining zeros the same. The pseudo-inverse of $C^\dagger$ is

$$(C^\dagger)^* = (C^*)^\dagger = U(M^*)^T V^\dagger = U(M^\dagger)^* V^\dagger.$$

Some important properties of the pseudo-inverse are the following:

**a.** $CC^*C = C$;

**b.** $C^*CC^* = C^*$;

**c.** $(C^*C)^\dagger = C^*C$;

**d.** $(CC^*)^\dagger = CC^*$.

The pseudo-inverse of an arbitrary $I$ by $J$ matrix $G$ can be used in much the same way as the inverse of non-singular matrices to find approximate or xact solutions of systems of equations $G\mathbf{x} = \mathbf{d}$. The following examples illustrate this point.

**Exercise 3:** If $I > J$ the system $G\mathbf{x} = \mathbf{d}$ probably has no exact solution. Show that whenever $G^\dagger G$ is invertible the pseudo-inverse of $G$ is $G^* = (G^\dagger G)^{-1} G^\dagger$ so that the vector $\mathbf{x} = G^*\mathbf{d}$ is the least squares approximate solution.

**Exercise 4:** If $I < J$ the system $G\mathbf{x} = \mathbf{d}$ probably has infinitely many solutions. Show that whenever the matrix $GG^\dagger$ is invertible the pseudo-inverse of $G$ is $G^* = G^\dagger (GG^\dagger)^{-1}$, so that the vector $\mathbf{x} = G^*\mathbf{d}$ is the exact solution of $G\mathbf{x} = \mathbf{d}$ closest to the origin; that is, it is the minimum norm solution.

# Chapter 36

# Projection onto Convex Sets

In [185] Youla suggests that problems in signal processing and image restoration might be viewed geometrically and the method of projection onto convex sets (POCS) employed to solve such inverse problems. In the survey paper [186] he examines the POCS method as a particular case of iterative algorithms for finding fixed points of nonexpansive mappings. This point of view is increasingly important in applications such as medical imaging and a number of recent papers have addressed the theoretical and practical issues involved [9], [10], [8], [35], [39], [42], [70], [71], [73].

A subset $C$ of $R^N$ is *convex* if the line segment joining any two of its members lies entirely within $C$. In the plane $R^2$ the set $C$ of all points whose distance to the origin is less than one is convex; if we include the boundary of $C$, that is, the circumference of the circle, the set is also *closed*. But the circumference alone is not a convex set. If $C$ is a closed convex set and $\mathbf{x}$ is not in $C$, then there ia a unique point in $C$ closer to $\mathbf{x}$ than any other member of $C$; that point is called the *metric projection* of $\mathbf{x}$ onto $C$, written $P_C\mathbf{x}$. If the set is not convex there need not be a unique nearest point; the circle of radius one (not including the inside) is not convex, the origin is not in this set and every point on the circumference is the same distance from the origin, so there is no unique point nearest to the origin. Examples of closed convex sets include $R^N_+$, the set of all real $N$-dimensional vectors having nonnegative entries; the set of all $\mathbf{x}$ whose norm does not exceed a given value $r > 0$; the set of all $\mathbf{x}$ such that $A\mathbf{x} \leq \mathbf{b}$, for a given matrix $A$ and given vector $\mathbf{b}$; and the set of all real vectors $\mathbf{x}$ with entries $x_n$ in the interval $[\alpha_n, \beta_n]$, for each $n$.

In this geometric approach the restored $N$-dimensional signal or image is a solution of the *convex feasibility problem* (CFP), that is, it lies within the

intersection of finitely many closed nonempty convex sets $C_m, m = 1, ..., M$, in $R^N$ (or sometimes, in infinite dimensional Hilbert space, when we talk about functions, instead of vectors).

For each vector $\mathbf{x}$ and each convex set $C$ the metric projection of $\mathbf{x}$ onto $C$ satisfies the inequality

$$(\mathbf{c} - P_C\mathbf{x}) \cdot (P_C\mathbf{x} - \mathbf{x}) \geq 0, \tag{36.1}$$

for any $\mathbf{c}$ in the set $C$. This just says that the angle between the vectors $\mathbf{c} - P_C\mathbf{x}$ and $P_C\mathbf{x} - \mathbf{x}$ does not exceed $\pi/2$, which happens because $C$ is convex (Draw a picture!).

The iterative methods used to solve the CFP employ these metric projections. Algorithms for solving the CFP are discussed in the papers cited above, as well as in the books by Censor and Zenios [63], Stark and Yang [170] and Borwein and Lewis [19].

The simplest example of the CFP is the solving of a system of linear equations $A\mathbf{x} = \mathbf{b}$. Let $A$ be an $M$ by $N$ real matrix and for $m = 1, ..., M$ let $B_m = \{\mathbf{x}|(A\mathbf{x})_m = b_m\}$, where $b_m$ denotes the $m$-th entry of the vector $\mathbf{b}$. Now let $C_m = B_m$. Any solution of $A\mathbf{x} = \mathbf{b}$ lies in the intersection of the $C_m$; if the system is inconsistent then the intersection is empty. The Kaczmarz algorithm [122] for solving the system of linear equations $A\mathbf{x} = \mathbf{b}$ has the iterative step

$$x_n^{k+1} = x_n^k + A_{m(k)n}(b_{m(k)} - (A\mathbf{x}^k)_{m(k)}), \tag{36.2}$$

for $n = 1, ..., N$, $k = 0, 1, ...$ and $m(k) = k(\mathrm{mod}\, M) + 1$. This algorithm was rediscovered by Gordon, Bender and Herman [102], who called it the *algebraic reconstruction technique* (ART). This algorithm is an example of the method of *successive orthogonal projections* (SOP) [105] whereby we generate the sequence $\{\mathbf{x}^k\}$ by taking $\mathbf{x}^{k+1}$ to be the point in $C_{m(k)}$ closest to $\mathbf{x}^k$. Kaczmarz's algorithm can also be viewed as a method for constrained optimization: whenever $A\mathbf{x} = \mathbf{b}$ has solutions, the limit of the sequence generated by equation (36.2) minimizes the function $||\mathbf{x} - \mathbf{x}^0||$ over all solutions of $A\mathbf{x} = \mathbf{b}$.

In the example just discussed the sets $C_m$ are hyperplanes in $R^N$; suppose now that we take the $C_m$ to be half-spaces and consider the problem of finding $\mathbf{x}$ such that $A\mathbf{x} \geq \mathbf{b}$. For each $m$ let $H_m$ be the half-space $H_m = \{\mathbf{x}|(A\mathbf{x})_m \geq b_m\}$. Then $\mathbf{x}$ will be in the intersection of the sets $C_m = H_m$ if and only if $A\mathbf{x} \geq \mathbf{b}$. Methods for solving this CFP, such as Hildreth's algorithm, are discussed in [63]. The Agmon-Motzkin-Schoenberg (AMS) algorithm [1] [145] for solving such systems of inequalities $A\mathbf{x} \geq \mathbf{b}$ has the iterative step

$$x_n^{k+1} = x_n^k + A_{m(k)n}(b_{m(k)} - (A\mathbf{x}^k)_{m(k)})_+, \tag{36.3}$$

where, for any real number $t$, the number $t_+$ is $t$ if $t \geq 0$ and 0 otherwise. The AMS algorithm converges to a solution of $A\mathbf{x} \geq \mathbf{b}$, if there are solutions. If there are no solutions the AMS algorithm converges cyclically, that is, subsequences associated with the same $m$ converge [84],[10].

The Gerchberg-Papoulis (GP) algorithm discussed earlier is another example of a POCS method. For any sequence of Fourier coefficients $g = \{g(n)\}$ let $Dg$ denote the sequence whose terms are $g(n)$ for $n \in \{M, M+1, ..., N\}$ and zero otherwise. Let $\mathcal{F}g = G$ be the operator taking a sequence of Fourier coefficients $g$ into the function

$$G(\omega) = \sum\nolimits_{n=-\infty}^{+\infty} g(n) \exp(in\omega),$$

for $\omega \in (-\pi, \pi)$. Let $\mathcal{H} = L^2(-\pi, \pi)$, $C_1 = L^2(-\Omega, \Omega)$ and $C_2$ the set of all members $G(\omega)$ of $\mathcal{H}$ whose Fourier coefficients satisfy $g(n) = f(n)$ for $n = M, M+1, ..., N$. The *metric projection* of a function $G(\omega) \in \mathcal{H}$ onto $C_1$ is $\chi_\Omega G(\omega)$; this is the function in $C_1$ closest to $G(\omega)$. The metric projection onto $C_2$ is implemented by passing from $G(\omega)$ to the sequence of its Fourier coefficients $\mathcal{F}^{-1}G = g$, then replacing those coefficients for $n = M, M+1, ..., N$ with $f(n)$ and calculating the resulting Fourier series; that is, the metric projection of $G$ onto $C_2$ is $\mathcal{F}(Df + (I - D)\mathcal{F}^{-1}G)$. The GP algorithm consists in alternating metric projections onto the two sets $C_1$ and $C_2$.

Algorithms for solving the CFP fall into two classes: those that employ all the sets $C_m$ at each step of the iteration (the so-called *simultaneous methods*) and those that do not (the *row-action algorithms* or, more generally, *block-iterative methods*).

In the consistent case, in which the intersection of the convex sets $C_m$ is nonempty, all reasonable algorithms are expected to converge to a member of that intersection; the limit may or may not be the member of the intersection closest to the starting vector $\mathbf{x}^0$. Figure 36.1 illustrates the method of alternating projection; note that the limit is not the point in the intersection nearest to the starting point.

In the inconsistent case, in which the intersection of the $C_m$ is empty, simultaneous methods typically converge to a minimizer of a *proximity function* [42], such as

$$f(\mathbf{x}) = \sum\nolimits_{m=1}^{M} ||\mathbf{x} - P_{C_m}\mathbf{x}||^2,$$

if a minimizer exists.

In the next chapter we consider an iterative POCS solution of the split feasibility problem.

In a later chapter we shall encounter the EMML and SMART algorithms. These algorithms can also be viewed as POCS methods, but with a twist. The projections onto convex sets that are involved there are with

respect to a different notion of distance between vectors; instead of the usual euclidean distance we use the cross-entropy distance.

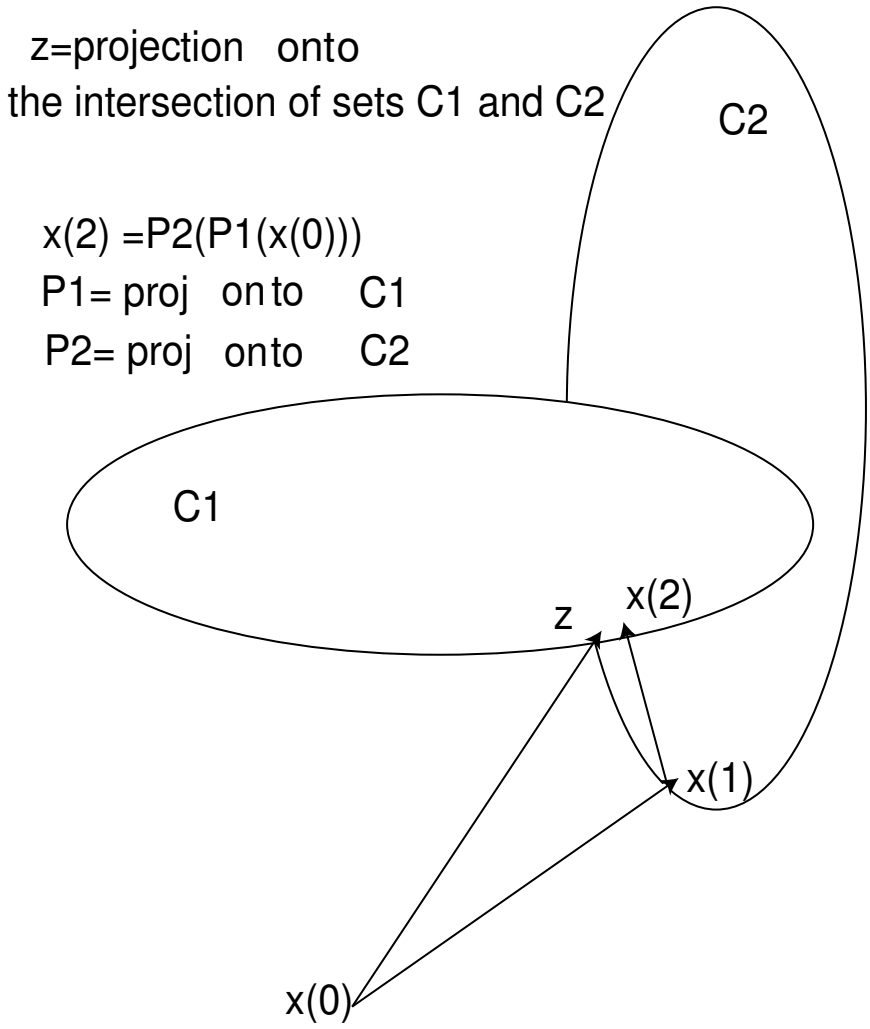



Figure 36.1: Alternating projections in POCS.

# Chapter 37

# The Split Feasibility Problem

In digital image processing it is typical to represent the image in vectorized form, as an $N$ by 1 column vector $\mathbf{x}$, where $N$ is the number of pixels we have chosen to use. The measured data pertaining to the image can then usually be represented as dot products of $\mathbf{x}$ with certain vectors $\mathbf{a}^m$, $m = 1, ..., M$; that is, the data is $b_m = \mathbf{a}^m \cdot \mathbf{x},$, for $m = 1, ..., M$. This problem is called *image reconstruction from projections*. With $\mathbf{b} = (b_1, ..., b_M)^T$ and $A$ the $M$ by $N$ matrix whose $m$-th row is the conjugate transpose of the column vector $\mathbf{a}^m$, we can write $A\mathbf{x} = \mathbf{b}$. Usually the measurements are noisy and we do not really want to solve this system of linear equations exactly; we might just want $A\mathbf{x}$ to be near $\mathbf{b}$, or perhaps we want $A\mathbf{x}$ to lie in a convex set $Q$ that may involve $\mathbf{b}$. We may also have additional information about the image that can be expressed by saying the $\mathbf{x}$ lies in some convex set $C$; for example, $\mathbf{x}$ may have nonnegative entries, so we would take $C$ to be the nonnegative cone in $N$-dimensional space. Such problems lead us to the split feasibility problem, which generalizes the problem of finding exact or approximate solutions of linear systems of equations.

The *split feasibility problem* (SFP) [59] is to find $\mathbf{c} \in C$ with $A\mathbf{c} \in Q$, if such points exist, where $A$ is a real $M$ by $N$ matrix and $C$ and $Q$ are nonempty, closed convex sets in $R^N$ and $R^M$, respectively. In [39] the CQ algorithm for solving the SFP was presented. The CQ algorithm has the iterative step

$$\mathbf{x}^{k+1} = P_C(\mathbf{x}^k - \gamma A^T(I - P_Q)A\mathbf{x}^k), \tag{37.1}$$

where $\gamma \in (0, 2/\rho(A^T A))$, for $\rho(A^T A)$ the spectral radius of the matrix $A^T A$, which is also its largest eigenvalue.

The CQ algorithm converges to a solution of the SFP, for any starting vector $x^0$, whenever the SFP has solutions. When the SFP has no solutions, the CQ algorithm converges to a minimizer of the function

$$f(\mathbf{x}) = \frac{1}{2}||P_Q A\mathbf{x} - A\mathbf{x}||^2$$

over the set $C$, provided such constrained minimizers exist. Therefore the CQ algorithm is an iterative constrained optimization method.

The function $f(\mathbf{x})$ is convex and differentiable on $R^N$ and its derivative is the operator

$$\nabla f(\mathbf{x}) = A^T(I - P_Q)A\mathbf{x}.$$

Let $B = P_C(I - \gamma A^T(I - P_Q)A)$. If $\gamma \in (0, 2/\lambda)$ the orbit sequence $\{B^k\mathbf{x}\}$ converges to a fixed point of $B$, whenever such points exist. If $\mathbf{z}$ is a fixed point of $B$, that is, $B\mathbf{z} = \mathbf{z}$, then $\mathbf{z} = P_C(\mathbf{z} - \gamma A^T(I - P_Q)A\mathbf{z})$. Therefore, according to the inequality (36.1), for any $\mathbf{c}$ in $C$ we have

$$(\mathbf{c} - \mathbf{z}) \cdot (\mathbf{z} - (\mathbf{z} - \gamma A^T(I - P_Q)A\mathbf{z})) \geq 0.$$

This tells us that

$$(\mathbf{c} - \mathbf{z}) \cdot (A^T(I - P_Q)A\mathbf{z}) = (\mathbf{c} - \mathbf{z}) \cdot \nabla f(\mathbf{z}) \geq 0,$$

which means that $\mathbf{z}$ minimizes $f(\mathbf{x})$ relative to $\mathbf{x}$ in the set $C$.

The $CQ$ algorithm employs the relaxation parameter $\gamma$ in the interval $(0, 2/L)$, where $L$ is the largest eigenvalue of the matrix $A^TA$, or, equivalently, the square of the largest singular value of $A$. Choosing the best relaxation parameter in any algorithm is a nontrivial procedure. Generally speaking, we want to select $\gamma$ near to $1/L$. In practice, it would be helpful to have a quick method for estimating $L$. In [39] we presented such a method that is particularly useful for sparse matrices. In the next chapter we take a look at that method for estimating $L$.

A number of well known iterative algorithms, such as the Landweber [130] and projected Landweber methods (see [12]), are particular cases of the CQ algorithm. The Gerchberg-Papoulis algorithm is, in turn, a particular case of the Landweber method.

## The Landweber algorithms

It is easy to find important examples of the SFP: if $C \subseteq R^N$ and $Q = \{\mathbf{b}\}$ then solving the SFP amounts to solving the linear system of equations $A\mathbf{x} = \mathbf{b}$; if $C$ is a proper subset of $R^N$, such as the nonnegative cone, then we seek solutions of $A\mathbf{x} = \mathbf{b}$ that lie within $C$, if there are any. The SFP is currently of some interest in dynamic PET medical image reconstruction, for reasons discussed in detail in [39]. Generally, we cannot solve the SFP in closed form and iterative methods are needed.

**The Landweber algorithm:** With $\mathbf{x}^0$ arbitrary and $k = 0, 1, ...$ let

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \gamma A^T(\mathbf{b} - A\mathbf{x}^k). \tag{37.2}$$

For general nonempty closed convex $C$ we obtain the projected Landweber method for finding a solution of $A\mathbf{x} = \mathbf{b}$ in $C$:

**The projected Landweber algorithm:** for $\mathbf{x}^0$ arbitrary and $k = 0, 1, ...$ let

$$\mathbf{x}^{k+1} = P_C(\mathbf{x}^k + \gamma A^T(\mathbf{b} - A\mathbf{x}^k)). \tag{37.3}$$

From the convergence theorem for the CQ algorithm it follows that the Landweber algorithm converges to a solution of $A\mathbf{x} = \mathbf{b}$ and the projected Landweber algorithm converges to a solution of $A\mathbf{x} = \mathbf{b}$ in $C$, whenever such solutions exist. When there are no solutions of the desired type, the Landweber algorithm converges to a least squares approximate solution of $A\mathbf{x} = \mathbf{b}$, while the projected Landweber method will converge to a minimizer, over the set $C$, of the function $||\mathbf{b} - A\mathbf{x}||$, whenever such a minimizer exists. Examples of the Landweber method include the Gerchberg-Papoulis iterative procedure for bandlimited extrapolation and super-resolution and the *simultaneous algebraic reconstruction technique* (SART) [3] for solving $A\mathbf{x} = \mathbf{b}$, for nonnegative matrix $A$.

**The SART algorithm:** Let $A$ be an $M$ by $N$ matrix with nonnegative entries. Let $A_{i+} > 0$ be the sum of the entries in the $i$th row of $A$ and $A_{+j} > 0$ be the sum of the entries in the $j$th column of $A$. Consider the (possibly inconsistent) system $A\mathbf{x} = \mathbf{b}$. The SART algorithm has the following iterative step:

$$\mathbf{x}_j^{k+1} = \mathbf{x}_j^k + \frac{1}{A_{+j}} \sum\nolimits_{i=1}^{M} (b_i - (A\mathbf{x}^k)_i)/A_{i+}.$$

We make the following changes of variables:

$$B_{ij} = A_{ij}/(A_{i+})^{1/2}(A_{+j})^{1/2},$$

$$z_j = x_j(A_{+j})^{1/2},$$

and

$$c_i = b_i/(A_{i+})^{1/2}.$$

Then the SART iterative step can be written as

$$\mathbf{z}^{k+1} = \mathbf{z}^k + B^T(\mathbf{c} - B\mathbf{z}^k).$$

This is a particular case of the Landweber algorithm, with $\gamma = 1$. The convergence of SART follows, once we know that the largest eigenvalue of $B^T B$ is less than two; in fact, we showed it is one [39].

# Chapter 38

# Singular Values of Sparse Matrices

In image reconstruction from projections the $M$ by $N$ matrix $A$ is usually quite large and often $\epsilon$-sparse, that is, most of its elements do not exceed $\epsilon$ in absolute value, where $\epsilon$ denotes a small positive quantity. In transmission tomography each column of $A$ corresponds to a single pixel in the digitized image, while each row of $A$ corresponds to a line segment through the object, along which an x-ray beam has travelled. The entries of a given row of $A$ are non-zero only for those columns whose associated pixel lies on that line segment; clearly most of the entries of any given row of $A$ will then be zero. In emission tomography the $I$ by $J$ nonnegative matrix $P$ has entries $P_{ij} \geq 0$; for each detector $i$ and pixel $j$ $P_{ij}$ is the probability that an emission at the $j$-th pixel will be detected at the $i$-th detector. When a detection is recorded at the $i$-th detector we want the likely source of the emission to be one of only a small number of pixels. For single photon emission tomography (SPECT) a collimator is used to permit detection of only those photons approaching the detector straight on. In positron emission tomography (PET) coincidence detection serves much the same purpose. In both cases the probabilities $P_{ij}$ will be zero (or nearly zero) for most combinations of $i$ and $j$. Such matrices are called *sparse* (or *almost sparse*). In this chapter we provide a convenient estimate for the largest singular value of an almost sparse matrix $A$, which, for notational convenience only, we take to be real.

In [39] it was shown that if $A$ is normalized so that each row has length one, then the spectral radius of $A^T A$, which is the square of the largest singular value of $A$ itself, does not exceed the maximum number of nonzero elements in any column of $A$. A similar upper bound on $\rho(A^T A)$ can be obtained for non-normalized, $\epsilon$-sparse $A$.

Let $A$ be an $M$ by $N$ matrix. For each $n = 1, ..., N$, let $s_n > 0$ be the number of nonzero entries in the $n$-th column of $A$ and let $s$ be the maximum of the $s_n$. Let $G$ be the $M$ by $N$ matrix with entries

$$G_{mn} = A_{mn}/(\sum_{l=1}^{N} s_l A_{ml}^2)^{1/2}.$$

Lent has shown that the eigenvalues of the matrix $G^T G$ do not exceed one [136]. This result suggested the following proposition, whose proof was given in [39].

**Proposition 38.1** *Let $A$ be an $M$ by $N$ matrix. For each $m = 1, ..., M$ let $\nu_m = \sum_{n=1}^{N} A_{mn}^2 > 0$. For each $n = 1, ..., N$ let $\sigma_n = \sum_{m=1}^{M} e_{mn}\nu_m$, where $e_{mn} = 1$ if $A_{mn} \neq 0$ and $e_{mn} = 0$ otherwise. Let $\sigma$ denote the maximum of the $\sigma_n$. Then the eigenvalues of the matrix $A^T A$ do not exceed $\sigma$. If $A$ is normalized so that the Euclidean length of each of its rows is one, then the eigenvalues of $A^T A$ do not exceed $s$, the maximum number of nonzero elements in any column of $A$.*

**Proof:** For simplicity, we consider only the normalized case; the proof for the more general case is similar.

Let $A^T A\mathbf{v} = c\mathbf{v}$ for some nonzero vector $\mathbf{v}$. We show that $c \leq s$. We have $AA^T A\mathbf{v} = cA\mathbf{v}$ and so $\mathbf{w}^T AA^T\mathbf{w} = \mathbf{v}^T A^T AA^T A\mathbf{v} = c\mathbf{v}^T A^T A\mathbf{v} = c\mathbf{w}^T\mathbf{w}$, for $\mathbf{w} = A\mathbf{v}$. Then, with $e_{mn} = 1$ if $A_{mn} \neq 0$ and $e_{mn} = 0$ otherwise, we have

$$(\sum_{m=1}^{M} A_{mn}w_m)^2 = (\sum_{m=1}^{M} A_{mn}e_{mn}w_m)^2$$

$$\leq (\sum_{m=1}^{M} A_{mn}^2 w_m^2)(\sum_{m=1}^{M} e_{mn}^2) =$$

$$(\sum_{m=1}^{M} A_{mn}^2 w_m^2)s_j \leq (\sum_{m=1}^{M} A_{mn}^2 w_m^2)s.$$

Therefore,

$$\mathbf{w}^T A^T A\mathbf{w} = \sum_{n=1}^{N}(\sum_{m=1}^{M} A_{mn}w_m)^2 \leq \sum_{n=1}^{N}(\sum_{m=1}^{M} A_{mn}^2 w_m^2)s,$$

and

$$\mathbf{w}^T A^T A\mathbf{w} = c\sum_{m=1}^{M} w_m^2 = c\sum_{m=1}^{M} w_m^2(\sum_{n=1}^{N} A_{mn}^2)$$

$$= c\sum_{m=1}^{M}\sum_{n=1}^{N} w_m^2 A_{mn}^2.$$

The result follows immediately. ∎

If we normalize $A$ so that its rows have length one, then the trace of the matrix $AA^T$ is $\mathrm{tr}(AA^T) = M$, which is also the sum of the eigenvalues of

$A^TA$. Consequently, the maximum eigenvalue of $A^TA$ does not exceed $M$; the result above improves that considerably, if $A$ is sparse and so $s << M$.

In image reconstruction from projection data that includes scattering we often encounter matrices $A$ most of whose entries are small, if not exactly zero. A slight modification of the proof above provides us with a useful upper bound for $L$, the largest eigenvalue of $A^TA$, in such cases. Assume that the rows of $A$ have length one. For $\epsilon > 0$ let $s$ be the largest number of entries in any column of $A$ whose magnitudes exceed $\epsilon$. Then we have

$$L \leq s + MN\epsilon^2 + 2\epsilon(MNs)^{1/2}.$$

The proof of this result is similar to that for the proposition above.

# Chapter 39

# Discrete Random Processes

The most common model used in signal processing is that of a sum of complex exponential functions plus noise. The noise is viewed as a sequence of random variables, and the signal components also may involve random parameters, such as random amplitudes and phase angles. Such models are best studied as *discrete random processes*.

A discrete random process is an infinite sequence $\{X_n\}_{n=-\infty}^{+\infty}$ in which each $X_n$ is a complex-valued random variable. The *autocorrelation function* associated with the random process is defined for all index values $m$ and $n$ by $r_x(m,n) = E(X_m\overline{X_n})$, where $E(\cdot)$ is the expectation or expected value operator. For $m = n$ we get $r(n,n) = \text{variance}(X_n)$. We say that the random process is *wide-sense stationary* if $E(X_n)$ is independent of $n$ and $r_x(m,n)$ is a function only of the difference, $m-n$, so that variance$(X_n)$ is independent of $n$. The autocorrelation function can then be redefined as $r_x(k) = E(X_{n+k}\overline{X_n})$. The *power spectrum* $R_x(\omega)$ of the random process is defined using the values $r_x(k)$ as its Fourier coeffcients:

$$R_x(\omega) = \sum_{k=-\infty}^{+\infty} r_x(k)e^{ik\omega},$$

for all $\omega$ in the interval $[-\pi,\pi]$. It can be proved that the power spectrum is a nonnegative function of the form $R_x(\omega) = |G(\omega)|^2$ and the autocorrelation sequence $\{r_x(k)\}$ satisfies the equations

$$r_x(k) = \sum_{n=-\infty}^{+\infty} g_{k+n}\overline{g_n},$$

for

$$G(\omega) = \sum_{n=-\infty}^{+\infty} g(n)e^{in\omega}.$$

In practice we will have actual values $X_n = x_n$, for only finitely many of the $X_n$, say for $n = 1, ..., m$. These can be used to estimate the values $r_x(k)$, at least for values of $k$ between, say, $-M/5$ and $M/5$. For example, we could estimate $r_x(k)$ by averaging all the products of the form $x_{k+m}\overline{x_m}$ that we can compute from the data. Clearly, as $k$ gets farther away from zero we have fewer such products, so our average is a less accurate estimate.

Once we have $r_x(k)$, $|k| \leq N$ we form the $N+1$ by $N+1$ *autocorrelation matrix* $R$ having the entries $R_{m,n} = r_x(m-n)$. This autocorrelation matrix is what is used in the design of optimal filtering.

The matrix $R$ is *Hermitian*, that is, $R_{n,m} = \overline{R_{m,n}}$, so that $R^\dagger = R$. An $M$ by $M$ Hermitian matrix $H$ is said to be *nonnegative-definite* if, for all complex column vectors $\mathbf{a} = (a_1, ..., a_M)^T$, the *quadratic form* $\mathbf{a}^\dagger H\mathbf{a}$ is a nonnegative number and *positive-definite* if such a quadratic form is always positive.

**Exercise 1:** Show that the autocorrelation matrix $R$ is nonnegative definite. Hint: Let

$$A(\omega) = \sum_{n=1}^{N+1} a_n e^{in\omega}$$

and express the integral

$$\int |A(\omega)|^2 R(\omega) d\omega$$

in terms of the $a_n$ and the $R_{m,n}$. Under what conditions can $R$ fail to be positive-definite?

Later we shall consider the *maximum entropy* method for estimating the power spectrum from finitely many values of $r_x(k)$.

**Autoregressive processes:** We noted at the beginning of the chapter that the case of a discrete-time signal with additive random noise provides a good example of a discrete random process; there are others. One particularly important type is the *autoregressive* (AR) process, which is closely related to ordinary linear differential equations.

When a smooth periodic function has noise added the new function is rough. Imagine, though, a fairly weighty pendulum of a clock, moving smoothly and periodically. Now imagine that a young child is throwing small stones at the bob of the pendulum. The movement of the pendulum is no longer periodic, but it is not rough. The pendulum is moving randomly in response to the random external disturbance, but not as if a random noise component has been added to its motion. To model such random processes we need to extend the notion of an ordinary differential equation. That leads us to the AR processes.

Recall that an ordinary linear $M$-th order differential equation with constant coefficients has the form

$$x^{(M)}(t) + c_1 x^{(M-1)}(t) + c_2 x^{(M-2)}(t) + ... + c_{M-1} x'(t) + c_M x(t) = f(t),$$

where $x^{(m)}(t)$ denotes the $m$-th derivative of the function $x(t)$ and the $c_m$ are constants. In many applications the variable $t$ is time and the function $f(t)$ is an external effect driving the linear system, with system response given by the unknown function $x(t)$. How the system responds to a variety of external drivers is of great interest. It is sometimes convenient to replace this continuous formulation with a discrete analog, called a *difference equation*.

In switching from differential equations to difference equations we discretize the time variable and replace the driving function $f(t)$ with $f_n$, $x(t)$ with $x_n$, the first derivative at time $t$, $x'(t)$, with the first difference, $x_n - x_{n-1}$, the second derivative $x''(t)$ with the second difference, $(x_n - x_{n-1}) - (x_{n-1} - x_{n-2})$, and so on. The differential equation is then replaced by the difference equation

$$x_n - a_1 x_{n-1} - a_2 x_{n-2} - ... - a_M x_{n-M} = f_n \tag{39.1}$$

for some constants $a_m$; the negative signs are a technical convenience only.

We now assume that the driving function is a discrete random process $\{f_n\}$, so that the system response becomes a discrete random process, $\{X_n\}$. If we assume that the driver $f_n$ is white noise, independent of the $\{X_n\}$, then the process $\{X_n\}$ is called an autoregressive (AR) process. What the system does at time $n$ depends partly on what it has done at the $M$ discrete times prior to time $n$, as well as what the external disturbance $f_n$ is at time $n$. Our goal is usually to determine the constants $a_m$; this is *system identification*. Our data is typically some number of consecutive measurements of the $X_n$.

Multiplying both sides of equation (39.1) by $\overline{X_{n-k}}$, for some $k > 0$ and taking the expected value, we obtain

$$E(X_n \overline{X_{n-k}}) - ... - a_M E(X_{n-M} \overline{X_{n-k}}) = 0.$$

or

$$r_x(k) - a_1 r_x(k-1) - ... - a_M r_x(k-M) = 0.$$

Taking $k = 0$ we get

$$r_x(0) - a_1 r_x(-1) - ... - a_M r_x(-M) = E(|f_n|^2) = \text{var}\,(f_n).$$

To find the $a_m$ we use the data to estimate $r_x(k)$ at least for $k = 0, 1, ..., M$. Then we use these estimates in the linear equations above, solving them for the $a_m$.

**Linear systems with random input:** In our discussion of discrete linear filters, also called time-invariant linear systems, we noted that it is common to consider as the input to such a system a discrete random process, $\{X_n\}$. The output is then another random process $\{Y_n\}$ given by

$$Y_n = \sum_{m=-\infty}^{+\infty} g_m X_{n-m},$$

for each $n$.

**Exercise 2:** Show that if the input process is wide-sense stationary then so is the output. Show that the power spectrum $R_y(\omega)$ of the output is

$$R_y(\omega) = |G(\omega)|^2 R_x(\omega).$$

# Chapter 40

# Prediction

An important problem in signal processing is the estimation of the next term in a sequence of numbers from knowledge of the previous values. This is called the *prediction problem*. The numbers might be the values at closing of a certain stock market index; knowing what has happened up to today, can we predict, with some accuracy, tomorrow's closing value? The numbers might describe the position in space of a missile; knowing where it has been for the past few minutes, can we predict where it will be for the next few? The numbers might be the noontime temperature in New York City on successive days; can we predict tomorrow's temperature from our knowledge of the temperatures on previous days? It is helpful, in weather prediction and elsewhere, to use not only the previous values of the sequence of interest, but those of related sequences; the recent temperatures in Pittsburgh might be helpful in predicting tomorrow's weather in New York City. In this chapter we begin a discussion of the prediction problem.

**Prediction through interpolation:** Suppose our data are the real numbers $x_1, ..., x_m$, corresponding to times $t = 1, ..., m$. Our goal is to estimate $x_{m+1}$. One way to do this is by interpolation.

A function $f(t)$ is said to interpolate the data if $f(n) = x_n$ for $n = 1, ..., m$. Having found such an interpolating function, we can take as our prediction of $x_{m+1}$ the number $\hat{x}_{m+1} = f(m + 1)$. Of course, there are infinitely many choices for the interpolating function $f(t)$. In our discussion of Fourier transform estimation we considered methods of interpolation that incorporated prior knowledge about the function being sampled, such as that it was bandlimited. In the absence of such additional information polynomial interpolation is one obvious choice.

Polynomial interpolation involves selecting as the function $f(t)$ the polynomial of least degree that interpolates the data. Given $m$ data points, we seek a polynomial of degree $m - 1$. Lagrange's method is a well known

procedure for solving this problem.

For $k = 1, ..., m$ let $L_k(t)$ be the unique polynomial with the properties $L_k(k) = 1$ and $L_k(n) = 0$ for $n = 1, ..., m$ and $n \neq k$. We can write each $L_k(t)$ explicitly, since we know its zeros:

$$L_k(t) = \frac{(t-1)\cdots(t-(k-1))(t-(k+1))\cdots(t-m)}{(k-1)\cdots(k-(k-1))(k-(k+1))\cdots(k-m)}.$$

Then the polynomial

$$P_m(t) = \sum_{k=1}^{m} x_k L_k(t)$$

is the interpolating polynomial we seek.

**Exercise 1:** Show that for $m = 1$ the predicted value of $x_2$ is $\hat{x}_2 = x_1$, so that

$$\hat{x}_2 - x_1 = 0.$$

This is the 'Tomorrow will be like today' prediction.

**Exercise 2:** Show that for $m = 2$ the predicted value of $x_3$ is $\hat{x}_3 = 2x_2 - x_1$, or $\hat{x}_3 - x_2 = (x_2 - x_1)$ so that

$$\hat{x}_3 - 2x_2 + x_1 = 0.$$

This prediction amounts to assuming the change from today to tomorrow will be the same as the change from yesterday to today; that is, we assume a constant slope.

**Exercise 3:** Show that for $m = 3$ the predicted value of $x_4$ is $\hat{x}_4 = 3x_3 - 3x_2 + x_1$, so that

$$\hat{x}_4 - 3x_3 + 3x_2 - x_1 = 0.$$

**Exercise 4:** The coefficients in the previous exercises fit a pattern. Using this pattern, determine the predicted value of $x_5$ for the case of $m = 4$. In general, what will be the predicted value of $x_{m+1}$ based on the $m$ previous values?

The concept of divided difference plays a significant role in interpolation, as we shall see.

**Divided differences:** The zeroth *divided difference* of a function $f(t)$ with respect to the point $t_0$ is $f[t_0] = f(t_0)$. The first divided difference with respect to the points $t_0$ and $t_1$ is

$$f[t_0, t_1] = \frac{f(t_1) - f(t_0)}{t_1 - t_0}.$$

The $m$th divided difference with respect to the points $t_0, ..., t_m$ is

$$f[t_0, ..., t_m] = \frac{f[t_1, ..., t_m] - f[t_0, ..., t_{m-1}]}{t_m - t_0}.$$

These quantities are discrete analogs of the derivatives of a function. Indeed, if $f(t)$ is a polynomial of degree at most $m-1$ then the $m$th divided difference is zero, for any points $t_0, ..., t_m$.

When the points $t_0, ..., t_m$ are consecutive integers the divided differences take on a special form. Suppose $t_0 = 1, t_1 = 2, ..., t_m = m+1$. Then

$$f[t_0, t_1] = f(2) - f(1);$$

$$f[t_0, t_1, t_2] = \frac{1}{2}(f(3) - 2f(2) + f(1));$$

$$f[t_0, t_1, t_2, t_3] = \frac{1}{6}(f(4) - 3f(3) + 3f(2) - f(1))$$

and so on, with each successive divided difference involving the coefficients in the expansion of the binomial $(a-b)^k$.

For each fixed value of $m \geq 1$ and $1 \leq n \leq m$ we have $f(n) = x_n$ and $f(m+1) = \hat{x}_{m+1}$. According to the exercises above, for $m = 1$ we can write

$$\hat{x}_2 - x_1 = 0,$$

which says that the first divided difference is zero; that is, $f[1, 2] = 0$. For $m = 2$ we have

$$[\hat{x}_3 - x_2] - [x_2 - x_1] = 0,$$

or $f[1, 2, 3] = 0$, so the second divided difference is zero. For $m = 3$

$$[[\hat{x}_4 - x_3] - [x_3 - x_2]] - [[x_3 - x_2] - [x_2 - x_1]] = 0,$$

which says that the third divided difference, $f[1, 2, 3, 4]$, is zero. The interpolation is achieved by assuming that the $m$ data points as well as the point to be interpolated lie on a polynomial of degree at most $m-1$. Under this assumption the $m$th divided difference with respect to the points $1, 2, ..., m+1$ would be zero. The interpolated value can then be calculated by setting the $m$th divided difference equal to zero, but replacing $x_{m+1}$ with the estimate $\hat{x}_{m+1}$.

The coefficients that occur in these various predictors are those in the expansion of the binomial $(a-b)^m$. To investigate this matter further, we define the *first difference operator* on an arbitrary sequence $x = \{x_n\}$ to be the operator $D$ such that $y = Dx$, where $y = \{y_n\}$ is the sequence with entries $y_n = x_n - x_{n-1}$. Notice that the operator $D$ can be written as $D = I - S$, where $I$ is the identity operator and $S$ is the shift operator; that is, $Sx = z$ where $z = \{z_n\}$ is the sequence with entries $z_n = x_{n-1}$.

The $k$-th difference operator is $D^k = (I - S)^k$; expanding this product in terms of powers of $S$ leads to the binomial coefficients that we saw earlier.

This method of predicting using the interpolating polynomial of degree $m-1$ will be perfectly accurate if the sequence $\{x_n\}$ is formed by taking values from a polynomial of degree $m-1$ or less. Typically, our data contains noise and interpolating the data exactly, while theoretically possible, is not wise or useful.

The prediction method used here is linear in the sense that our predicted value is a linear combination of the data values and the coefficients we use do not involve the data. Another approach, linear predictive coding, is somewhat different.

**Linear Predictive Coding:** Suppose once again that we have the data $x_1, ..., x_m$ and we want to predict $x_{m+1}$. Instead of using a linear combination of all the values $x_1, ..., x_m$ we choose to use as our prediction of $x_{m+1}$ a linear combination of $x_{m-p}, x_{m-p+1}, ..., x_m$, where $p$ is a positive integer much smaller than $m$. So our prediction has the form

$$\hat{x}_{m+1} = a_0 x_m + a_1 x_{m-1} + ... + a_p x_{m-p}.$$

To find the best coefficients $a_0, ..., a_p$ to use we imagine trying out each possible choice of coefficients, using them to predict data values we already know. Specifically, for each set of coefficients $\{a_0, ..., a_p\}$ we form the predictions

$$\hat{x}_{p+2} = a_0 x_{p+1} + a_1 x_p + a_2 x_{p-1} + ... + a_p x_1,$$

$$\hat{x}_{p+3} = a_0 x_{p+2} + a_1 x_{p+1} + a_2 x_p + ... + a_p x_2,$$

and so on, down to

$$\hat{x}_m = a_0 x_{m-1} + a_1 x_{m-2} + ... + a_p x_{m-(p+1)}.$$

Since we already know what the true values are, we can compare the predicted values with the true ones and then find the choice of coefficients that minimizes the average squared error. This amounts to finding the least squares solution of the system of equations obtained by replacing the predictions with the true values on the left side of the equations above:

$$\begin{bmatrix} x_{p+1} & x_p & ... & x_1 \\ x_{p+2} & x_{p+1} & ... & x_2 \\ . & & & \\ . & & & \\ . & & & \\ x_m & x_{m-1} & ... & x_{m-p-1} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ . \\ . \\ . \\ a_p \end{bmatrix} = \begin{bmatrix} x_{p+2} \\ x_{p+3} \\ . \\ . \\ . \\ x_m \end{bmatrix},$$

which we write as $G\mathbf{a} = \mathbf{b}$. Since $m$ is typically larger than $p$, this system is overdetermined. The least squares solution is

$$\mathbf{a} = (G^\dagger G)^{-1} G^\dagger \mathbf{b}.$$

The resulting set of coefficients is then used to make a linear combination of the values $x_m, ..., x_{m-p}$, which is then our predicted value. But note that although a linear combination of data forms the predicted value, the coefficients are determined from the data values themselves, so the overall method is nonlinear.

This method of prediction forms the basis of a data compression technique known as *linear predictive coding* (LPC). In many applications a long sequence of numbers has a certain amount of local redundancy and many of the values can be well predicted from a small number of previous ones, using the method just described. Instead of transmitting the entire sequence of numbers, only some of the numbers, along with the coefficients and occasional outliers, are sent.

The entry in the $k$th row, $n$th column of the matrix $G^\dagger G$ is

$$(G^\dagger G)_{kn} = \sum_{j=1}^{m-p} x_{p+1-k+j} \overline{x_{p+1-n+j}}.$$

If we view the data as values of a stationary random process, then the quantity $\frac{1}{m-p}(G^\dagger G)_{kn}$ is an estimate of the autocorrelation value $r_x(n-k)$. Similarly, the $k$th entry of the vector $G^\dagger \mathbf{b}$ is

$$(G^\dagger \mathbf{b})_k = \sum_{j=1}^{m-p} x_{p+1-k+j} \overline{x_{p+1+j}}$$

and $\frac{1}{m-p}(G^\dagger \mathbf{b})_k$ is an estimate of $r_x(-k)$, for $k = 1, ..., p+1$. This brings us to the problem of predicting the next value for a (possibly nonstationary) random process.

**Stochastic prediction:** In time series analysis similar linear prediction methods are studied. In that case the numbers $x_n$ are viewed as values of a discrete random process $\{X_n\}$. The coefficients are determined by considering the statistical description of how the random variable $X_{m+1}$ is related to the previous $X_n$. The prediction of $X_{m+1}$ is a linear combination of the random variables $X_n$, $n = 1, ..., m$,

$$\hat{X}_{m+1} = a_0 X_m + a_1 X_{m-1} + ... + a_{m-1} X_1,$$

with the coefficients determined using the orthogonality principle. Consequently, the coefficients satisfy the system of linear equations

$$E(X_{m+1}\overline{X_k}) = a_0 E(X_m \overline{X_k}) + ... + a_{m-1} E(X_1 \overline{X_k}),$$

for $k = 1, 2, ..., m$. The expected values in these equations are the autocorrelations associated with the random process.

**Prediction for an autoregressive process:** Suppose that the random process $\{X_n\}$ is an $M$th order AR process, so that

$$X_n - a_1 X_{n-1} - ... - a_M X_{n-M} = f_n,$$

where $\{f_n\}$ is white noise independent of the $\{X_n\}$.

**Exercise 5:** Use our earlier discussion of the relationship between the autocorrelation values $r_x(k)$ and the coefficients $a_m$ to show that the best linear predictor for the random variable $X_n$ in terms of the values of $X_{n-1}, ..., X_{n-M}$ is

$$\hat{X}_n = a_1 X_{n-1} + ... + a_M X_{n-M}$$

and the mean squared error is

$$E(|\hat{X}_n - X_n|^2) = \text{var}\,(f_n).$$

In fact, it can be shown that, because the process is an $M$th order AR process, this is the best linear predictor of $X_n$ in terms of the entire history of the process.

# Chapter 41

# Best Linear Unbiased Estimation

Detection is often like finding a needle in a haystack. One way to find the needle is to bring in some cows and have them eat the hay and leave the needle. Of course they would not be ordinary cows; they would be well trained to distinguish hay from needles. Because hay may vary in its length, shape, flavor, color, smell and so on, the cows need to learn what hay is like *on average*, with this statistical description broad enough to include almost any hay they are likely to encounter, but not so broad as to include needles. The more a needle looks, tastes or smells like hay the harder it is for the cows. The cows are not perfect. They may eat a needle now and then; we call this a *false negative*. They may fail to eat some hay, thinking it a needle; this is a false positive.

In most signal and image processing applications the measured data includes (or may include) a signal component we want and unwanted components called *noise*. Estimation involves determining the precise nature and strength of the signal component; deciding if that strength is zero or not is detection.

Noise often appears as an additive term, which we then try to remove. If we knew precisely the noisy part added to each data value we would simply subtract it; of course, we never have such information. How then do we remove something when we don't know what it is? Statistics provides a way out.

The basic idea in statistics is to use procedures that perform well on average, when applied to a class of problems. The procedures are built using properties of that class, usually involving probabilistic notions, and are evaluated by examining how they would have performed had they been applied to every problem in the class. To use such methods to remove

additive noise we need a description of the class of noises we expect to encounter, not specific values of the noise component in any one particular instance. We also need some idea about what signal components look like. In this chapter we discuss solving this noise removal problem using the *best linear unbiased estimation* (BLUE) . We begin with the simplest case and then proceed to discuss increasingly complex scenarios.

## The simplest problem:

Suppose our data is $z_j = c + v_j$, for $j = 1, ..., J$, where $c$ is an unknown constant to be estimated and the $v_j$ are additive noise. We assume that $E(v_j) = 0, E(v_j\overline{v_k}) = 0$, for $j \neq k$ and $E(|v_j|^2) = \sigma_j^2$. So the additive noises are assumed to have mean zero and to be independent (or at least uncorrelated). In order to estimate $c$ we adopt the following rules:

**a.** The estimate $\hat{c}$ is *linear* in the data $\mathbf{z} = (z_1, ..., z_J)^T$; that is, $\hat{c} = \mathbf{k}^\dagger\mathbf{z}$, for some vector $\mathbf{k} = (k_1, ..., k_J)^T$.

**b.** The estimate is *unbiased*; that is $E(\hat{c}) = c$. This means $\sum_{j=1}^J k_j = 1$.

**c.** The estimate is best in the sense that it minimizes the expected error squared; that is, $E(|\hat{c} - c|^2)$ is minimized.

The resulting vector $\mathbf{k}$ is calculated to be

$$k_i = \sigma_i^{-2}/(\sum_{j=1}^{J} \sigma_j^{-2})$$

and the BLUE estimator of $c$ is then

$$\hat{c} = \sum_{i=1}^{J} z_i\sigma_i^{-2}/(\sum_{j=1}^{J} \sigma_j^{-2}).$$

## The general case of the BLUE:

Suppose now that our data vector is $\mathbf{z} = H\mathbf{x}+\mathbf{v}$. Here $\mathbf{x}$ is a random vector whose value is to be estimated, the random vector $\mathbf{v}$ is additive noise whose mean is $E(\mathbf{v}) = 0$ and whose known covariance matrix is $Q = E(\mathbf{v}\mathbf{v}^\dagger)$, not necessarily diagonal, and the known matrix $H$ is $J$ by $N$, with $J > N$. Now we seek an estimate of the vector $\mathbf{x}$. The rules we use are now

**a.** The estimate $\hat{\mathbf{x}}$ must have the form $\hat{\mathbf{x}} = K^\dagger\mathbf{z}$, where the matrix $K$ is to be determined.

**b.** The estimate is unbiased; that is, $E(\hat{\mathbf{x}}) = E(\mathbf{x})$.

**c.** The $K$ is determined as the minimizer of the expected squared error; that is, once again we minimize $E(|\hat{\mathbf{x}} - \mathbf{x}|^2)$.

**Exercise 1:** Show that

$$E(|\hat{\mathbf{x}} - \mathbf{x}|^2) = \operatorname{trace} K^\dagger Q K.$$

**Hints:** Write the left side as

$$E(\operatorname{trace}((\hat{\mathbf{x}} - \mathbf{x})(\hat{\mathbf{x}} - \mathbf{x})^\dagger)).$$

Also use the fact that the trace and expected value operations commute.

**Exercise 2:** Show that for the estimator to be unbiased we need $K^\dagger H = I$, the identity matrix.

The problem then is to minimize trace $K^\dagger Q K$ subject to the constraint equation $K^\dagger H = I$. We solve this problem using a technique known as *prewhitening*.

Since the noise covariance matrix $Q$ is Hermitian and nonnegative definite, we have $Q = UDU^\dagger$, where the columns of $U$ are the (mutually orthogonal) eigenvectors of $Q$ and $D$ is a diagonal matrix whose diagonal entries are the (necessarily nonnegative) eigenvalues of $Q$; therefore, $U^\dagger U = I$. We call $C = UD^{1/2}U^\dagger$ the Hermitian square root of $Q$, since $C^\dagger = C$ and $C^2 = Q$. We assume that $Q$ is invertible, so that $C$ is also. Given the system of equations

$$\mathbf{z} = H\mathbf{x} + \mathbf{v},$$

as above, we obtain a new system

$$\mathbf{y} = G\mathbf{x} + \mathbf{w}$$

by multiplying both sides by $C^{-1} = Q^{-1/2}$; here $G = C^{-1}H$ and $\mathbf{w} = C^{-1}\mathbf{v}$. The new noise correlation matrix is

$$E(\mathbf{w}\mathbf{w}^\dagger) = C^{-1}QC^{-1} = I,$$

so the new noise is white. For this reason the step of multiplying by $C^{-1}$ is called *prewhitening*.

With $J = CK$ and $M = C^{-1}H$ we have

$$K^\dagger Q K = J^\dagger J$$

and

$$K^\dagger H = J^\dagger M.$$

Our problem then is to minimize trace $J^\dagger J$, subject to $J^\dagger M = I$.

Let $L = L^\dagger = (M^\dagger M)^{-1}$ and let $f(J)$ be the function

$$f(J) = \text{trace}[(J^\dagger - L^\dagger M^\dagger)(J - ML)].$$

The minimum value of $f(J)$ is zero, which occurs when $J = ML$. Note that this choice for $J$ has the property $J^\dagger M = I$. So minimizing $f(J)$ is equivalent to minimizing $f(J)$ subject to the constraint $J^\dagger M = I$ and both problems have the solution $J = ML$. But minimizing $f(J)$ subject to $J^\dagger M = I$ is equivalent to minimizing trace $J^\dagger J$ subject to $J^\dagger M = I$, which is our original problem. Therefore the optimal choice for $J$ is $J = ML$. Consequently the optimal choice for $K$ is

$$K = Q^{-1}HL = Q^{-1}H(H^\dagger Q^{-1}H)^{-1}.$$

and the BLUE estimate of $\mathbf{x}$ is

$$\mathbf{x}_{BLUE} = \hat{\mathbf{x}} = K^\dagger \mathbf{z} = (H^\dagger Q^{-1}H)^{-1}H^\dagger Q^{-1}\mathbf{z}.$$

The simplest case can be obtained from this more general formula by taking $N = 1$, $H = (1, 1, ..., 1)^T$ and $\mathbf{x} = c$.

Note that if the noise is *white*, that is, $Q = \sigma^2 I$, then $\hat{\mathbf{x}} = (H^\dagger H)^{-1}H^\dagger \mathbf{z}$, which is the least squares solution of the equation $\mathbf{z} = H\mathbf{x}$. The effect of requiring that the estimate be unbiased is that, in this case, we simply ignore the presence of the noise and calculate the least squares solution of the noise-free equation $\mathbf{z} = H\mathbf{x}$.

The BLUE estimator involves nested inversion, making it difficult to calculate, especially for large matrices. In the exercise that follows we discover an approximation of the BLUE that is easier to calculate.

**Exercise 3:** Show that for $\epsilon > 0$ we have

$$(H^\dagger Q^{-1}H + \epsilon I)^{-1}H^\dagger Q^{-1} = H^\dagger (HH^\dagger + \epsilon Q)^{-1}. \tag{41.1}$$

**Hint:** Use the identity

$$H^\dagger Q^{-1}(HH^\dagger + \epsilon Q) = (H^\dagger Q^{-1}H + \epsilon I)H^\dagger.$$

It follows from the identity (41.1) that

$$\mathbf{x}_{BLUE} = \lim_{\epsilon \to 0} H^\dagger (HH^\dagger + \epsilon Q)^{-1}\mathbf{z}. \tag{41.2}$$

Therefore we can get an approximation of the BLUE estimate by selecting $\epsilon > 0$ near zero, solving the system of linear equations

$$(HH^\dagger + \epsilon Q)\mathbf{a} = \mathbf{z}$$

for $\mathbf{a}$ and taking $\mathbf{x} = H^\dagger \mathbf{a}$.

## The BLUE with a prior estimate

In Kalman filtering we have the situation in which we want to estimate the random vector $\mathbf{x}$ given measurements $\mathbf{z} = H\mathbf{x} + \mathbf{v}$, but also given a prior estimate $\mathbf{y}$ of $\mathbf{x}$. It is the case there that $E(\mathbf{y}) = E(\mathbf{x})$, so we write $\mathbf{y} = \mathbf{x} + \mathbf{w}$, with $\mathbf{w}$ independent of both $\mathbf{x}$ and $\mathbf{v}$ and $E(\mathbf{w}) = \mathbf{0}$. The covariance matrix for $\mathbf{w}$ we denote by $E(\mathbf{w}\mathbf{w}^\dagger) = R$. We now require that the estimate $\hat{\mathbf{x}}$ be linear in both $\mathbf{z}$ and $\mathbf{y}$; that is, the estimate has the form

$$\hat{\mathbf{x}} = C^\dagger \mathbf{z} + D^\dagger \mathbf{y},$$

for matrices $C$ and $D$ to be determined.

The approach is to apply the BLUE to the combined system of linear equations

$$\mathbf{z} = H\mathbf{x} + \mathbf{v},$$

$$\mathbf{y} = \mathbf{x} + \mathbf{w}.$$

In matrix language this combined system becomes $\mathbf{u} = J\mathbf{x} + \mathbf{n}$, with $\mathbf{u}^T = [\mathbf{z}^T\ \mathbf{y}^T]$, $J^T = [H^T\ I^T]$ and $\mathbf{n}^T = [\mathbf{v}^T\ \mathbf{w}^T]$. The noise covariance matrix becomes

$$P = \begin{bmatrix} Q & 0 \\ 0 & R \end{bmatrix}.$$

The BLUE estimate is $K^\dagger \mathbf{u}$, with $K^\dagger J = I$. Minimizing the variance, we find that the optimal $K^\dagger$ is

$$K^\dagger = (J^\dagger P^{-1} J)^{-1} J^\dagger P^{-1}.$$

The optimal estimate is then

$$\hat{\mathbf{x}} = (H^\dagger Q^{-1} H + R^{-1})^{-1} (H^\dagger Q^{-1} \mathbf{z} + R^{-1} \mathbf{y}).$$

Therefore

$$C^\dagger = (H^\dagger Q^{-1} H + R^{-1})^{-1} H^\dagger Q^{-1}$$

and

$$D^\dagger = (H^\dagger Q^{-1} H + R^{-1})^{-1} R^{-1}.$$

Using the matrix identities in equations (33.2) and (33.3) we can rewrite this estimate in the more useful form

$$\hat{\mathbf{x}} = \mathbf{y} + G(\mathbf{z} - H\mathbf{y}),$$

for

$$G = RH^\dagger (Q + HRH^\dagger)^{-1}. \tag{41.3}$$

The covariance matrix of the optimal estimator is $K^\dagger PK$, which can be written as

$$K^\dagger PK = (R^{-1} + H^\dagger Q^{-1} H)^{-1} = (I - GH)R.$$

In the context of the Kalman filter $R$ is the covariance of the prior estimate of the current state, $G$ is the Kalman gain matrix and $K^\dagger PK$ is the posterior covariance of the current state. The algorithm proceeds recursively from one state to the next in time.

## Adaptive BLUE

We have assumed so far that we know the covariance matrix $Q$ corresponding to the measurement noise. If we do not, then we may attempt to estimate $Q$ from the measurements themselves; such methods are called *noise-adaptive*. To illustrate, let the *innovations* vector be $\mathbf{e} = \mathbf{z} - H\mathbf{y}$. Then the covariance matrix of $\mathbf{e}$ is $S = HRH^\dagger + Q$. Having obtained an estimate $\hat{S}$ of $S$ from the data, we use $\hat{S} - HRH^\dagger$ in place of $Q$ in equation (41.3).

In this chapter we have focused on the filtering problem: given the data vector $\mathbf{z}$, estimate $\mathbf{x}$, assuming that $\mathbf{z}$ consists of noisy measurements of $H\mathbf{x}$; that is, $\mathbf{z} = H\mathbf{x} + \mathbf{v}$. An important extension of this problem is that of stochastic prediction. In a later chapter we discuss the Kalman filter method for solving this more general problem.

# Chapter 42

# The BLUE and the Least Squares Estimators

As we saw in the previous chapter, the best linear unbiased estimate of $\mathbf{x}$, given the observed vector $\mathbf{z} = H\mathbf{x} + \mathbf{v}$, is

$$\mathbf{x}_{BLUE} = (H^\dagger Q^{-1} H)^{-1} H^\dagger Q^{-1} \mathbf{z}, \tag{42.1}$$

where $Q$ is the invertible covariance matrix of the mean zero noise vector $\mathbf{v}$ and $H$ is a $J$ by $N$ matrix with $J \geq N$ and $H^\dagger H$ invertible. Even if we know $Q$ exactly, the double inversion in equation (42.1) makes it difficult to calculate the BLUE estimate, especially for large vectors $\mathbf{z}$. It is often the case in practice that we do not know $Q$ precisely and must estimate or model it. Because good approximations of $Q$ do not necessarily lead to good approximations of $Q^{-1}$, the calculation of the BLUE is further complicated. For these reasons one may decide to use the least squares estimate

$$\mathbf{x}_{LS} = (H^\dagger H)^{-1} H^\dagger \mathbf{z} \tag{42.2}$$

instead. We are therefore led to consider when the two estimation methods produce the same answers; that is, when do we have

$$(H^\dagger H)^{-1} H^\dagger = (H^\dagger Q^{-1} H)^{-1} H^\dagger Q^{-1}. \tag{42.3}$$

In this chapter we state and prove a theorem that answers this question. The proof relies on the results of several exercises that involve basic facts from linear algebra.

**A little linear algebra:** We begin with some definitions. Let $S$ be a subspace of finite-dimensional Euclidean space $R^J$ and $Q$ a $J$ by $J$ Hermitian

matrix. We denote by $Q(S)$ the set

$$Q(S) = \{\mathbf{t} | \text{there exists } \mathbf{s} \in S \text{ with } \mathbf{t} = Q\mathbf{s}\},$$

and by $Q^{-1}(S)$ the set

$$Q^{-1}(S) = \{\mathbf{u} | Q\mathbf{u} \in S\}.$$

Note that the set $Q^{-1}(S)$ is defined whether or not $Q$ is invertible.

We denote by $S^{\perp}$ the set of vectors $\mathbf{u}$ that are orthogonal to every member of $S$; that is,

$$S^{\perp} = \{\mathbf{u} | \mathbf{u}^{\dagger}\mathbf{s} = 0, \text{for every } \mathbf{s} \in S\}.$$

Let $H$ be a $J$ by $N$ matrix. Then $CS(H)$, the column space of $H$, is the subspace of $R^J$ consisting of all the linear combinations of the columns of $H$. The null space of $H^{\dagger}$, denoted $NS(H^{\dagger})$, is the subspace of $R^J$ containing all the vectors $\mathbf{w}$ for which $H^{\dagger}\mathbf{w} = 0$.

**Exercise 1:** Show that $CS(H)^{\perp} = NS(H^{\dagger})$.

**Hint:** If $\mathbf{v} \in CS(H)^{\perp}$, then $\mathbf{v}^{\dagger}H\mathbf{x} = 0$ for all $\mathbf{x}$, including $\mathbf{x} = H^{\dagger}\mathbf{v}$.

**Exercise 2:** Show that $CS(H) \cap NS(H^{\dagger}) = \{\mathbf{0}\}$.

**Hint:** If $\mathbf{y} = H\mathbf{x} \in NS(H^{\dagger})$ consider $||\mathbf{y}||^2 = \mathbf{y}^{\dagger}\mathbf{y}$.

**Exercise 3:** Let $S$ be any subspace of $R^J$. Show that if $Q$ is invertible and $Q(S) = S$ then $Q^{-1}(S) = S$.

**Hint:** If $Q\mathbf{t} = Q\mathbf{s}$ then $\mathbf{t} = \mathbf{s}$.

**Exercise 4:** Let $Q$ be Hermitian. Show that $Q(S)^{\perp} = Q^{-1}(S^{\perp})$ for every subspace $S$. If $Q$ is also invertible then $Q^{-1}(S)^{\perp} = Q(S^{\perp})$. Find an example of a non-invertible $Q$ for which $Q^{-1}(S)^{\perp}$ and $Q(S^{\perp})$ are different.

We assume, for the remainder of this chapter, that $Q$ is Hermitian and invertible and that the matrix $H^{\dagger}H$ is invertible. Note that the matrix $H^{\dagger}Q^{-1}H$ need not be invertible under these assumptions. We shall denote by $S$ an arbitrary subspace of $R^J$.

**Exercise 5:** Show that $Q(S) = S$ if and only if $Q(S^{\perp}) = S^{\perp}$.

**Hint:** Use Exercise 4.

**Exercise 6:** Show that if $Q(CS(H)) = CS(H)$ then $H^\dagger Q^{-1} H$ is invertible.

**Hint:** Show that $H^\dagger Q^{-1} H\mathbf{x} = \mathbf{0}$ if and only if $\mathbf{x} = \mathbf{0}$. Recall that $Q^{-1}H\mathbf{x} \in CS(H)$, by Exercise 4. Then use Exercise 2.

**When are the BLUE and the LS estimator the same?**

We are looking for conditions on $Q$ and $H$ that imply equation (42.3), which we rewrite as

$$H^\dagger = (H^\dagger Q^{-1} H)(H^\dagger H)^{-1} H^\dagger Q \tag{42.4}$$

or

$$H^\dagger T\mathbf{x} = \mathbf{0}$$

for all $\mathbf{x}$, where

$$T = I - Q^{-1}H(H^\dagger H)^{-1}H^\dagger Q.$$

In other words, we want $T\mathbf{x} \in NS(H^\dagger)$ for all $\mathbf{x}$. The theorem is the following:

**Theorem 42.1** *We have $T\mathbf{x} \in NS(H^\dagger)$ for all $\mathbf{x}$ if and only if $Q(CS(H)) = CS(H)$.*

An equivalent form of this theorem was proven by Anderson in [2]; he attributes a portion of the proof to Magness and McQuire [139]. The proof we give here is due to Kheifets [126] and is much simpler than Anderson's proof. The proof of the theorem is simplified somewhat by first establishing the result in the next exercise.

**Exercise 7:** Show that if equation (42.4) holds then the matrix $H^\dagger Q^{-1} H$ is invertible.

**Hints:** Recall that we have assumed that $CS(H^\dagger) = R^J$ when we assumed that $H^\dagger H$ is invertible. From equation (42.4) it follows that $CS(H^\dagger Q^{-1} H) = R^J$.

**The proof of the theorem:** Assume first that $Q(CS(H)) = CS(H)$, which, as we now know, also implies $Q(NS(H^\dagger)) = NS(H^\dagger)$, as well as $Q^{-1}(CS(H)) = CS(H)$, $Q^{-1}(NS(H^\dagger)) = NS(H^\dagger)$ and the invertibility of the matrix $H^\dagger Q^{-1} H$. Every $\mathbf{x} \in R^J$ has the form $\mathbf{x} = H\mathbf{a} + \mathbf{w}$, for some $\mathbf{a}$ and $\mathbf{w} \in NS(H^\dagger)$. We show that $T\mathbf{x} = \mathbf{w}$, so that $T\mathbf{x} \in NS(H^\dagger)$ for all $\mathbf{x}$. We have

$$T\mathbf{x} = TH\mathbf{a} + T\mathbf{w} =$$

$$\mathbf{x} - Q^{-1}H(H^\dagger H)^{-1}H^\dagger QH\mathbf{a} - Q^{-1}H(H^\dagger H)^{-1}H^\dagger Q\mathbf{w}.$$

We know that $QH\mathbf{a} = H\mathbf{b}$ for some $\mathbf{b}$, so that $H\mathbf{a} = Q^{-1}H\mathbf{b}$. We also know that $Q\mathbf{w} = \mathbf{v} \in NS(H^\dagger)$, so that $\mathbf{w} = Q^{-1}\mathbf{v}$. Then, continuing our calculations, we have

$$T\mathbf{x} = \mathbf{x} - Q^{-1}H\mathbf{b} - \mathbf{0} = \mathbf{x} - H\mathbf{a} = \mathbf{w},$$

so $T\mathbf{x} \in NS(H^\dagger)$.

Conversely, suppose now that $T\mathbf{x} \in NS(H^\dagger)$ for all $\mathbf{x}$, which, as we have seen, is equivalent to equation (42.4). We show that $Q^{-1}(NS(H^\dagger) = NS(H^\dagger)$. First, let $\mathbf{v} \in Q^{-1}(NS(H^\dagger))$; we show $\mathbf{v} \in NS(H^\dagger)$. We have

$$H^\dagger\mathbf{v} = (H^\dagger Q^{-1}H)(H^\dagger H)^{-1}H^\dagger Q\mathbf{v},$$

which is zero, since $H^\dagger Q\mathbf{v} = \mathbf{0}$. So we have shown that $Q^{-1}(NS(H^\dagger)) \subseteq NS(H^\dagger)$. To complete the proof we take an arbitrary member $\mathbf{v}$ of $NS(H^\dagger)$ and show that $\mathbf{v}$ is in $Q^{-1}(NS(H^\dagger))$, that is, $Q\mathbf{v} \in NS(H^\dagger)$. We know that $Q\mathbf{v} = H\mathbf{a} + \mathbf{w}$, for $\mathbf{w} \in NS(H^\dagger)$ and

$$\mathbf{a} = (H^\dagger H)^{-1}H^\dagger Q\mathbf{v},$$

so that

$$H\mathbf{a} = H(H^\dagger H)^{-1}H^\dagger Q\mathbf{v}.$$

Then, using Exercise 7, we have

$$\begin{aligned} Q\mathbf{v} &= H(H^\dagger H)^{-1}H^\dagger Q\mathbf{v} + \mathbf{w} \\ &= H(H^\dagger Q^{-1}H)^{-1}H^\dagger Q^{-1}Q\mathbf{v} + \mathbf{w} \\ &= H(H^\dagger Q^{-1}H)^{-1}H^\dagger\mathbf{v} + \mathbf{w} = \mathbf{w}. \end{aligned}$$

So $Q\mathbf{v} = \mathbf{w}$, which is in $NS(H^\dagger)$. This completes the proof.

**A recursive approach:** In array processing and elsewhere it sometimes happens that the matrix $Q$ is estimated from several measurements $\{\mathbf{v}^n,\ n = 1, ..., N\}$ of the noise vector $\mathbf{v}$ as

$$Q = \frac{1}{N}\sum_{n=1}^{N}\mathbf{v}^n(\mathbf{v}^n)^\dagger.$$

Then the inverses of $Q$ and of $H^\dagger Q^{-1}H$ can be obtained recursively, using the *matrix inversion identity*

$$(A + \mathbf{x}\mathbf{x}^\dagger)^{-1} = \frac{1}{1 + \mathbf{x}^\dagger A^{-1}\mathbf{x}}A^{-1}\mathbf{x}\mathbf{x}^\dagger A^{-1}, \tag{42.5}$$

which requires that $\mathbf{x}^\dagger A^{-1}\mathbf{x}$ not equal minus one. Since the matrices involved here are nonnegative definite this denominator will always be at least one. The idea is to define $Q_0 = \epsilon I$, for some $\epsilon > 0$, and, for $n = 1, ..., N$,

$$Q_n = Q_{n-1} + \mathbf{v}^n(\mathbf{v}^n)^\dagger.$$

Then $Q_n^{-1}$ can be obtained from $Q_{n-1}^{-1}$ and $(H^\dagger Q_n^{-1} H)^{-1}$ from $(H^\dagger Q_{n-1}^{-1} H)^{-1}$ using the identity in equation (42.5).

**The vector Wiener filter:** Instead of using the LS estimator as a substitute for the BLUE we can approximate the BLUE using equation (41.2). This approximation of the BLUE is actually an optimal estimator in its own right, called the *vector Wiener filter* (VWF). Assume that $\mathbf{z} = H\mathbf{x} + \mathbf{v}$=$\mathbf{s} + \mathbf{v}$, with $\mathbf{v}$ as above, the signal component $\mathbf{s} = H\mathbf{x}$ and $\mathbf{x}$ a random vector with mean zero and covariance matrix $E(\mathbf{x}\mathbf{x}^\dagger) = \sigma^2 I$. We take our estimate $\hat{\mathbf{s}}$ of the signal $\mathbf{s}$ to be linear in $\mathbf{z}$; that is, $\hat{\mathbf{s}} = B^\dagger \mathbf{z}$ for some matrix $B$. We then find the $B$ for which the expected squared error is minimized; that is, we minimize $E(|\hat{\mathbf{s}} - \mathbf{s}|^2)$. As we shall see when we consider the VWF in more detail in a subsequent chapter, the optimal $B$ is

$$B = \sigma^2(\sigma^2 HH^\dagger + Q)^{-1} HH^\dagger$$

and so the VWF estimate of $\mathbf{x}$ is

$$\mathbf{x}_{VWF} = H^\dagger(HH^\dagger + \sigma^{-2}Q)^{-1}\mathbf{z}.$$

We see from this that the $\epsilon > 0$ in Exercise 8 is the reciprocal of the signal power in the VWF case; the noise power is the sum of the variances of the entries of $\mathbf{v}$, which is the trace of $Q$. The VWF estimate converges to the BLUE estimate as the signal-to-noise ratio approaches infinity.

**Prewhitening:** Using its eigenvalue/eigenvector decomposition $Q = ULU^\dagger$ we find that $Q$ has a Hermitian square root $C = U\sqrt{L}U^\dagger$. Multiplying both sides of $\mathbf{z} = H\mathbf{x} + \mathbf{v}$ by $C^{-1}$ gives

$$\mathbf{y} = G\mathbf{x} + \mathbf{w} \tag{42.6}$$

for $G = C^{-1}H$, $\mathbf{y} = C^{-1}\mathbf{z}$ and $\mathbf{w} = C^{-1}\mathbf{v}$. Then $E(\mathbf{w}\mathbf{w}^\dagger) = I$, so the system in equation (42.6) has a noise component that is white. For this system the BLUE and the LS estimate coincide. Therefore, we can use iterative methods, such as the double ART (DART), to calculate the BLUE.

**Using a norm constraint:** The LS estimator is the one for which the error term $||H\mathbf{x} - \mathbf{z}||^2$ is minimized. If $N = J$ then the LS estimate is an exact solution, which is not necessarily desirable, since we are assuming the presence of a noise term $\mathbf{v}$ in $\mathbf{z}$. Even when $N$ is smaller than $J$ the LS estimate may force $H\mathbf{x}$ to be too close to $\mathbf{z}$. Evidence that this is happening may show up in the norm of $\mathbf{x}_{LS}$ being larger than expected. One way to force the estimation process to take the noise into account is to impose an additional norm constraint, by minimizing

$$||H\mathbf{x} - \mathbf{z}||^2 + \epsilon||\mathbf{x}||^2,$$

for some small $\epsilon > 0$. The $\mathbf{x}$ obtained in this way is

$$\mathbf{x} = (H^\dagger H + \epsilon I)^{-1} H^\dagger \mathbf{z}.$$

If we apply a norm constraint to the prewhitened equation $\mathbf{y} = G\mathbf{x} + \mathbf{w}$ we find that the optimal $\mathbf{x}$ is

$$\mathbf{x} = (H^\dagger Q^{-1} H + \epsilon I)^{-1} H^\dagger Q^{-1} \mathbf{z} = H^\dagger (H H^\dagger + \epsilon Q)^{-1} \mathbf{z},$$

which is the approximation of the BLUE given in equation (41.2).

# Chapter 43

# Kalman Filters

One area in which prediction plays an important role is the tracking of moving targets, such as ballistic missiles, using radar. The range to the target, its angle of elevation and its azimuthal angle are all functions of time governed by linear differential equations. The *state vector* of the system at time $t$ might then be a vector with nine components, the three functions just mentioned, along with their first and second derivatives. In theory, if we knew the initial state perfectly and our differential equations model of the physics was perfect, that would be enough to determine the future states. In practice neither of these is true and we need to assist the differential equation by taking radar measurements of the state at various times. The problem then is to estimate the state at time $t$ using both the measurements taken prior to time $t$ and the estimate based on the physics.

When such tracking is performed digitally the functions of time are replaced by discrete sequences. Let the state vector at time $k\Delta t$ be denoted by $\mathbf{x}_k$, for $k$ an integer and $\Delta t > 0$. Then, with the derivatives in the differential equation approximated by divided differences, the physical model for the evolution of the system in time becomes

$$\mathbf{x}_k = A_{k-1}\mathbf{x}_{k-1} + \mathbf{m}_{k-1}.$$

The matrix $A_{k-1}$, which we assume is known, is obtained from the differential equation, which may have nonconstant coefficients, as well as from the divided difference approximations to the derivatives. The random vector sequence $\mathbf{m}_{k-1}$ represents the error in the physical model due to the discretization and necessary simplification inherent in the original differential equation itself. We assume that the expected value of $\mathbf{m_k}$ is zero for each $k$. The covariance matrix is $E(\mathbf{m}_k\mathbf{m}_k^\dagger) = M_k$.

At time $k\Delta t$ we have the measurements

$$\mathbf{z}_k = H_k\mathbf{x}_k + \mathbf{v}_k,$$

where $H_k$ is a known matrix describing the nature of the linear measurements of the state vector and the random vector $\mathbf{v}_k$ is the noise in these measurements. We assume that the mean value of $\mathbf{v}_k$ is zero for each $k$. The covariance matrix is $E(\mathbf{v}_k\mathbf{v}_k^\dagger) = Q_k$. We assume that the initial state vector $\mathbf{x}_0$ is random and independent of the noise sequences.

Given an unbiased estimate $\hat{\mathbf{x}}_{k-1}$ of the state vector $\mathbf{x}_{k-1}$, our prior estimate of $\mathbf{x}_k$ based solely on the physics is

$$\mathbf{y}_k = A_{k-1}\hat{\mathbf{x}}_{k-1}.$$

**Exercise 1:** Show that $E(\mathbf{y}_k - \mathbf{x}_k) = 0$, so the prior estimate of $\mathbf{x}_k$ is unbiased. We can then write $\mathbf{y}_k = \mathbf{x}_k + \mathbf{w}_k$, with $E(\mathbf{w}_k) = \mathbf{0}$.

**Kalman filtering:** The *Kalman filter* [124], [98], [68] is a recursive algorithm to estimate the state vector $\mathbf{x}_k$ at time $k\Delta t$ as a linear combination of the vectors $\mathbf{z}_k$ and $\mathbf{y}_k$. The estimate $\hat{\mathbf{x}}_k$ will have the form

$$\hat{\mathbf{x}}_k = C_k^\dagger\mathbf{z}_k + D_k^\dagger\mathbf{y}_k, \tag{43.1}$$

for matrices $C_k$ and $D_k$ to be determined. As we shall see, this estimate can also be written as

$$\hat{\mathbf{x}}_k = \mathbf{y}_k + G_k(\mathbf{z}_k - H_k\mathbf{y}_k), \tag{43.2}$$

which shows that the estimate involves a prior prediction step, the $\mathbf{y}_k$, followed by a correction step, in which $H_k\mathbf{y}_k$ is compared to the measured data vector $\mathbf{z}_k$; such estimation methods are sometimes called *predictor-corrector methods*.

In our discussion of the BLUE we saw how to incorporate a prior estimate of the vector to be estimated. The trick was to form a larger matrix equation and then to apply the BLUE to that system. The Kalman filter does just that.

The correction step in the Kalman filter uses the BLUE to solve the combined linear system

$$\mathbf{z}_k = H_k\mathbf{x}_k + \mathbf{v}_k$$

and

$$\mathbf{y}_k = \mathbf{x}_k + \mathbf{w}_k.$$

The covariance matrix of $\hat{\mathbf{x}}_{k-1} - \mathbf{x}_{k-1}$ is denoted $P_{k-1}$ and we let $Q_k = E(\mathbf{w}_k\mathbf{w}_k^\dagger)$. The covariance matrix of $\mathbf{y}_k - \mathbf{x}_k$ is

$$\mathrm{cov}(\mathbf{y}_k - \mathbf{x}_k) = R_k = M_{k-1} + A_{k-1}P_{k-1}A_{k-1}^\dagger.$$

It follows from our earlier discussion of the BLUE that the estimate of $\mathbf{x}_k$ is

$$\hat{\mathbf{x}}_k = \mathbf{y}_k + G_k(\mathbf{z}_k - H\mathbf{y}_k),$$

with

$$G_k = R_k H_k^\dagger (Q_k + H_k R_k H_k^\dagger)^{-1}.$$

Then the covariance matrix of $\hat{\mathbf{x}}_k - \mathbf{x}_k$ is

$$P_k = (I - G_k H_k) R_k.$$

The recursive procedure is to go from $P_{k-1}$ and $M_{k-1}$ to $R_k$, then to $G_k$, from which $\hat{\mathbf{x}}_k$ is formed, and finally to $P_k$, which, along with the known matrix $M_k$, provides the input to the next step. The time-consuming part of this recursive algorithm is the matrix inversion in the calculation of $G_k$. Simpler versions of the algorithm are based on the assumption that the matrices $Q_k$ are diagonal, or on the convergence of the matrices $G_k$ to a limiting matrix $G$ [68].

There are many variants of the Kalman filter, corresponding to variations in the physical model, as well as in the statistical assumptions. The differential equation may be nonlinear, so that the matrices $A_k$ depend on $\mathbf{x}_k$. The system noise sequence $\{\mathbf{w}_k\}$ and the measurement noise sequence $\{\mathbf{v}_k\}$ may be correlated. For computational convenience the various functions that describe the state may be treated separately. The model may include known external inputs to drive the differential system, as in the tracking of spacecraft capable of firing booster rockets. Finally, the noise covariance matrices may not be known *a priori* and adaptive filtering may be needed. We discuss this last issue briefly in the next section.

**Adaptive Kalman filtering:** As in [68] we consider only the case in which the covariance matrix $Q_k$ of the measurement noise $\mathbf{v}_k$ is unknown. As we saw in the discussion of adaptive BLUE, the covariance matrix of the innovations vector $\mathbf{e}_k = \mathbf{z}_k - H_k \mathbf{y}_k$ is

$$S_k = H_k R_k H_k^\dagger + Q_k.$$

Once we have an estimate for $S_k$, we estimate $Q_k$ using

$$\hat{Q}_k = \hat{S}_k - H_k R_k H_k^\dagger.$$

We might assume that $S_k$ is independent of $k$ and estimate $S_k = S$ using past and present innovations; for example, we could use

$$\hat{S} = \frac{1}{k-1} \sum_{j=1}^{k} (\mathbf{z}_j - H_j \mathbf{y}_j)(\mathbf{z}_j - H_j \mathbf{y}_j)^\dagger.$$

# Chapter 44

# The Vector Wiener Filter

The vector Wiener filter (VWF) provides another method for estimating the vector $\mathbf{x}$ given noisy measurements $\mathbf{z}$, where

$$\mathbf{z} = H\mathbf{x} + \mathbf{v},$$

with $\mathbf{x}$ and $\mathbf{v}$ independent random vectors and $H$ a known matrix. We shall assume throughout this chapter that $E(\mathbf{v}) = \mathbf{0}$ and let $Q = E(\mathbf{v}\mathbf{v}^\dagger)$.

It is common to formulate the VWF in the context of filtering a signal vector $\mathbf{s}$ from signal plus noise. The data is the vector

$$\mathbf{z} = \mathbf{s} + \mathbf{v}$$

and we want to estimate $\mathbf{s}$. Each entry of our estimate of the vector $\mathbf{s}$ will be a linear combination of the data values; that is, our estimate is $\hat{\mathbf{s}} = B^\dagger \mathbf{z}$ for some matrix $B$ to be determined. This $B$ will be called the *vector Wiener filter*. To extract the signal from the noise we must know something about possible signals and possible noises. We consider several stages of increasing complexity and correspondence with reality.

Suppose, initially, that all signals must have the form $\mathbf{s} = a\mathbf{u}$, where $a$ is an unknown scalar and $\mathbf{u}$ is a known vector. Suppose that all noises must have the form $\mathbf{v} = b\mathbf{w}$, where $b$ is an unknown scalar and $\mathbf{w}$ is a known vector. Then to estimate $\mathbf{s}$ we must find $a$. So long as $J \geq 2$ we should be able to solve for $a$ and $b$. We form the two equations

$$\mathbf{u}^\dagger \mathbf{z} = a\mathbf{u}^\dagger \mathbf{u} + b\mathbf{u}^\dagger \mathbf{w}$$

and

$$\mathbf{w}^\dagger \mathbf{z} = a\mathbf{w}^\dagger \mathbf{u} + b\mathbf{w}^\dagger \mathbf{w}.$$

This system of two equations in two unknowns will have a unique solution unless $\mathbf{u}$ and $\mathbf{w}$ are proportional, in which case we cannot expect to distinguish signal from noise.

We move now to a somewhat more complicated model. Suppose now that all signals must have the form

$$\mathbf{s} = \sum_{n=1}^{N} a_n \mathbf{u}^n,$$

where the $a_n$ are unknown scalars and the $\mathbf{u}^n$ are known vectors. Suppose that all noises must have the form

$$\mathbf{v} = \sum_{m=1}^{M} b_m \mathbf{w}^m,$$

where the $b_m$ are unknown scalars and $\mathbf{w}^m$ are known vectors. Then to estimate $\mathbf{s}$ we must find the $a_n$. So long as $J \geq N+M$ we should be able to solve for the unique $a_n$ and $b_m$. However, we usually do not know a great deal about the signal and the noise, so we find ourselves in the situation in which the $N$ and $M$ are large. Let $U$ be the $J$ by $N$ matrix whose $n$th column is $\mathbf{u}^n$ and $W$ the $J$ by $M$ matrix whose $m$th column is $\mathbf{w}^m$. Let $V$ be the $J$ by $N+M$ matrix whose first $N$ columns contain $U$ and whose last $M$ columns contain $W$; so $V = [U \quad W]$. Let $\mathbf{c}$ be the $N+M$ by 1 column vector whose first $N$ entries are the $a_n$ and whose last $M$ entries are the $b_m$. We want to solve $\mathbf{z} = V\mathbf{c}$. But this system of linear equations has too many unknowns when $N+M > J$, so we seek the minimum norm solution. In closed form this solution is

$$\hat{\mathbf{c}} = V^\dagger (VV^\dagger)^{-1} \mathbf{z}.$$

The matrix $VV^\dagger = (UU^\dagger + WW^\dagger)$ involves the *signal correlation matrix* $UU^\dagger$ and the *noise correlation matrix* $WW^\dagger$. Consider $UU^\dagger$. The matrix $UU^\dagger$ is $J$ by $J$ and the $(i,j)$ entry of $UU^\dagger$ is given by

$$UU^\dagger_{ij} = \sum_{n=1}^{N} u_i^n u_j^n,$$

so the matrix $\frac{1}{N} UU^\dagger$ has for its entries the average, over all the $n = 1, ..., N$, of the product of the $i$th and $j$th entries of the vectors $\mathbf{u}^n$. Therefore, $\frac{1}{N} UU^\dagger$ is statistical information about the signal; it tells us how these products look, on average, over all members of the family $\{\mathbf{u}^n\}$, the *ensemble*, to use the statistical word.

To pass to a more formal statistical framework, we let the coefficient vectors $\mathbf{a} = (a_1, a_2, ..., a_N)^T$ and $\mathbf{b} = (b_1, b_2, ..., b_M)^T$ be independent random white noise vectors, both with mean zero and covariance matrices $E(\mathbf{a}\mathbf{a}^\dagger) = I$ and $E(\mathbf{b}\mathbf{b}^\dagger) = I$. Then

$$UU^\dagger = E(\mathbf{s}\mathbf{s}^\dagger) = R_s$$

and

$$WW^\dagger = E(\mathbf{v}\mathbf{v}^\dagger) = Q = R_v.$$

The estimate of $\mathbf{s}$ is the result of applying the vector Wiener filter to the vector $\mathbf{z}$ and is given by

$$\hat{\mathbf{s}} = UU^\dagger(UU^\dagger + WW^\dagger)^{-1}\mathbf{z}.$$

**Exercise 1:** Apply the vector Wiener filter to the simplest problem discussed earlier; here let $N = 1$. It will help to use the *matrix inversion identity*

$$(Q + \mathbf{u}\mathbf{u}^\dagger)^{-1} = Q^{-1} - (1 + \mathbf{u}^\dagger Q^{-1}\mathbf{u})^{-1}Q^{-1}\mathbf{u}\mathbf{u}^\dagger Q^{-1}. \tag{44.1}$$

**The VWF and the BLUE:** To apply the VWF to the problem considered in the discussion of the BLUE let the vector $\mathbf{s}$ be $H\mathbf{x}$. We assume, in addition, that the vector $\mathbf{x}$ is a white noise vector; that is, $E(\mathbf{x}\mathbf{x}^\dagger) = \sigma^2 I$. Then $R_s = \sigma^2 HH^\dagger$.

In the VWF approach we estimate $\mathbf{s}$ using

$$\hat{\mathbf{s}} = B^\dagger \mathbf{z},$$

where the matrix $B$ is chosen so as to minimize the mean squared error, $E|\hat{\mathbf{s}} - \mathbf{s}|^2$. This is equivalent to minimizing

$$\operatorname{trace} E((B\mathbf{z} - \mathbf{s})(B\mathbf{z} - \mathbf{s})^\dagger).$$

Expanding the matrix products and using the definitions above, we see that we must minimize

$$\operatorname{trace}(B^\dagger(R_s + R_v)B - R_sB - B^\dagger R_s + R_s).$$

Differentiating with respect to the matrix $B$ using equations (34.1) and (34.2), we find

$$(R_s + R_v)B - R_s = 0,$$

so that

$$B = (R_s + R_v)^{-1}R_s.$$

Our estimate of the signal component is then

$$\hat{\mathbf{s}} = R_s(R_s + R_v)^{-1}\mathbf{z}.$$

With $\mathbf{s} = H\mathbf{x}$, our estimate of $\mathbf{s}$ is

$$\hat{\mathbf{s}} = \sigma^2 HH^\dagger(\sigma^2 HH^\dagger + Q)^{-1}\mathbf{z}$$

and the VWF estimate of $\mathbf{x}$ is

$$\hat{\mathbf{x}} = \sigma^2 H^\dagger(\sigma^2 HH^\dagger + Q)^{-1}\mathbf{z}.$$

How does this estimate relate to the one we got from the BLUE?

The BLUE estimate of $\mathbf{x}$ is

$$\hat{\mathbf{x}} = (H^\dagger Q^{-1} H)^{-1} H^\dagger Q^{-1} \mathbf{z}.$$

From the matrix identity in equation (33.3) we know that

$$(H^\dagger Q^{-1} H + \sigma^{-2} I)^{-1} H^\dagger Q^{-1} = \sigma^2 H^\dagger (\sigma^2 H H^\dagger + Q)^{-1}.$$

Therefore the VWF estimate of $\mathbf{x}$ is

$$\hat{\mathbf{x}} = (H^\dagger Q^{-1} H + \sigma^{-2} I)^{-1} H^\dagger Q^{-1} \mathbf{z}.$$

Note that the BLUE estimate is unbiased and unaffected by changes in the signal strength or the noise strength. In contrast, the VWF is not unbiased and does depend on the signal-to-noise ratio; that is, it depends on the ratio $\sigma^2/\text{trace}\,(Q)$. The BLUE estimate is the limiting case of the VWF estimate, as the signal-to-noise ratio goes to infinity.

The BLUE estimates $\mathbf{s} = H\mathbf{x}$ by first finding the BLUE estimate of $\mathbf{x}$ and then multiplying it by $H$ to get the estimate of the signal $\mathbf{s}$.

**Exercise 2:** Show that the mean squared error in the estimation of $\mathbf{s}$ is

$$E(|\hat{\mathbf{s}} - \mathbf{s}|^2) = \text{trace}\,(H(H^\dagger Q^{-1} H)^{-1} H^\dagger).$$

The VWF finds the linear estimate of $\mathbf{s} = H\mathbf{x}$ that minimizes the mean squared error $E(|\hat{\mathbf{s}} - \mathbf{s}|^2)$. Consequently, the mean squared error in the VWF is less than that in the BLUE.

**Exercise 3:** Assume that $E(\mathbf{x}\mathbf{x}^\dagger) = \sigma^2 I$. Show that the mean squared error for the VWF estimate is

$$E(|\hat{\mathbf{s}} - \mathbf{s}|^2) = \text{trace}\,(H(H^\dagger Q^{-1} H + \sigma^{-2} I)^{-1} H^\dagger).$$

**The functional Wiener filter** The Wiener filter is often presented in the context of random functions of, say, time. In this model signal is $s(t)$ and noise is $q(t)$, where these functions of time are viewed as random functions (stochastic processes). The data is taken to be $z(t)$, a function of $t$, so that the matrices $UU^\dagger$ and $WW^\dagger$ are now *infinite matrices*; the discrete index $j = 1, ..., J$ is now replaced by the continuous index variable $t$. Instead of the finite family $\{\mathbf{u}^n, n = 1..., N\}$, we now have an infinite family of functions $u(t)$ in $\mathcal{U}$. The entries of $UU^\dagger$ are essentially the average values of the products $u(t_1)\overline{u(t_2)}$ over all the members of $\mathcal{U}$. It is often assumed that this average of products is a function not of $t_1$ and $t_2$ separately, but only of their difference $t_1 - t_2$; this is called *stationarity*. So, $aver\{u(t_1)\overline{u(t_2)}\} = r_s(t_1 - t_2)$ comes from a function $r_s(\tau)$ of a

single variable. The Fourier transform of $r_s(\tau)$ is $R_s(\omega)$, the signal power spectrum. The matrix $UU^\dagger$ is then an infinite Toeplitz matrix, constant on each diagonal. The Wiener filtering can actually be achieved by taking Fourier transforms and multiplying and dividing by power spectra, instead of inverting infinite matrices. It is also common to discretize the time variable and to consider the Wiener filter operating on infinite sequences, as we see in the next chapter.

# Chapter 45

# Wiener Filter Approximation

As we saw in the previous chapter, when the data is a finite vector composed of signal plus noise the vector Wiener filter can be used to estimate the signal component, provided we know something about the possible signals and possible noises. In theoretical discussion of filtering signal from signal plus noise it is traditional to assume that both components are doubly infinite sequences of random variables. In this case the Wiener filter is a convolution filter that operates on the input signal plus noise sequence to produce the output estimate of the signal-only sequence. The derivation of the Wiener filter is in terms of the autocorrelation sequences of the two components, as well as their respective power spectra.

Suppose now that the discrete stationary random process to be filtered is the doubly infinite sequence $\{z_n = s_n + q_n\}_{n=-\infty}^{\infty}$, where $\{s_n\}$ is the signal component with autocorrelation function $r_s(k) = E(s_{n+k}\overline{s_n})$ and power spectrum $R_s(\omega)$ defined for $\omega$ in the interval $[-\pi, \pi]$, $\{q_n\}$ is the noise component with autocorrelation function $r_q(k)$ and power spectrum $R_q(\omega)$ defined for $\omega$ in $[-\pi, \pi]$. We assume that for each $n$ the random variables $s_n$ and $q_n$ have mean zero and that the signal and noise are independent of one another. Then the autocorrelation function for the signal plus noise sequence $\{z_n\}$ is

$$r_z(n) = r_s(n) + r_q(n)$$

for all $n$ and

$$R_z(\omega) = R_s(\omega) + R_q(\omega).$$

is the signal plus noise power spectrum.

Let $h = \{h_k\}_{k=-\infty}^{\infty}$ be a linear filter with *transfer function*

$$H(\omega) = \sum_{k=-\infty}^{\infty} h_k e^{ik\omega},$$

for $\omega$ in $[-\pi, \pi]$. Given the sequence $\{z_n\}$ as input to this filter, the output is the sequence

$$y_n = \sum_{k=-\infty}^{\infty} h_k z_{n-k}. \tag{45.1}$$

The goal of Wiener filtering is to select the filter $h$ so that the output sequence $y_n$ approximates the signal $s_n$ sequence as well as possible. Specifically, we seek $h$ so as to minimize the expected squared error, $E(|y_n - s_n|^2)$, which, because of stationarity, is independent of $n$. We have

$$E(|y_n|^2) = \sum_{k=-\infty}^{\infty} h_k \left(\sum_{j=-\infty}^{\infty} \overline{h_j}(r_s(j-k) + r_q(j-k))\right)$$

$$= \sum_{k=-\infty}^{\infty} h_k (r_z * \overline{h})_k$$

which, by the Parseval equation, equals

$$\frac{1}{2\pi} \int H(\omega) R_z(\omega) \overline{H(\omega)} d\omega = \frac{1}{2\pi} \int |H(\omega)|^2 R_z(\omega) d\omega.$$

Similarly,

$$E(s_n \overline{y_n}) = \sum_{j=-\infty}^{\infty} \overline{h_j} r_s(j)$$

which equals

$$\frac{1}{2\pi} \int R_s(\omega) \overline{H(\omega)} d\omega,$$

and

$$E(|s_n|^2) = \frac{1}{2\pi} \int R_s(\omega) d\omega.$$

Therefore,

$$E(|y_n - s_n|^2) = \frac{1}{2\pi} \int |H(\omega)|^2 R_z(\omega) d\omega - \frac{1}{2\pi} \int R_s(\omega) \overline{H(\omega)} d\omega$$

$$-\frac{1}{2\pi} \int R_s(\omega) H(\omega) d\omega + \frac{1}{2\pi} \int R_s(\omega) d\omega.$$

As we shall see shortly, minimizing $E(|y_n - s_n|^2)$ with respect to the function $H(\omega)$ leads to the equation

$$R_z(\omega) H(\omega) = R_s(\omega),$$

so that the transfer function of the optimal filter is

$$H(\omega) = R_s(\omega) / R_z(\omega).$$

The *Wiener filter* is then the sequence $\{h_k\}$ of the Fourier coefficients of this function $H(\omega)$.

To prove that this choice of $H(\omega)$ minimizes $E(|y_n - s_n|^2)$ we note that

$$|H(\omega)|^2 R_z(\omega) - R_s(\omega)\overline{H(\omega)} - R_s(\omega)H(\omega) + R_s(\omega)$$
$$= |H(\omega) - R_s(\omega)/R_z(\omega)|^2 - R_s(\omega) + R_s(\omega)^2/R_z(\omega).$$

Only the first term involves the function $H(\omega)$.

Since $H(\omega)$ is a nonnegative function of $\omega$, therefore real-valued, its Fourier coefficients $h_k$ will be *conjugate symmetric*, that is, $h_{-k} = \overline{h_k}$. This poses a problem when the random process $z_n$ is a discrete time series, with $z_n$ denoting the measurement recorded at time $n$. From the equation (45.1) we see that to produce the output $y_n$ corresponding to time $n$ we need the input for every time, past and future. To remedy this we can obtain the best causal approximation of the Wiener filter $h$.

A filter $g = \{g_k\}_{k=-\infty}^{\infty}$ is said to be *causal* if $g_k = 0$ for $k < 0$; this means that given the input sequence $\{z_n\}$, the output

$$w_n = \sum\nolimits_{k=-\infty}^{\infty} g_k z_{n-k} = \sum\nolimits_{k=0}^{\infty} g_k z_{n-k}$$

requires only values of $z_m$ up to $m = n$. To obtain the causal filter $g$ that best approximates the Wiener filter, we find the coeffcients $g_k$ that minimize the quantity $E(|y_n - w_n|^2)$, or, equivalently,

$$\int_{-\pi}^{\pi} |H(\omega) - \sum\nolimits_{k=0}^{+\infty} g_k e^{ik\omega}|^2 R_z(\omega) d\omega. \tag{45.2}$$

The orthogonality principle tells us that the optimal coefficients must satisfy the equations

$$r_s(m) = \sum\nolimits_{k=0}^{+\infty} g_k r_z(m-k), \tag{45.3}$$

for all $m$. These are the *Wiener-Hopf equations* [152].

Even having a causal filter does not completely solve the problem, since we would have to record and store the infinite past. Instead, we can decide to use a filter $f = \{f_k\}_{k=-\infty}^{\infty}$ for which $f_k = 0$ unless $-K \le k \le L$ for some positive integers $K$ and $L$. This means we must store $L$ values and wait until time $n + K$ to obtain the output for time $n$. Such a linear filter is a *finite memory, finite delay* filter, also called a *finite impulse response* (FIR) filter. Given the input sequence $\{z_n\}$ the output of the FIR filter is

$$v_n = \sum\nolimits_{k=-K}^{L} f_k z_{n-k}.$$

To obtain such an FIR filter $f$ that best approximates the Wiener filter, we find the coefficients $f_k$ that minimize the quantity $E(|y_n - v_n|^2)$, or, equivalently,

$$\int_{-\pi}^{\pi} |H(\omega) - \sum\nolimits_{k=-K}^{L} f_k e^{ik\omega}|^2 R_z(\omega) d\omega. \tag{45.4}$$

The orthogonality principle tells us that the optimal coefficients must satisfy the equations

$$r_s(m) = \sum_{k=-K}^{L} f_k r_z(m-k), \tag{45.5}$$

for $-K \leq m \leq L$.

In [44] it was pointed out that the linear equations that arise in Wiener filter approximation also occur in image reconstruction from projections, with the image to be reconstructed playing the role of the power spectrum to be approximated. The methods of Wiener filter approximation were then used to derive linear and nonlinear image reconstruction procedures.

# Chapter 46

# Adaptive Wiener Filters

Once again, we consider a stationary random process $z_n = s_n + v_n$ with autocorrelation function $E(z_n\overline{z_{n-m}}) = r_z(m) = r_s(m) + r_v(m)$. The finite causal Wiener filter (FCWF) $\mathbf{f} = (f_0, f_1, ..., f_L)^T$ is convolved with $\{z_n\}$ to produce an estimate of $s_n$ given by

$$\hat{s}_n = \sum_{k=0}^{L} f_k z_{n-k}.$$

With $\mathbf{y}_n^\dagger = (z_n, z_{n-1}, ..., z_{n-L})$ we can write $\hat{s}_n = \mathbf{y}_n^\dagger \mathbf{f}$. The FCWF $\mathbf{f}$ minimizes the expected squared error

$$J(\mathbf{f}) = E(|s_n - \hat{s}_n|^2)$$

and is obtained as the solution of the equations

$$r_s(m) = \sum\nolimits_{k=0}^{L} f_k r_z(m-k),$$

for $0 \leq m \leq L$. Therefore, to use the FCWF we need the values $r_s(m)$ and $r_z(m-k)$ for $m$ and $k$ in the set $\{0, 1, ..., L\}$. When these autocorrelation values are not known we can use adaptive methods to approximate the FCWF.

**An adaptive least mean square approach:** We assume now that we have $z_0, z_1, ..., z_N$ and $p_0, p_1, ..., p_N$, where $p_n$ is a prior estimate of $s_n$, but that we do not know the correlation functions $r_z$ and $r_s$.

The gradient of the function $J(\mathbf{f})$ is

$$\nabla J(\mathbf{f}) = R_{zz}\mathbf{f} - \mathbf{r}_s,$$

where $R_{zz}$ is the square matrix with entries $r_z(m-n)$ and $\mathbf{r}_s$ is the vector with entries $r_s(m)$. An iterative gradient descent method for solving the system of equations $R_{zz}\mathbf{f} = \mathbf{r}_s$ is

$$\mathbf{f}_\tau = \mathbf{f}_{\tau-1} - \mu_\tau \nabla J(\mathbf{f}_{\tau-1}),$$

for some step-size parameters $\mu_\tau > 0$.

The adaptive *least mean square* (LMS) approach [55] replaces the gradient of $J(\mathbf{f})$ with an approximation of the gradient of the function $G(\mathbf{f}) = |s_n - \hat{s}_n|^2$, which is $-2(s_n - \hat{s}_n)\mathbf{y}_n$. Since we do not know $s_n$ we replace that term with the estimate $p_n$. The iterative step of the LMS method is

$$\mathbf{f}_\tau = \mathbf{f}_{\tau-1} + \mu_\tau(p_\tau - \mathbf{y}_\tau^\dagger \mathbf{f}_{\tau-1})\mathbf{y}_\tau, \tag{46.1}$$

for $L \leq \tau \leq N$. Notice that it is the approximate gradient of the function $|s_\tau - \hat{s}_\tau|^2$ that is used at this step, in order to involve all the data $z_0, ..., z_N$ as we iterate from $\tau = L$ to $\tau = N$. We illustrate the use of this method in adaptive interference cancellation.

**Adaptive interference cancellation:** Adaptive interference cancellation (AIC) [181] is used to suppress a dominant noise component $v_n$ in the discrete sequence $z_n = s_n + v_n$. It is assumed that we have available a good estimate $q_n$ of $v_n$. The main idea is to switch the roles of signal and noise in the adaptive LMS method and design a filter to estimate $v_n$. Once we have that estimate, we subtract it from $z_n$ to get our estimate of $s_n$.

In the role of $z_n$ we use

$$q_n = v_n + \epsilon_n,$$

where $\epsilon_n$ denotes a low level error component. In the role of $p_n$ we take $z_n$, which is approximately $v_n$, since the signal $s_n$ is much lower than the noise $v_n$. Then $\mathbf{y}_n^\dagger = (q_n, q_{n-1}, ..., q_{n-L})$. The iterative step used to find the filter $\mathbf{f}$ is then

$$\mathbf{f}_\tau = \mathbf{f}_{\tau-1} + \mu_\tau(z_\tau - \mathbf{y}_\tau^\dagger \mathbf{f}_{\tau-1})\mathbf{y}_\tau,$$

for $L \leq \tau \leq N$. When the iterative process has converged to $\mathbf{f}$ we take as our estimate of $s_n$

$$\hat{s}_n = z_n - \sum_{k=0}^{L} f_k q_{n-k}.$$

It has been suggested that this procedure be used in computerized tomography to correct artifacts due to patient motion [85].

**Recursive least squares:** An alternative to the LMS method is to find the least squares solution of the system of $N - L + 1$ linear equations

$$p_n = \sum_{k=0}^{L} f_k z_{n-k},$$

for $L \leq n \leq N$. The *recursive least squares* (RLS) method is a recursive approach to solving this system.

For $L \leq \tau \leq N$ let $Z_\tau$ be the matrix whose rows are $\mathbf{y}_n^\dagger$ for $n = L, ..., \tau$, $\mathbf{p}_\tau^T = (p_L, p_{L+1}, ..., p_\tau)$ and $Q_\tau = Z_\tau^\dagger Z_\tau$. The least squares solution we seek is

$$\mathbf{f} = Q_N^{-1} Z_N^\dagger \mathbf{p}_N.$$

**Exercise 1:** Show that $Q_\tau = Q_{\tau-1} + \mathbf{y}_\tau \mathbf{y}_\tau^\dagger$, for $L < \tau \leq N$ .

**Exercise 2:** Use the matrix inversion identity in equation (44.1) to write $Q_\tau^{-1}$ in terms of $Q_{\tau-1}^{-1}$.

**Exercise 3:** Using the previous exercise, show that the desired least squares solution $\mathbf{f}$ is $\mathbf{f} = \mathbf{f}_N$, where, for $L \leq \tau \leq N$ we let

$$\mathbf{f}_\tau = \mathbf{f}_{\tau-1} + (\frac{p_\tau - \mathbf{y}_\tau^\dagger \mathbf{f}_{\tau-1}}{1 + \mathbf{y}_\tau^\dagger Q_{\tau-1}^{-1} \mathbf{y}_\tau}) Q_{\tau-1}^{-1} \mathbf{y}_\tau.$$

Comparing this iterative step with that given by equation (46.1) we see that the former gives an explicit value for $\mu_\tau$ and uses $Q_{\tau-1}^{-1}\mathbf{y}_\tau$ instead of $\mathbf{y}_\tau$ as the direction vector for the iterative step. The RMS iteration produces a more accurate estimate of the FCWF than does the LMS method, but requires more computation.

# Chapter 47

# Classical and Modern Methods

In [55] Candy locates the beginning of the classical period of spectral estimation in Schuster's use of Fourier techniques in 1898 to analyze sun spot data [164]. The role of Fourier techniques grew with the discovery, by Wiener in the USA and Khintchine in the USSR, of the relation between the power spectrum and the autocorrelation function. Much of Wiener's important work on control and communication remained classified and became known only with the publication of his classic text *Time Series* in 1949 [182]. The book by Blackman and Tukey, *Measurement of Power Spectra* [15], provides perhaps the best description of the classical methods. With the discovery of the FFT by Cooley and Tukey in 1965, all the pieces were in place for the rapid development of this DFT-based approach to spectral estimation.

Until about the middle of the 1970's most signal processing depended almost exclusively on the DFT, as implemented using the FFT. Algorithms such as the Gerchberg-Papoulis bandlimited extrapolation method were performed as iterative operations on finite vectors, using the FFT at every step. Linear filters and related windowing methods involving the FFT were also used to enhance the resolution of the reconstructed objects. The proper design of these filters was an area of interest to quite a number of researchers, John Tukey among them. Then around the end of that decade interest in entropy maximization began to grow, as researchers began to wonder if high-resolution methods developed for seismic oil exploration could be applied successfully in other areas.

John Burg had developed his MEM while working in the oil industry in the 1960's. He then went to Stanford as a mature graduate student and received his doctorate in 1975 for a thesis based largely on his earlier

work on MEM [27]. This thesis and a handful of earlier presentations at meetings [25], [26] fueled the interest in entropy.

It was not only the effectiveness of Burg's techniques that attracted attention. The classical methods seemed to some to be *ad hoc* and they sought a more intellectually satisfying basis for spectral estimation. Classical methods start with the time series data, say $x_n$, for $n = 1, ..., N$. In the direct approach, slightly simplified, the data is *windowed*, that is, $x_n$ is replaced with $x_n w_n$ for some choice of constants $w_n$. Then the DFT is computed, using the FFT, and the magnitude squared of the DFT is the desired estimate of the power spectrum. In the more indirect approach, autocorrelation values $r_x(m)$ are first estimated, for $m = 0, 1, ..., M$, where $M$ is some fraction of the data length $N$. Then these estimates of $r_x(m)$ are windowed and the DFT calculated, again using the FFT.

What some people objected to was the use of these windows. After all, the measured data was $x_n$, not $x_n w_n$, so why corrupt the data at the first step? The classical methods produced answers that depended to some extent on which window function one used; there had to be a better way. Entropy maximization was the answer to their prayers.

In 1981 the first of several international workshops on entropy maximization was held at the University of Wyoming, bring together most of the people working in this area. The books [168] and [169] contain the papers presented at those workshops. As one can see from reading those papers, the general theme is that a new day has dawned.

It was soon recognized that maximum entropy methods were closely related to model-based techniques that had been part of statistical time series for decades. This realization led to a broader use of *autoregressive* (AR) and *autoregressive, moving average* (ARMA) models for spectral estimation [158], as well as of eigenvector methods, such as Pisarenko's method [156]. What Candy describes as the modern approach to spectral estimation is one based on explicit parametric models, in contrast to the classical non-parametric approach. The book edited by Don Childers [65] is a collection of journal articles that captures the state-of-the-art at the end of the 1970's.

In a sense the transition from the classical ways to the modern methods solved little; the choice of models is as *ad hoc* as the choice of windows was before. On the other hand, we do have a wider collection of techniques from which to choose and we can examine these techniques to see when they perform well and when they do not. We do not expect one approach to work in all cases. High-speed computation permits the use of more complicated parametric models tailored to the physics of a given situation.

At the end of the day our estimates are going to be used for some purpose. In medical imaging a doctor is going to make a diagnosis based in part on what the image reveals. How good the image needs to be depends on the purpose for which it is made. Judging the quality of a reconstructed

image based on somewhat subjective criteria such as how useful it is to a doctor is a problem that is not yet solved. Human observer studies are one way to obtain this non-mathematical evaluation of reconstruction and estimation methods. The next step beyond that is to develop computer software that judges the images or spectra as a human would.

# Chapter 48

# Entropy Maximization

The problem of estimating the nonnegative function $R(\omega)$, for $|\omega| \leq \pi$, from the finitely many Fourier transform values

$$r(n) = \int_{-\pi}^{\pi} R(\omega) \exp(-in\omega) d\omega/2\pi,\ n = -N, ..., N$$

is an *underdetermined problem*, meaning that the data alone is insufficient to determine a unique answer. In such situations we must select one solution out of the infinitely many that are mathematically possible. The obvious questions we need to answer are: What criteria do we use in this selection? How do we find algorithms that meet our chosen criteria? In this chapter we look at some of the answers people have offered and at one particular algorithm, Burg's *maximum entropy* method (MEM) [25], [26].

These values $r(n)$ are autocorrelation function values associated with a random process having $R(\omega)$ for its power spectrum. In many applications, such as seismic remote sensing, these autocorrelation values are estimates obtained from relatively few samples of the underlying random process, so that $N$ is not large. The DFT estimate,

$$R_{DFT}(\omega) = \sum_{n=-N}^{N} r(n) \exp(in\omega),$$

is real-valued and consistent with the data, but is not necessarily nonnegative. For small values of $N$ the DFT may not be sufficiently resolving to be useful. This suggests that one criterion we can use to perform our selection process is to require that the method provide better resolution than the DFT for relatively small values of $N$, when reconstructing power spectra that consist mainly of delta functions.

## A brief side trip to philosophy:

Generally speaking, we would expect to do a better job of estimating a function from data pertaining to that function if we also possess additional prior information about the function to be estimated and are able to employ estimation techniques that make use of that additional information. There is the danger, however, that we may end up with an answer that is influenced more by our prior guesses than by the actual measured data. Striking a balance between including prior knowledge and letting the data speak for itself is a noble goal; how to achieve that is the question. At this stage, we begin to suspect that the problem is as much philosophical as it is mathematical.

We are essentially looking for principles of induction that enable us to extrapolate from what we have measured to what we have not. Unwilling to turn the problem over entirely to the philosophers, a number of mathematicians and physicists have sought mathematical solutions to this inference problem, framed in terms of what the *most likely* answer is, or which answer involves the smallest amount of additional prior information [78]. This is not, of course, a new issue; it has been argued for centuries with regard to the use of what we now call Bayesian statistics; *objective* Bayesians allow the use of prior information, but only if it is the right prior information. The interested reader should consult the books [168] and [169], containing papers by Ed Jaynes, Roy Frieden and others originally presented at workshops on this topic held in the early 1980's.

The maximum entropy method is a general approach to such problems that includes Burg's algorithm as a particular case. It is argued that by maximizing entropy we are, in some sense, being maximally noncommittal about what we do not know and thereby introducing a minimum of prior knowledge (some would say prior guesswork) into the solution. In the case of Burg's MEM a somewhat more mathematical argument is available.

Let $\{x_n\}_{n=-\infty}^{\infty}$ be a stationary random process with autocorrelation sequence $r(m)$ and power spectrum $R(\omega)$, $|\omega| \leq \pi$. The prediction problem is the following: suppose we have measured the values of the process prior to time $n$ and we want to predict the value of the process at time $n$. On average, how much error do we expect to make in predicting $x_n$ from knowledge of the infinite past? The answer, according to Szegö's theorem [114], is

$$\exp[\int_{-\pi}^{\pi} \log R(\omega) d\omega];$$

the integral

$$\int_{-\pi}^{\pi} \log R(\omega) d\omega$$

is the *Burg entropy* of the random process [158]. Processes that are very predictable have low entropy, while those that are quite unpredictable, or,

like white noise, completely unpredictable, have high entropy; to make entropies comparable we assume a fixed value of $r(0)$. Given the data $r(n)$, $|n| \leq N$, Burg's method selects that power spectrum consistent with these autocorrelation values that corresponds to the most unpredictable random process.

Other similar procedures are also based on selection through optimization. We have seen the minimum norm approach to finding a solution to an underdetermined system of linear equations, the minimum expected squared error approach in statistical filtering and later we shall see the maximum likelihood method used in detection. We must keep in mind that, however comforting it may be to know that we are on solid philosophical ground (if such exists) in choosing our selection criteria, if the method does not work well, we must use something else. As we shall see, the MEM, like every other reasonable method, works well sometimes and not so well other times. There is certainly philosophical precedent for considering the consequences of our choices, as Blaise Pascal's famous wager about the existence of God nicely illustrates. As an attentive reader of the books [168] and [169] will surely note, there is a certain theological tone to some of the arguments offered in support of entropy maximization. One group of authors (reference omitted) went so far as to declare that entropy maximization was what one did if one cared what happened to one's data.

The objective of Burg's MEM for estimating a power spectrum is to seek better resolution by combining nonnegativity and data-consistency in a single closed-form estimate. The MEM is remarkable in that it is the only closed-form (that is, noniterative) estimation method that is guaranteed to produce an estimate that is both nonnegative and consistent with the autocorrelation samples. Later we shall consider a more general method, the inverse PDFT (IPDFT), that is both data-consistent and positive in most cases.

## Properties of the sequence $\{r(n)\}$:

We begin our discussion with a look at important properties of the sequence $\{r(n)\}$. Because $R(\omega) \geq 0$, the values $r(n)$ are often called *autocorrelation values*.

Since $R(\omega) \geq 0$, it follows immediately that $r(0) \geq 0$. In addition, $r(0) \geq |r(n)|$ for all $n$:

$$|r(n)| = |\int_{-\pi}^{\pi} R(\omega) \exp(-in\omega) d\omega/2\pi|$$

$$\leq \int_{-\pi}^{\pi} R(\omega) |\exp(-in\omega)| d\omega/2\pi = r(0).$$

In fact, if $r(0) = |r(n)| > 0$ for some $n > 0$, then $R$ is a sum of at most $n + 1$ delta functions with nonnegative amplitudes. To see this, suppose

that $r(n) = |r(n)| \exp(i\theta) = r(0) \exp(i\theta)$. Then

$$\int_{-\pi}^{\pi} R(\omega)|1 - \exp(i(\theta + n\omega))|^2 d\omega/2\pi$$

$$= \int_{-\pi}^{\pi} R(\omega)(1 - \exp(i(\theta + n\omega))(1 - \exp(-i(\theta + n\omega))d\omega/2\pi$$

$$= \int_{-\pi}^{\pi} R(\omega)[2 - \exp(i(\theta + n\omega)) - \exp(-i(\theta + n\omega))]d\omega/2\pi$$

$$= 2r(0) - \exp(i\theta)\overline{r(n)} - \exp(-i\theta)r(n) = 2r(0) - r(0) - r(0) = 0.$$

Therefore, $R(\omega) > 0$ only at the values of $\omega$ where $|1-\exp(i(\theta+n\omega))|^2 = 0$; that is, only at $\omega = n^{-1}(2\pi k - \theta)$ for some integer $k$. Since $|\omega| \leq \pi$ there are only finitely many such $k$.

This result is important in any discussion of resolution limits. It is natural to feel that if we have only the Fourier coefficients $r(n)$ for $|n| \leq N$ then we have only the low frequency information about the function $R(\omega)$. How is it possible to achieve higher resolution? Notice, however, that in the case just considered, the infinite sequence of Fourier coefficients is periodic. Of course, we do not know this *a priori*, necessarily. The fact that $|r(N)| = r(0)$ does not, *by itself*, tell us that $R(\omega)$ consists solely of delta functions and that the sequence of Fourier coefficients is periodic. But, under the added assumption that $R(\omega) \geq 0$, it does! When we put in this prior information about $R(\omega)$ we find that the data now tells us more than it did before. This is a good example of the point made in the Introduction: To get information out we need to put information in.

In discussing the Burg MEM estimate we shall need to refer to the concept of *minimum phase* vectors. We consider that briefly now.

## Minimum phase vectors:

We say that the finite column vector with complex entries $(a_0, a_1, ..., a_N)^T$ is a *minimum phase* vector if the complex polynomial

$$A(z) = a_0 + a_1 z + ... + a_N z^N$$

has the property that $A(z) = 0$ implies that $|z| > 1$; that is, all roots of $A(z)$ are outside the unit circle. Consequently, the function $B(z)$ given by $B(z) = 1/A(z)$ is analytic in a disk centered at the origin and including the unit circle. Therefore, we can write

$$B(z) = b_0 + b_1 z + b_2 z^2 + ...$$

and taking $z = \exp(i\omega)$, we get

$$B(\exp(i\omega)) = b_0 + b_1 \exp(i\omega) + b_2 \exp(2i\omega) + ....$$

The point here is that $B(\exp(i\omega))$ is a one-sided trigonometric series, with only terms corresponding to $\exp(in\omega)$ for nonnegative $n$.

## Burg's MEM:

The approach is to estimate $R(\omega)$ by the function $S(\omega) > 0$ that maximizes the so-called Burg entropy, $\int_{-\pi}^{\pi} \log S(\theta) d\theta$, subject to the data constraints.

The Euler-Lagrange equation from the calculus of variations allows us to conclude that $S(\omega)$ has the form

$$S(\omega) = 1/H(\omega)$$

for

$$H(\omega) = \sum_{n=-N}^{N} h_n e^{in\omega} > 0.$$

From the Fejér-Riesz theorem 14.1 we know that $H(\omega) = |A(e^{i\omega})|^2$ for minimum phase $A(z)$ as above. As we now show, the coefficients $a_n$ satisfy a system of linear equations formed using the data $r(n)$.

Given the data $r(n), |n| \leq N$, we form the *autocorrelation matrix* $R$ with entries $R_{mn} = r(m-n)$, for $-N \leq m, n \leq N$. Let $\delta$ be the column vector $\delta = (1, 0, ..., 0)^T$. Let $\mathbf{a} = (a_0, a_1, ..., a_N)^T$ be the solution of the system $R\mathbf{a} = \delta$. Then Burg's MEM estimate is the function $S(\omega) = R_{MEM}(\omega)$ given by

$$R_{MEM}(\omega) = a_0/|A(\exp(i\omega))|^2, |\omega| \leq \pi.$$

Once we show that $a_0 \geq 0$ then it will be obvious that $R_{MEM}(\omega) \geq 0$. We also must show that $R_{MEM}$ is data-consistent; that is,

$$r(n) = \int_{-\pi}^{\pi} R_{MEM}(\omega) \exp(-in\omega) d\omega/2\pi =, \; n = -N, ..., N.$$

Let us write $R_{MEM}(\omega)$ as a Fourier series; that is

$$R_{MEM}(\omega) = \sum_{n=-\infty}^{+\infty} q(n) \exp(in\omega), \; |\omega| \leq \pi.$$

From the form of $R_{MEM}(\omega)$ we have

$$R_{MEM}(\omega)\overline{A(\exp(i\omega))} = a_0 B(\exp(i\omega)).$$

Suppose, as we shall shortly show, that $A(z)$ has all its roots outside the unit circle and so $B(\exp(i\omega))$ is a one-sided trigonometric series, with only terms corresponding to $\exp(in\omega)$ for nonnegative $n$. Then, multiplying on the left side of the equation above and equating coefficients corresponding to $n = 0, -1, -2, ...$, we find that, provided $q(n) = r(n), \text{ for } |n| \leq N$, we

must have $R\mathbf{a} = \delta$. Notice that these are precisely the same equations we solve in calculating the coefficients of an AR process. For that reason the MEM is sometimes called an autoregressive method for spectral estimation.

We now show that if $R\mathbf{a} = \delta$ then $A(z)$ has all its roots outside the unit circle. Let $r\exp(i\theta)$ be a root of $A(z)$. Then write

$$A(z) = (z - r\exp(i\theta))C(z),$$

where

$$C(z) = c_0 + c_1 z + c_2 z^2 + ... + c_{N-1} z^{N-1}.$$

Then the vector $\mathbf{a} = (a_0, a_1, ..., a_N)^T$ can be written as $\mathbf{a} = -r\exp(i\theta)\mathbf{c} + \mathbf{d}$, where $\mathbf{c} = (c_0, c_1, ..., c_{N-1}, 0)^T$ and $\mathbf{d} = (0, c_0, c_1, ..., c_{N-1})^T$. So $\delta = R\mathbf{a} = -r\exp(i\theta)R\mathbf{c} + R\mathbf{d}$ and

$$0 = \mathbf{d}^\dagger \delta = -r\exp(i\theta)\mathbf{d}^\dagger R\mathbf{c} + \mathbf{d}^\dagger R\mathbf{d},$$

so that

$$r\exp(i\theta)\mathbf{d}^\dagger R\mathbf{c} = \mathbf{d}^\dagger R\mathbf{d}.$$

From the Cauchy inequality we know that

$$|\mathbf{d}^\dagger R\mathbf{c}|^2 \leq (\mathbf{d}^\dagger R\mathbf{d})(\mathbf{c}^\dagger R\mathbf{c}) = (\mathbf{d}^\dagger R\mathbf{d})^2, \tag{48.1}$$

where the last equality comes from the special form of the matrix $R$ and the similarity between $\mathbf{c}$ and $\mathbf{d}$.

With

$$D(\omega) = c_0 e^{i\omega} + c_1 e^{2i\omega} ... + c_{N-1} e^{iN\omega}$$

and

$$C(\omega) = c_0 + c_1 e^{i\omega} + ... + c_{N-1} e^{i(N-1)\omega},$$

we can easily show that

$$\mathbf{d}^\dagger R\mathbf{d} = \mathbf{c}^\dagger R\mathbf{c} = \frac{1}{2\pi}\int_{-\pi}^{\pi} R(\omega)|D(\omega)|^2 d\omega$$

and

$$\mathbf{d}^\dagger R\mathbf{c} = \frac{1}{2\pi}\int_{-\pi}^{\pi} R(\omega)\overline{D(\omega)}C(\omega)d\omega.$$

If there is equality in the Cauchy inequality (48.1) then $r = 1$ and we would have

$$\exp(i\theta)\frac{1}{2\pi}\int_{-\pi}^{\pi} R(\omega)\overline{D(\omega)}C(\omega)d\omega = \frac{1}{2\pi}\int_{-\pi}^{\pi} R(\omega)|D(\omega)|^2 d\omega.$$

From the Cauchy inequality for integrals, we can conclude that

$$\exp(i\theta)\overline{D(\omega)}C(\omega) = |D(\omega)|^2$$

for all $\omega$ for which $R(\omega) > 0$. But

$$\exp(i\omega)C(\omega) = D(\omega).$$

Therefore we cannot have $r = 1$ unless $R(\omega) = \delta(\omega - \theta)$. In all other cases we have

$$|\mathbf{d}^\dagger R\mathbf{c}|^2 < |r|^2|\mathbf{d}^\dagger R\mathbf{c}|^2,$$

from which we conclude that $|r| > 1$.

**Solving $R\mathbf{a} = \delta$ using Levinson's algorithm:** Because the matrix $R$ is Toeplitz (constant on diagonals) and positive definite, there is a fast algorithm for solving $R\mathbf{a} = \delta$ for $\mathbf{a}$. Instead of a single $R$ we let $R_M$ be the matrix defined for $M = 0, 1, ..., N$ by

$$R_M = \begin{bmatrix} r(0) & r(-1) & ... & r(-M) \\ r(1) & r(0) & ... & r(-M+1) \\ . & & & \\ . & & & \\ . & & & \\ r(M) & r(M-1) & ... & r(0) \end{bmatrix}$$

so that $R = R_N$. We also let $\delta^M$ be the $M+1$-dimensional column vector $\delta^M = (1, 0, ..., 0)^T$. We want to find the column vector $\mathbf{a}^M = (a_0^M, a_1^M, ..., a_M^M)^T$ that satisfies the equation $R_M\mathbf{a}^M = \delta^M$. The point of Levinson's algorithm is to calculate $\mathbf{a}^{M+1}$ quickly from $\mathbf{a}^M$.

For fixed $M$ find constants $\alpha$ and $\beta$ so that

$$\delta^M = R_M\left\{\alpha\begin{bmatrix} a_0^{M-1} \\ a_1^{M-1} \\ . \\ . \\ . \\ a_{M-1}^{M-1} \\ 0 \end{bmatrix} + \beta\begin{bmatrix} 0 \\ \overline{a}_{M-1}^{M-1} \\ \overline{a}_{M-2}^{M-1} \\ . \\ . \\ . \\ \overline{a}_0^{M-1} \end{bmatrix}\right\}$$

$$= \left\{\alpha\begin{bmatrix} 1 \\ 0 \\ . \\ . \\ . \\ 0 \\ \gamma^M \end{bmatrix} + \beta\begin{bmatrix} \overline{\gamma}^M \\ 0 \\ . \\ . \\ . \\ 0 \\ 1 \end{bmatrix}\right\},$$

where

$$\gamma^M = r(M)a_0^{M-1} + r(M-1)a_1^{M-1} + ... + r(1)a_{M-1}^{M-1}.$$

We then have

$$\alpha + \beta\overline{\gamma^M} = 1,\ \alpha\gamma^M + \beta = 0$$

or

$$\beta = -\alpha\gamma^M,\ \alpha - \alpha|\gamma^M|^2 = 1,$$

so

$$\alpha = 1/(1 - |\gamma^M|^2),\ \beta = -\gamma^M/(1 - |\gamma^M|^2).$$

Therefore, the algorithm begins with $M = 0$, $R_0 = [r(0)], a_0^0 = r(0)^{-1}$. At each step calculate the $\gamma^M$, solve for $\alpha$ and $\beta$ and form the next $\mathbf{a}^M$.

The MEM resolves better than the DFT when the true power spectrum being reconstructed is a sum of delta functions plus a flat background. When the background itself is not flat performance of the MEM degrades rapidly; the MEM tends to interpret any non-flat background in terms of additional delta functions. In the next chapter we consider an extension of the MEM, called the indirect PDFT (IPDFT), that corrects this flaw.

Why Burg's MEM and the IPDFT are able to resolve closely spaced sinusoidal components better than the DFT is best answered by studying the eigenvalues and eigenvectors of the matrix $R$; we turn to this topic in a later chapter.

## A sufficient condition for positive-definiteness:

If the function

$$R(\omega) = \sum\nolimits_{n=-\infty}^{\infty} r(n)e^{in\omega}$$

is nonnegative on the interval $[-\pi, \pi]$ then the matrices $R_M$ above are nonnegative-definite for every $M$. Theorems by Herglotz and by Bochner go in the reverse direction [4]. Katznelson [125] gives the following result.

**Theorem 48.1** *Let $\{f(n)\}_{n=-\infty}^{\infty}$ be a sequence of nonnegative real numbers converging to zero, with $f(-n) = f(n)$ for each $n$. If, for each $n > 0$, we have*

$$(f(n-1) - f(n)) - (f(n) - f(n+1)) > 0,$$

*then there is a nonnegative function $R(\omega)$ on the interval $[-\pi, \pi]$ with $f(n) = r(n)$ for each $n$.*

The figures below illustrate the behavior of the MEM. In Figures 48.1, 48.2 and 48.3 the true object has two delta functions at $0.95\pi$ and $1.05\pi$. The data is $f(n)$ for $|n| \leq 10$. The DFT cannot resolve the two spikes. The SNR is high in Figure 48.1 and the MEM easily resolves them. In Figure 48.2 the SNR is much lower and MEM no longer resolves the spikes.

**Exercise 1:** In Figure 48.3 the SNR is much higher than in Figure 48.1. Explain why the graph looks as it does.

In Figure 48.4 the true object is a box supported between $0.75\pi$ and $1.25\pi$. Here $N = 10$ again. The MEM does a poor job reconstructing the box. This weakness in MEM will become a problem in the last two figures, in which the true object consists of the box with the two spikes added. In Figure 48.5 we have $N = 10$, while in Figure 48.6 $N = 25$.

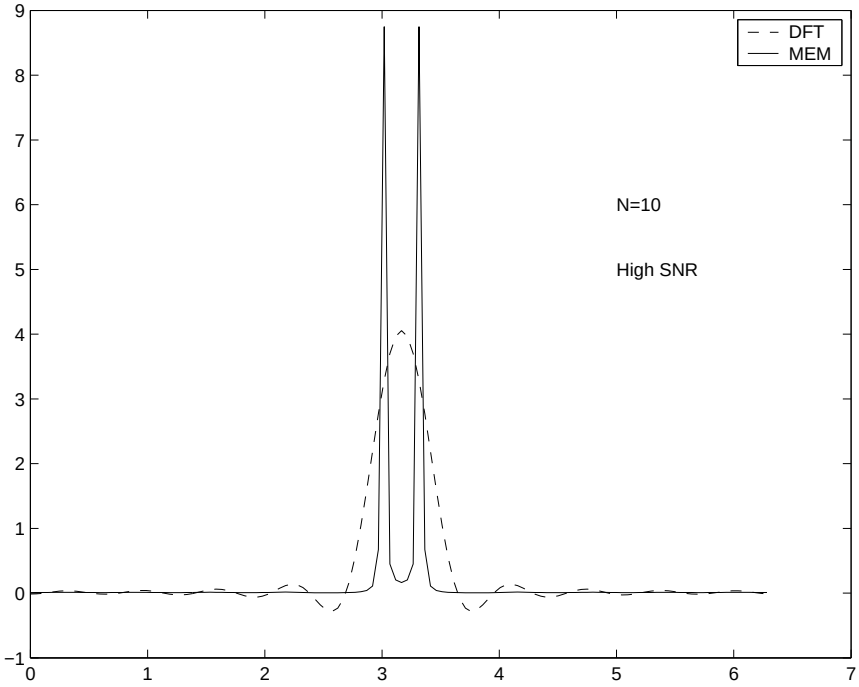



Figure 48.1: The DFT and MEM, $N = 10$, high SNR

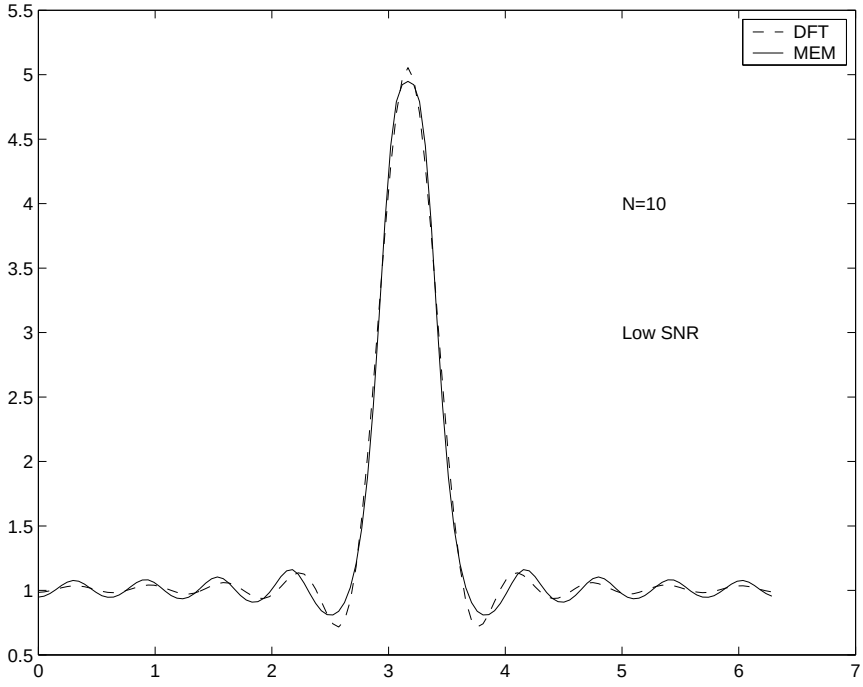



Figure 48.2: The DFT and MEM, $N = 10$, low SNR

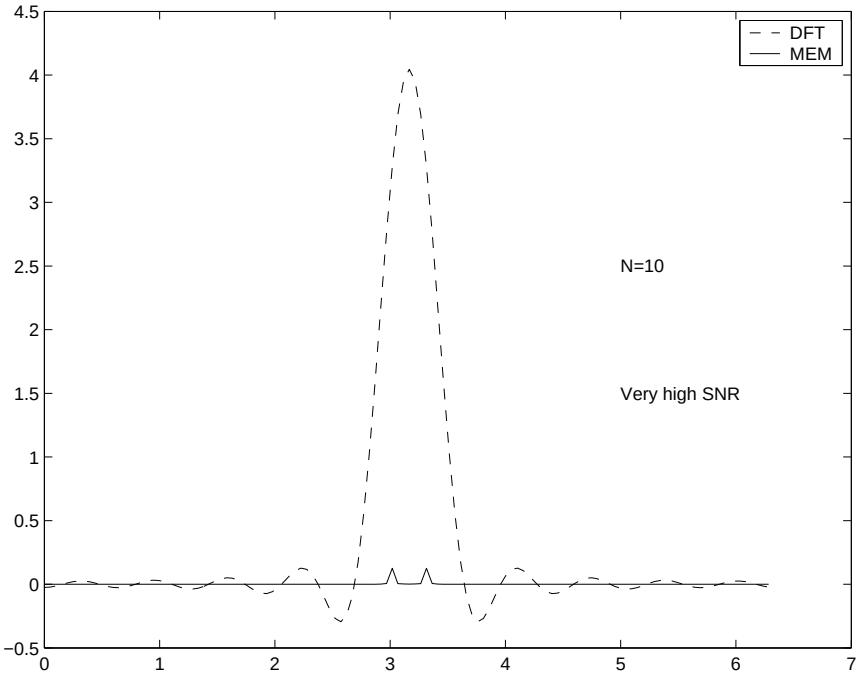



Figure 48.3: The DFT and MEM, $N = 10$, very high SNR. What happened?

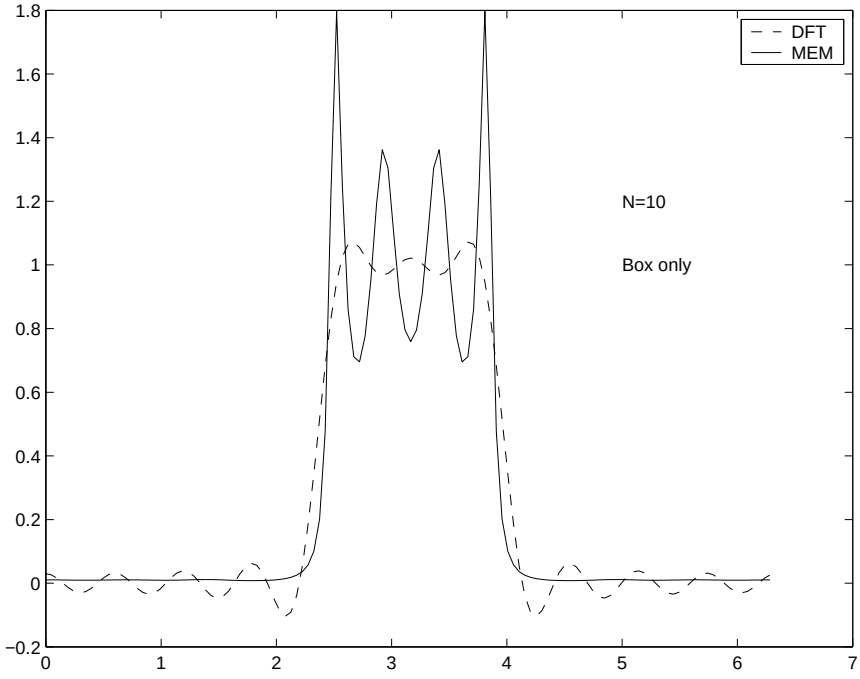



Figure 48.4: MEM and DFT for a box object; $N = 10$

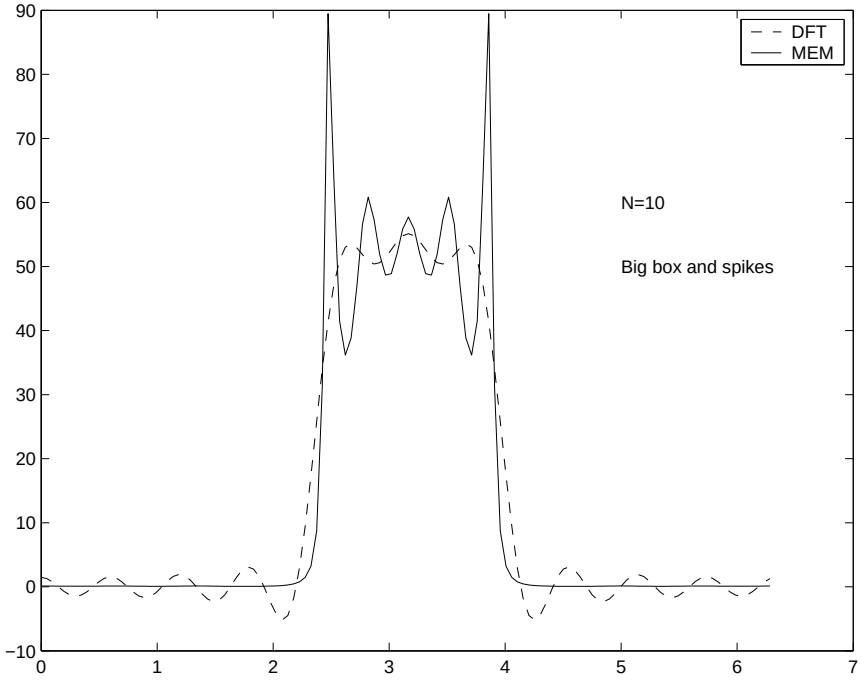



Figure 48.5: The DFT and MEM: two spikes on a large box; $N = 10$

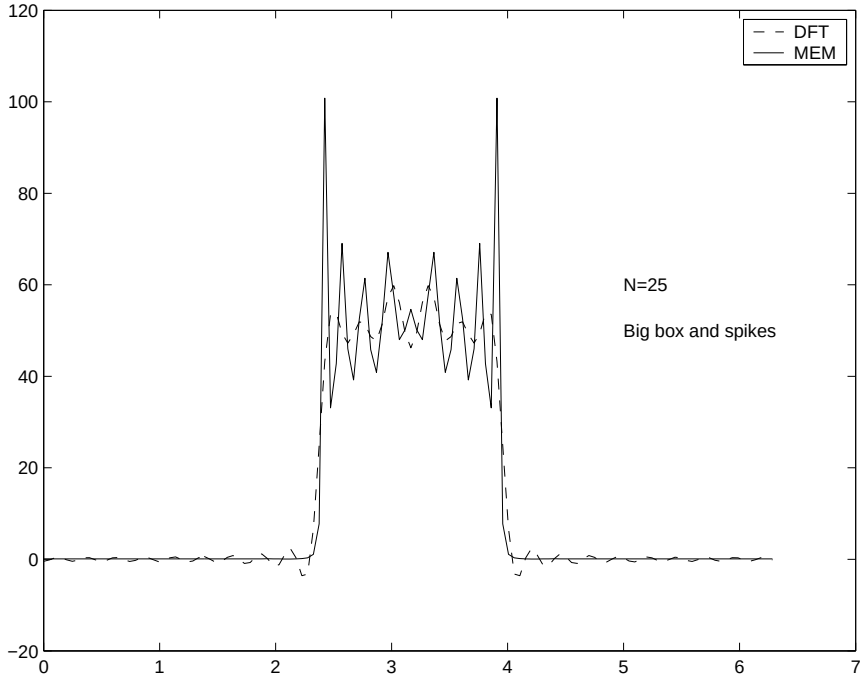



Figure 48.6: The DFT and MEM: two spikes on a large box; $N = 25$

# Chapter 49

# The IPDFT

Experience with Burg's MEM shows that it is capable of resolving closely spaced delta functions better than the DFT, provided that the background is flat. When the background is not flat MEM tends to interpret the non-flat background as additional delta functions to be resolved. In this chapter we consider an extension of MEM based on the PDFT that can resolve in the presence of non-flat background. This method is called the *indirect PDFT* (IPDFT) [48]. The IPDFT applies to the reconstruction of one-dimensional power spectra, but the main idea can be used to generate high resolution methods for multi-dimensional spectra as well. The IPDFT method is suggested by considering the MEM equations $R\mathbf{a} = \delta$ as a particular case of the equations that arise in Wiener filter approximation. As in the previous chapter, we assume that we have the autocorrelation values $r(n)$ for $|n| \leq N$, from which we wish to estimate the power spectrum

$$R(\omega) = \sum_{n=-\infty}^{+\infty} r(n)e^{in\omega},\ |\omega| \leq \pi.$$

In the chapter on Wiener filter approximation we saw that the best finite length filter approximation of the Wiener filter is obtained by minimizing the integral in equation (45.4)

$$\int_{-\pi}^{\pi} |H(\omega) - \sum_{k=-K}^{L} f_k e^{ik\omega}|^2 (R_s(\omega) + R_u(\omega))d\omega.$$

The optimal coefficients then must satisfy equations (45.5):

$$r_s(m) = \sum_{k=-K}^{L} f_k(r_s(m-k) + r_u(m-k)), \tag{49.1}$$

for $-K \leq m \leq L$.

Consider the case in which the power spectrum we wish to estimate consists of a signal component that is the sum of delta functions and a noise

component that is white noise. If we construct a finite length Wiener filter that filters out the signal component and leaves only the noise, then that filter should be able to zero out the delta function components. By finding the locations of those zeros we can find the supports of the delta functions. So the approach is to reverse the roles of signal and noise, viewing the signal as the component called $u$ and the noise as the component called $s$ in the discussion of the Wiener filter. The autocorrelation function $r_s(n)$ corresponds to the white noise now and so $r_s(n) = 0$ for $n \neq 0$. The terms $r_s(n) + r_u(n)$ are the data values $r(n)$, for $|n| \leq N$. Taking $K = 0$ and $L = N$ in equation (49.1), we obtain

$$\sum_{k=0}^{N} f_k r(m-k) = 0,$$

for $m = 1, 2, ..., N$ and

$$\sum_{k=0}^{N} f_k r(0-k) = r(0),$$

which is precisely that same system $R\mathbf{a} = \delta$ that occurs in MEM.

This approach reveals that the vector $\mathbf{a} = (a_0, ..., a_N)^T$ we find in MEM can be viewed an a finite length approximation of the Wiener filter designed to remove the delta function component and to leave the remaining flat white noise component untouched. The polynomial

$$A(\omega) = \sum_{n=0}^{N} a_n e^{in\omega}$$

will then have zeros near the supports of the delta functions. What happens to MEM when the background is not flat is that the filter tries to eliminate any component that is not white noise, so places the zeros of $A(\omega)$ in the wrong places.

Suppose we take $P(\omega) \geq 0$ to be our estimate of the background component of $R(\omega)$; that is, we believe that $R(\omega)$ equals a multiple of $P(\omega)$ plus a sum of delta functions. We now ask for the finite length approximation of the Wiener filter that removes the delta functions and leaves any background component that looks like $P(\omega)$ untouched. We then take $r_s(n) = p(n)$, where

$$P(\omega) = \sum_{n=-\infty}^{+\infty} p(n) e^{in\omega},\ |\omega| \leq \pi.$$

The desired filter is $\mathbf{f} = (f_0, ..., f_N)^T$ satisfying the equations

$$p(m) = \sum_{k=0}^{N} f_k r(m-k). \tag{49.2}$$

Once we have found $\mathbf{f}$ we form the polynomial

$$F(\omega) = \sum_{k=0}^{N} f_k e^{ik\omega},\ |\omega| \leq \pi.$$

The zeros of $F(\omega)$ should then be near the supports of the delta function components of the power spectrum $R(\omega)$, provided that our original estimate of the background is not too inaccurate.

In the PDFT it is important to select the prior estimate $P(\omega)$ nonzero wherever the function being reconstructed is nonzero; for the IPDFT the situation is different. Comparing equation (49.2) with equation (30.2) we see that in the IPDFT the true $R(\omega)$ is playing the role previously given to $P(\omega)$, while $P(\omega)$ is in the role previously played by the function we wished to estimate, which, in the IPDFT, is $R(\omega)$. It is important, therefore, that $R(\omega)$ not be zero where $P(\omega) \neq 0$; that is, we should choose the $P(\omega) = 0$ wherever $R(\omega) = 0$. Of course, we usually do not know the support of $R(\omega)$ *a priori*. The point is simply that it is better to make $P(\omega) = 0$ than to make it nonzero, if we have any doubt as to the value of $R(\omega)$.

In our discussion of the MEM we obtained an estimate for the function $R(\omega)$, not simply a way of locating the delta function components. As we shall show, the IPDFT can also be used to estimate $R(\omega)$. Although the resulting estimate is not guaranteed to be either nonnegative nor data consistent it usually is both of these.

For any function $G(\omega)$ on $[-\pi, \pi]$ with Fourier series

$$G(\omega) = \sum_{n=-\infty}^{\infty} g(n)e^{in\omega}$$

the *additive causal part* of the function $G(\omega)$ is

$$G_+(\omega) = \sum_{n=0}^{\infty} g(n)e^{in\omega}.$$

Any function such as $G_+$ that has Fourier coefficients that are zero for negative indices is called a *causal function*. The equation (49.2) then says that the two causal functions $P_+$ and $(FR)_+$ have Fourier coefficients that agree for $m = 0, 1, ..., N$.

Because $F(\omega)$ is a finite causal trigonometric polynomial we can write

$$(FR)_+(\omega) = R_+(\omega)F(\omega) + J(\omega),$$

where

$$J(\omega) = \sum_{m=0}^{N-1}[\sum_{k=1}^{N-m} r(-k)f(m+k)]e^{im\omega}.$$

Treating $P_+$ as approximately equal to $(FR)_+ = R_+F + J$, we obtain as an estimate of $R_+$ the function $Q = (P_+ - J)/F$. In order for this estimate of $R_+$ to be causal it is sufficient that the function $1/F$ be causal. This means that the trigonometric polynomial $F(\omega)$ be minimum phase; that is, all its roots lie outside the unit circle. In the chapter on MEM we saw that this is always the case for MEM. It is not always the case for the IPDFT, but it is usually the case in practice; in fact, it was difficult (but possible)

to construct a counterexample. We then construct our IPDFT estimate of $R(\omega)$, which is

$$R_{IPDFT}(\omega) = 2\text{Re}(Q(\omega)) - r(0).$$

The IPDFT estimate is real-valued and, when $1/F$ is causal, guaranteed to be data consistent. Although this estimate is not guaranteed to be nonnegative, it usually is.

We showed in the chapter on entropy maximization that the vector $\mathbf{a}$ that solves $R\mathbf{a} = \delta$ corresponds to a polynomial $A(z)$ having all its roots on or outside the unit circle; that is, it is minimum phase. The IPDFT involves the solution of the system $R\mathbf{f} = \mathbf{p}$, where $\mathbf{p} = (p(0), ..., p(N))^T$ is the vector of initial Fourier coefficients of another power spectrum, $P(\omega) \geq 0$ on $[-\pi, \pi]$. When $P(\omega)$ is constant we get $\mathbf{p} = \delta$. For the IPDFT to be data-consistent it is sufficient that the polynomial $F(z) = f_0 + ... + f_N z^N$ be minimum phase. Although this need not be the case, it is usually observed in practice.

**Exercise 1:** Find conditions on the power spectra $R(\omega)$ and $P(\omega)$ that cause $F(z)$ to be minimum phase.
Warning: This is probably not an easy exercise.

The figures below illustrate the IPDFT. The prior function in each case is the box object supported on the central fourth of the interval $[0, 2\pi]$. The value $r(0)$ has been increased slightly to regularize the matrix inversion. Figure 49.1 shows the behavior of the IPDFT when the object is only the box. Contrast this with the behavior of MEM in this case, as seen in Figure 48.4. Figures 49.2 and 49.3 show the abilty of the IPDFT to resolve the two spikes at $0.95\pi$ and $1.05\pi$ against the box background. Again, contrast this with the MEM reconstructions in Figures 48.5 and 48.6. To show that the IPDFT is actually indicating the presence of the spikes and not just rolling across the top of the box, we reconstruct two unequal spikes in Figure 49.4. Figure 49.5 shows how the IPDFT behaves when we increase the number of data points; now $N = 25$ and the SNR is very low.

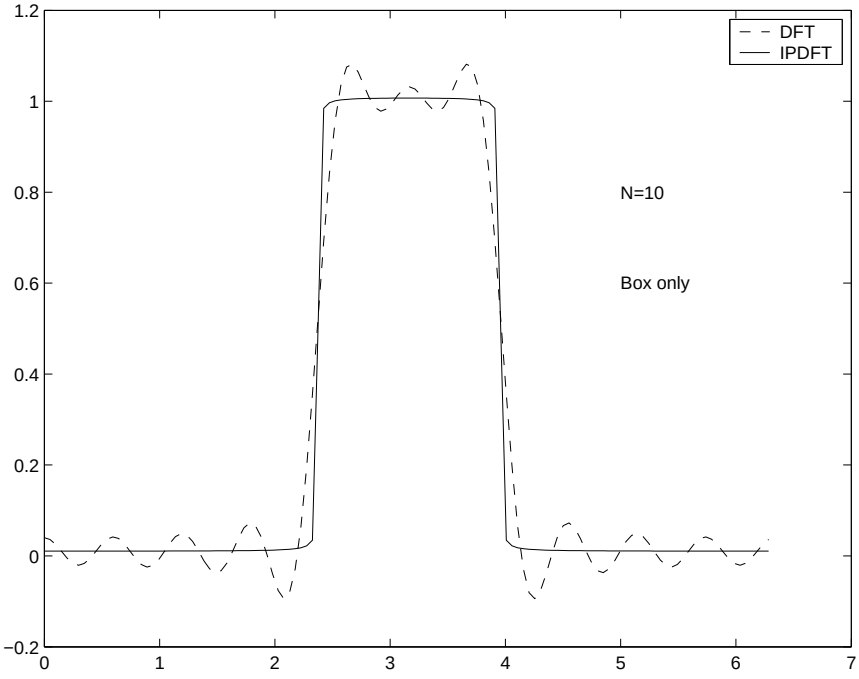



Figure 49.1: The DFT and IPDFT: box only, $N = 1$

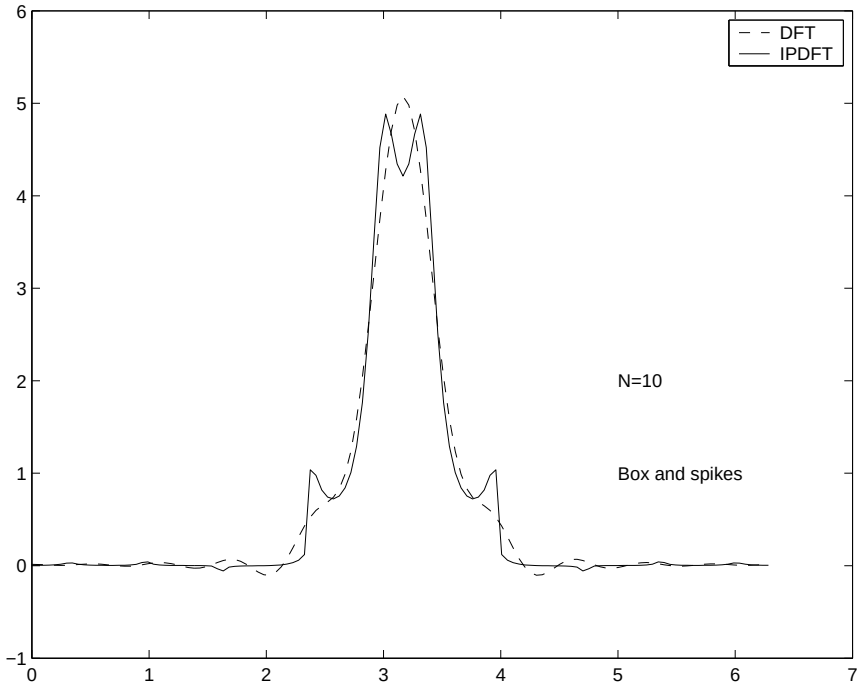



Figure 49.2: The DFT and IPDFT, box and two spikes, $N = 10$, high SNR

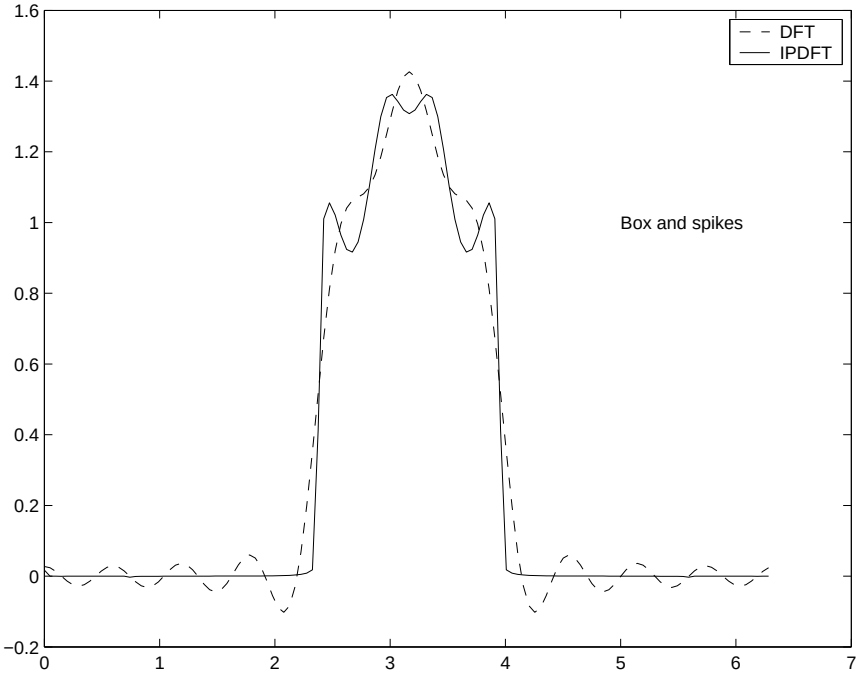



Figure 49.3: The DFT and IPDFT, box and two spikes, $N = 10$, moderate SNR

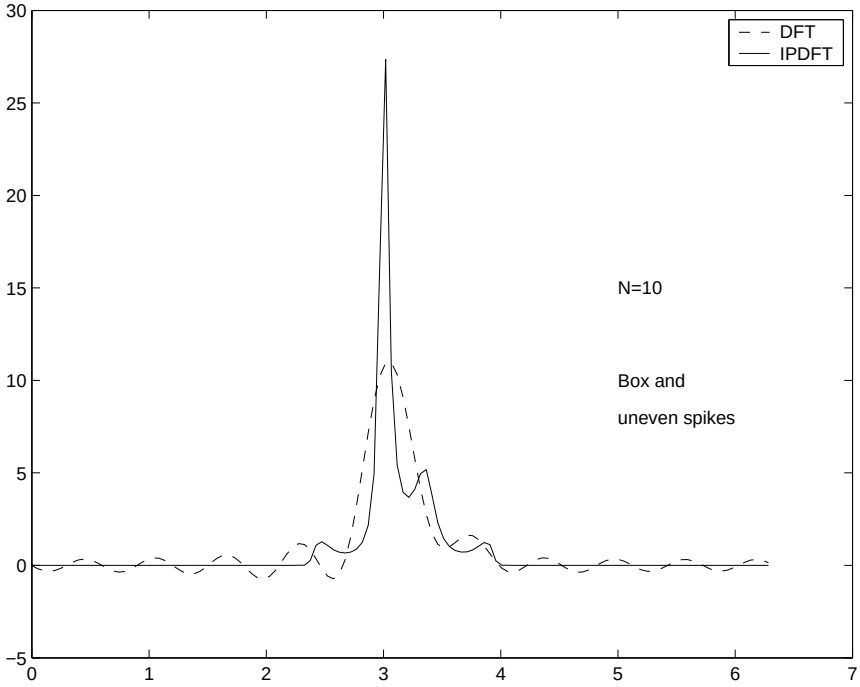



Figure 49.4: The DFT and IPDFT, box and unequal spikes, $N = 10$, high SNR

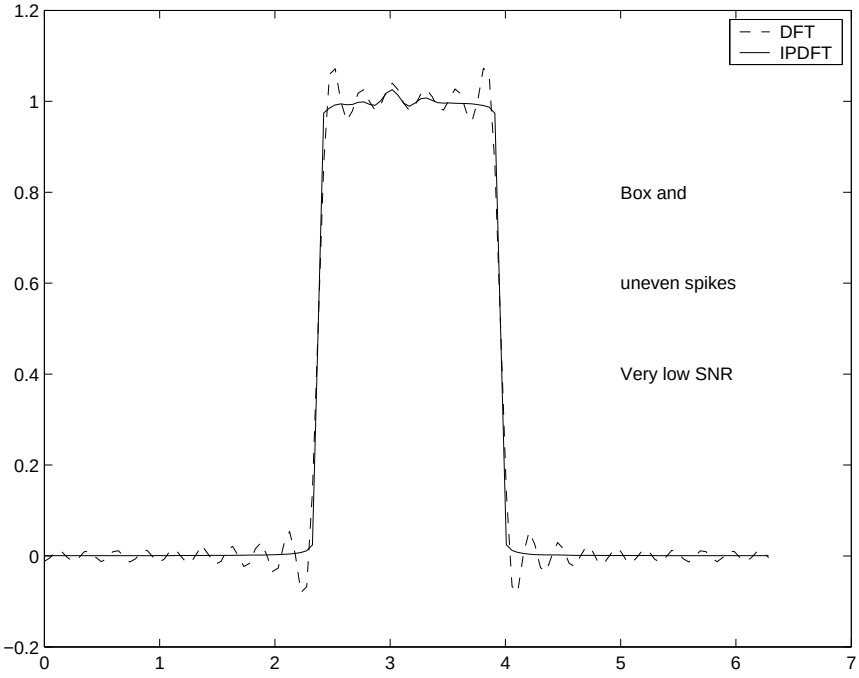



Figure 49.5: The DFT and IPDFT, box and unequal spikes, $N = 25$, very low SNR

# Chapter 50

# Prony's Method

The date of publication of [159] is often taken by editors to be a typographical error and is replaced by 1995, or, since it is not written in English, perhaps 1895. But the 1795 date is the correct one. The mathematical problem Prony solved arises also in signal processing and his method for solving it is still used today. Prony's method is also the inspiration for the eigenvector methods described in our next chapter.

**Prony's problem:** Prony considers a function of the form

$$s(t) = \sum_{n=1}^{N} a_n e^{\gamma_n t}, \tag{50.1}$$

where we allow the $a_n$ and the $\gamma_n$ to be complex. If we take the $\gamma_n = i\omega_n$ to be imaginary $s(t)$ becomes the sum of complex exponentials; if we take $\gamma_n$ to be real, then $s(t)$ is the sum of real exponentials, either increasing with $t$ or decreasing with $t$. The problem is to determine from samples of $s(t)$ the number $N$, the $\gamma_n$ and the $a_n$.

**Prony's method:** Suppose that we have data $y_m = s(m\Delta)$, for some $\Delta > 0$ and for $m = 1, ..., M$, where we assume that $M = 2N$. We seek a vector $\mathbf{c}$ with entries $c_j$, $j = 0, ..., N$ such that

$$c_0 y_{k+1} + c_1 y_{k+2} + c_2 y_{k+3} + ... + c_N y_{k+N+1} = 0, \tag{50.2}$$

for $k = 0, 1, ..., M - N - 1$. So we want a complex vector $\mathbf{c}$ in $C^{N+1}$ orthogonal to $M - N = N$ other vectors. In matrix-vector notation we are

solving the linear system

$$\begin{bmatrix} y_1 & y_2 & \cdots & y_{N+1} \\ y_2 & y_3 & \cdots & y_{N+2} \\ \cdot & & & \\ \cdot & & & \\ \cdot & & & \\ y_N & y_{N+1} & \cdots & y_M \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ \cdot \\ \cdot \\ \cdot \\ c_N \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{bmatrix},$$

which we write as $Y\mathbf{c} = \mathbf{0}$. Since $Y^\dagger Y\mathbf{c} = \mathbf{0}$ also, we see that $\mathbf{c}$ is an eigenvector associated with the eigenvalue zero of the hermitian nonnegative definite matrix $Y^\dagger Y$.

Fix a value of $k$ and replace each of the $y_{k+j}$ in equation (50.2) with the value given by equation (50.1) to get

$$0 = \sum_{n=0}^{N} a_n [\sum_{j=0}^{N} c_j e^{\gamma_n (k+j+1)\Delta}]$$

$$= \sum_{n=0}^{N} a_n e^{\gamma_n (k+1)\Delta} [\sum_{j=0}^{N} c_j (e^{\gamma_n \Delta})^j].$$

Since this is true for each of the $N$ fixed values of $k$, we conclude that the inner sum is zero for each $n$; that is,

$$\sum_{j=0}^{N} c_j (e^{\gamma_n \Delta})^j = 0,$$

for each $n$. Therefore, the polynomial

$$C(x) = \sum_{j=0}^{N} c_j x^j$$

has for its roots the $N$ values $x = e^{\gamma_n \Delta}$. Once we find the roots of this polynomial we have the values of $\gamma_n$. Then we obtain the $a_n$ by solving a linear system of equations. In practice we would not know $N$ so would overestimate $N$ somewhat in selecting $M$. As a result, some of the $a_n$ would be zero.

If we believe that the number $N$ is considerably smaller than $M$, we do not assume that $2N = M$. Instead, we select $L$ somewhat larger than we believe $N$ is and then solve the linear system

$$\begin{bmatrix} y_1 & y_2 & \cdots & y_{L+1} \\ y_2 & y_3 & \cdots & y_{L+2} \\ \cdot & & & \\ \cdot & & & \\ \cdot & & & \\ \cdot & & & \\ y_{M-L} & y_{M-L+1} & \cdots & y_M \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ \cdot \\ \cdot \\ \cdot \\ c_L \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \\ 0 \end{bmatrix}.$$

This system has $M-L$ equations and $L+1$ unknowns, so is quite overdetermined. We would then use the least squares aproach to obtain the vector $\mathbf{c}$. Again writing the system as $Y\mathbf{c} = \mathbf{0}$, we note that the matrix $Y^\dagger Y$ is $L+1$ by $L+1$ and has $\lambda = 0$ for its lowest eigenvalue; therefore it is not invertible. When there is noise in the measurements this matrix may become invertible, but will still have at least one very small eigenvalue.

Finding the vector $\mathbf{c}$ in either case can be tricky, because we are looking for a nonzero solution of a homogeneous system of linear equations. For a discussion of the numerical issues involved in these calculations the interested reader should consult the book by Therrien [174].

# Chapter 51

# Eigenvector Methods

Prony's method showed that information about the signal can sometimes be obtained from the roots of certain polynomials formed from the data. Eigenvector methods assume the data is correlation values and involve polynomials formed from the eigenvectors of the correlation matrix. Schmidt's *multiple signal classification* (MUSIC) algorithm is one such method [163]. A related technique used in direction-of-arrival array processing is the *estimation of signal parameters by rotational invariance techniques* (ESPRIT) of Paulraj, Roy and Kailath [154].

We suppose now that the function $f(t)$ being measured is signal plus noise, with the form

$$f(t) = \sum_{j=1}^{J} A_j e^{i\theta_j} e^{i\omega_j t} + n(t) = s(t) + n(t),$$

where the phases $\theta_j$ are random variables, independent and uniformly distributed in the interval $[0, 2\pi)$ and $n(t)$ denotes the random complex stationary noise component. Assume that $E(n(t)) = 0$ for all $t$ and that the noise is independent of the signal components. We want to estimate $J$, the number of sinusoidal components, their magnitudes $|A_j|$ and their frequencies $\omega_j$.

The autocorrelation function associated with $s(t)$ is

$$r_s(\tau) = \sum_{j=1}^{J} |A_j|^2 e^{-i\omega_j \tau}$$

and the signal power spectrum is the Fourier transform of $r_s(\tau)$,

$$R_s(\omega) = \sum_{j=1}^{J} |A_j|^2 \delta(\omega - \omega_j).$$

The noise autocorrelation is denoted $r_n(\tau)$ and the noise power spectrum is denoted $R_n(\omega)$. For the remainder of this section we shall assume that the noise is *white noise*, that is, $R_n(\omega)$ is constant and $r_n(\tau) = 0$ for $\tau \neq 0$.

We collect samples of the function $f(t)$ and use them to estimate some of the values of $r_s(\tau)$. From these values of $r_s(\tau)$ we estimate $R_s(\omega)$, primarily looking for the locations $\omega_j$ at which there are delta functions.

We assume that the samples of $f(t)$ have been taken over an interval of time sufficiently long to take advantage of the independent nature of the phase angles $\theta_j$ and the noise. This means that when we estimate the $r_s(\tau)$ from products of the form $f(t+\tau)\overline{f(t)}$ the cross terms between one signal component and another, as well as between a signal component and the noise, are nearly zero, due to destructive interference coming from the random phases.

Suppose now that we have the values $r_f(m)$ for $m = -(M-1), ..., M-1$, where $M > J$, $r_f(m) = r_s(m)$ for $m \neq 0$ and $r_f(0) = r_s(0) + \sigma^2$, for $\sigma^2$ the variance (or *power*) of the noise. We form the $M$ by $M$ autocorrelation matrix $R$ with entries $R_{m,k} = r_f(m-k)$.

**Exercise 1:** Show that the matrix $R$ has the following form:

$$R = \sum\nolimits_{j=1}^{J} |A_j|^2 \mathbf{e}_j \mathbf{e}_j^\dagger + \sigma^2 I,$$

where $\mathbf{e}_j$ is the column vector with entries $e^{-i\omega_j m}$, for $m = -(M-1), ..., M-1$.

Let $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_M > 0$ be the eigenvalues of $R$ and let $\mathbf{u}^m$ be a norm-one eigenvector associated with $\lambda_m$.

**Exercise 2:** Show that $\lambda_m = \sigma^2$ for $m = J+1, ..., M$, while $\lambda_m > \sigma^2$ for $m = 1, ..., J$. Hint: since $M > J$ the $M - J$ orthogonal eigenvectors $\mathbf{u}^m$ corresponding to $\lambda_m$ for $m = J+1, ..., M$ will be orthogonal to each of the $\mathbf{e}_j$. Then consider the quadratic forms $(\mathbf{u}^m)^\dagger R \mathbf{u}^m$.

By calculating the eigenvalues of $R$ and noting how many of them are greater than the smallest one we find $J$. Now we seek the $\omega_j$.

For each $\omega$ let $\mathbf{e}_\omega$ have the entries $e^{-i\omega m}$ and form the function

$$T(\omega) = \sum\nolimits_{m=J+1}^{M} |\mathbf{e}_\omega^\dagger \mathbf{u}^m|^2.$$

This function $T(\omega)$ will have zeros at precisely the values $\omega = \omega_j$, for $j = 1, ..., J$. Once we have determined $J$ and the $\omega_j$ we estimate the magnitudes $|A_j|$ using Fourier transform estimation techniques already discussed. This is basically Schmidt's MUSIC method.

We have made several assumptions here that may not hold in practice and we must modify this eigenvector approach somewhat. First, the time over which we are able to measure the function $f(t)$ may not be long enough

to give good estimates of the $r_f(\tau)$. In that case we may work directly with the samples of $f(t)$. Second, the smallest eigenvalues will not be exactly equal to $\sigma^2$ and some will be larger than others. If the $\omega_j$ are not well separated, or if some of the $|A_j|$ are quite small, it may be hard to tell what the value of $J$ is. Third, we often have measurements of $f(t)$ that have errors other than those due to background noise; inexpensive sensors can introduce their own random phases that can complicate the estimation process. Finally, the noise may not be white, so that the estimated $r_f(\tau)$ will not equal $r_s(\tau)$ for $\tau \neq 0$, as above. If we know the noise power spectrum or have a decent idea what it is we can perform a *prewhitening* to $R$, which will then return us to the case considered above, although this can be a tricky procedure.

When the noise power spectrum has a component that is not white the eigenvalues and eigenvectors of $R$ behave somewhat differently from the white noise case. The eigenvectors tend to separate into three groups. Those in the first group correspond to the smallest eigenvalues and are approximately orthogonal to both the signal components and the nonwhite noise component. Those in the second group, whose eigenvalues are somewhat larger than those in the previous group, tend to be orthogonal to the signal components but to have a sizable projection onto the nonwhite noise component. Those in the third group, with the largest eigenvalues, have sizable projection onto both the signal and nonwhite noise components. Since the DFT estimate uses $R$, as opposed to $R^{-1}$, the DFT spectrum is determined largely by the eigenvectors in the third group. The MEM estimator, which uses $R^{-1}$, makes most use of the eigenvectors in the first group, but in the formation of the denominator. In the presence of a nonwhite noise component the orthogonality of those eigenvectors to both the signals and the nonwhite noise shows up as peaks throughout the region of interest, masking or distorting the signal peaks we wish to see.

There is a second problem exacerbated by the nonwhite component-sensitivity of nonlinear and eigenvector methods to phase errors. We have assumed up to now that the data we have obtained is accurate, but there isn't enough of it. In some cases the machinery used to obtain the measured data may not be of the highest quality; certain applications of SONAR make use of relatively inexpensive hydrophones that will sink into the ocean after they have been used briefly. In such cases the complex numbers $r(n)$ will be distorted. Errors in the measurement of their phases are particularly damaging. The figures below illustrate these issues.

In the figures below the true power spectrum is the box and spikes object used earlier in our discussion of the MEM and IPDFT. It consists of two delta functions at $\omega = 0.95\pi$ and $1.05\pi$, along with a box extending from $0.75\pi$ to $1.25\pi$. There is also a small white noise component that is flat across $[0, 2\pi]$, contributing only to the $r(0)$ value. The data, in the absence of phase errors, is $r(n)$, $|n| \leq N = 25$. Three different amounts of

phase perturbation are introduced in the other cases.

Figure 51.1 shows the function $T(\omega)$ for the two eigenvectors in the second group; here $J = 18$ and $M = 21$. The approximate zeros at $0.95\pi$ and $1.05\pi$ are clearly seen in the error-free case and remain fairly stable as the phase errors are introduced. Figure 51.2 uses the eigenvectors in the first group, with $J = 0$ and $M = 18$. The approximate nulls at $0.95\pi$ and $1.05\pi$ are hard to distinguish even in the error-free case and get progressively worse as phase errors are introduced. Stable nonlinear methods, such as the IPDFT, rely most on the eigenvectors in the second group.

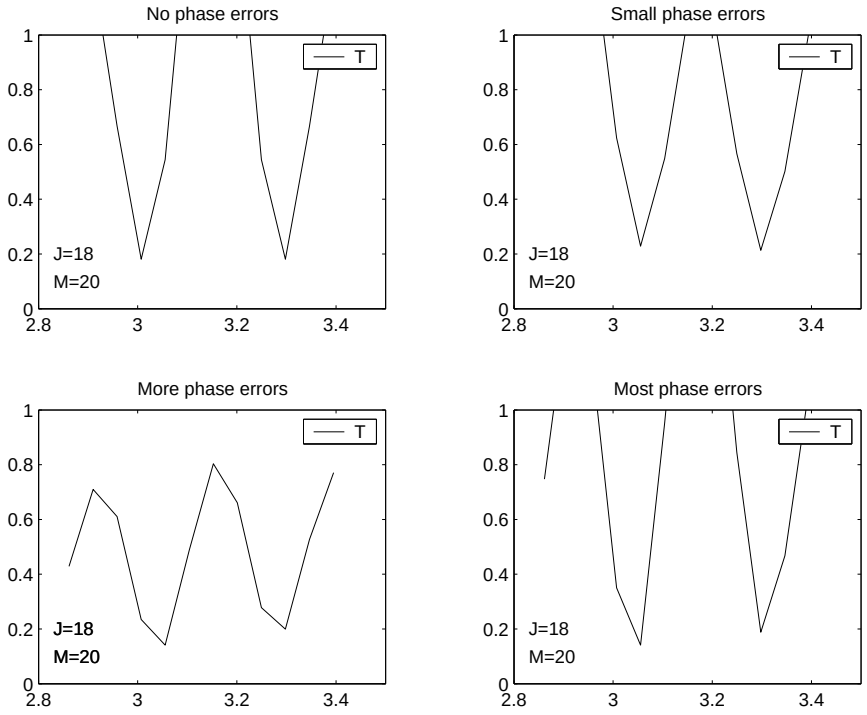



Figure 51.1: $T(\omega)$ for $J = 18$, $M = 21$, varying degrees of phase errors

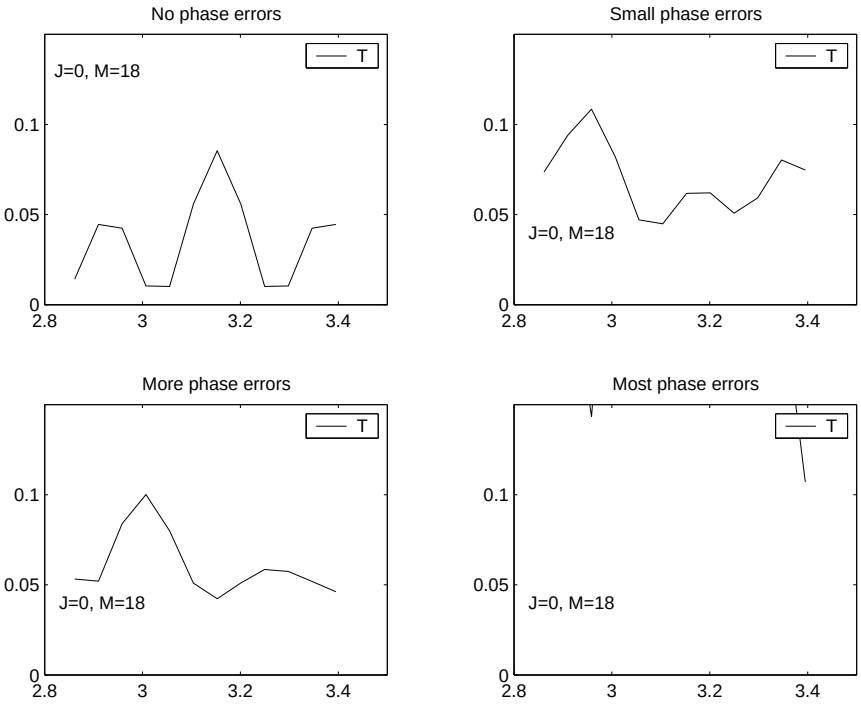



Figure 51.2: $T(\omega)$ for $J = 0$, $M = 18$, varying degrees of phase errors

# Chapter 52

# Resolution Limits

We began in the introductory chapter by saying that our data has been obtained through some form of sensing; physical models, often simplified, describe how the data we have obtained relates to the information we seek; there usually isn't enough data and what we have is corrupted by noise and other distortions. All of the models and algorithms we have considered have as their aim the overcoming of this inherent problem of limited data. But just how limited is the data and in what sense limited? After all, if Burg's maximum entropy method (MEM) resolves peaks that are left unresolved by the DFT, the problem would seem to lie not with the data, which must still retain the desired information, but with the method used. When Burg's MEM produces incorrect reconstructions in the presence of a background that is not flat, but the IPDFT is able to use an estimate of the background to provide a better answer, is it the data or the method that is limiting? On the other hand, when we say MEM has produced an incorrect answer what do we mean? We know that MEM gives a positive estimate of the power spectrum that is exactly consistent with the autocorrelation data; it is only incorrect because we know the true spectrum, having created it in our simulations. Such questions concern everyone using inversion methods, and yet have no completely satisfying answers. Bertero's paper [11] is a good place to start one's education in these matters. In this chapter we consider some of these issues, in so far as they concern the methods we have discussed in this text.

## The DFT:

The exercise following our discussion of the second approach to signal analysis uses the DFT to illustrate the notion of *resolution limit*. The signal there was the sum of two sinusoids, at frequencies $\omega_1 = -\alpha$ and $\omega_2 = \alpha$. As the $\alpha$ approached zero resolution in the DFT was eventually lost; for

larger data lengths the $\alpha$ could be smaller before this happened. We know from successful application of high-resolution methods that this does not mean that the information about the two sinusoids has been lost. What does it mean?

The DFT shows up almost everywhere in signal processing. As a finite Fourier series it can be viewed as a best approximation of the infinite Fourier series; as a matched filter it is the optimal linear method for detecting a single sinusoid in white noise. However, it is not the optimal linear method for detecting two sinusoids in white noise. If we know that the signal is the sum of two sinusoids (with equal amplitudes, for now) in additive white noise, the optimal linear filter is a matched filter of the form $\mathbf{e}_{\alpha\beta}^{\dagger}\mathbf{d}$, where $\mathbf{d}$ is the data vector and $\mathbf{e}_{\alpha\beta}$ is the data we would have received had the signal consisted solely of $e^{i\alpha t} + e^{i\beta t}$. The output of the matched filter is a function of the two variables $\alpha$ and $\beta$. We plot the magnitude of this function of two variables and select the pair for which the magnitude is greatest. If we apply this procedure to the signal in the exercise we would find that we could still determine that there are sinusoids at $\alpha$ and $\beta = -\alpha$. The DFT manages to resolve sinusoids when they are far enough apart to be treated as two separate signals, each with a single sinusoid. Otherwise, the DFT is simply not the proper estimate of frequency location for multiple sinusoids. A proper notion of resolution limit should be based on something other than the behavior of the DFT in the presence of two sinusoids.

## Bandlimited extrapolation reconsidered:

Suppose we want to estimate the function $F(\omega)$, known to be zero for $|\omega| > \Omega$, where $0 < \Omega < \pi$. Our data will be samples of the inverse Fourier transform, $f(x)$. Suppose, in addition, that we are able to select our finitely many samples only for $x$ within the bounded interval $[0, X]$, but are otherwise unrestricted; that is, we can take as many samples at whichever $x$ values we wish. What should we do?

Shannon's *sampling theorem* tells us that we can reconstruct $F(\omega)$ exactly if we know the values $f(n\frac{\pi}{\Omega})$ for all the integers $n$. Then we have

$$F(\omega) = \frac{\pi}{\Omega}\sum\nolimits_{n=-\infty}^{\infty} f(n\frac{\pi}{\Omega})e^{in\frac{\pi}{\Omega}\omega}.$$

The sampling rate of $\Delta = \frac{\pi}{\Omega}$ is the *Nyquist rate* and the doubly infinite sequence of samples at this rate is all we need. But, of course, we cannot actually measure infinitely many values of $f(x)$. Furthermore, we are restricted to the interval $[0, X]$. If

$$(N-1)\frac{\pi}{\Omega} \leq X < N\frac{\pi}{\Omega}$$

then there are $N$ Nyquist samples available within the interval $[0, X]$. Some have concluded that the sampling theorem tells us that we can do no better than to take the $N$ samples $f(n\frac{\pi}{\Omega})$, $n = 0, 1, ..., N-1$, that we have $N$ *degrees of freedom* in selecting data from within the interval $[0, X]$ and our freedom is thus exhausted when we have taken these $N$ samples. The questions are: Can we do better? and Is there a quantifiable limit to our freedom to extract information under these restrictions? If someone offered to give you the value of $f(x)$ at one new point $x$ within the interval $[0, X]$, would you take it?

No one would argue that the $N$ Nyquist samples determine completely the values of $f(x)$ for the remaining $x$ within the interval $[0, X]$. The problem is more how to use this new data value. The DFT

$$F_{DFT}(\omega) = \frac{\pi}{\Omega}\chi_\Omega(\omega)\sum\nolimits_{n=0}^{N-1} f(n\frac{\pi}{\Omega})e^{in\frac{\pi}{\Omega}\omega}$$

is zero outside the interval $[-\Omega, \Omega]$, is consistent with the data and therefore could be the right answer. If we are given the additional value $f(a)$ the estimate

$$\frac{\pi}{\Omega}\chi_\Omega(\omega)[f(a)e^{ia\omega} + \sum\nolimits_{n=0}^{N-1} f(n\frac{\pi}{\Omega})e^{in\frac{\pi}{\Omega}\omega}]$$

is not consistent with the data.

Using the non-iterative bandlimited extrapolation estimate given in equation (29.7) we can get an estimate with is consistent with this no longer uniformly spaced data as well as with the band limitation. So it is possible to make good use of the additional sample offered to us; we should accept it. Is there no end to this, however? Should we simply take as many samples as we desire, equispaced or not? Is there some limit to our freedom to squeeze information out of the behavior of the function $f(x)$ within the interval $[0, X]$? The answer is Yes, there are limits, but the limits depend in sometimes subtle ways on themethod being used and the amount and nature of the noise involved, which must include round-off error and quantization. Let's consider this more closely, with respect to the non-iterative bandlimited extrapolation method.

As we saw earlier, the non-iterative Gerchberg-Papoulis bandlimited extrapolation method leads to the estimate

$$F_\Omega(\omega) = \chi_\Omega(\omega)\sum\nolimits_{m=1}^{M} \frac{1}{\lambda_m}(\mathbf{u}^m)^\dagger \mathbf{d}\, U^m(\omega),$$

where $\mathbf{d}$ is the data vector. In contrast, the DFT estimate is

$$F_{DFT}(\omega) = \sum\nolimits_{m=1}^{M} (\mathbf{u}^m)^\dagger \mathbf{d}\, U^m(\omega).$$

The estimate $F_\Omega(\omega)$ can provide better resolution within the interval $[-\Omega, \Omega]$ because of the multiplier $1/\lambda_m$, causing the estimate to rely more heavily on

those functions $U_m(\omega)$ having more roots, therefore more structure, within that interval. But therein lies the danger, as well.

When the data is noise-free the dot product $(\mathbf{u}^m)^\dagger \mathbf{d}$ is relatively small for those eigenvectors $\mathbf{u}_m$ corresponding to the small eigenvalues; therefore the product $(1/\lambda_m)(\mathbf{u}^m)^\dagger \mathbf{d}$ is not large. However, when the data vector $\mathbf{d}$ contains noise, the dot product of the noise component with each of the eigenvectors is about the same size. Therefore, the product $(1/\lambda_m)(\mathbf{u}^m)^\dagger \mathbf{d}$ is now quite large and the estimate is dominated by the noise. This sensitivity to the noise is the limiting factor in the bandlimited extrapolation. Any reasonable definitions of *degrees of freedom* and *resolution limit* must include the signal-to-noise ratio, as well as the fall-off rate of the eigenvalues of the matrix. In our bandlimited extrapolation problem the matrix is the sinc matrix. The proportion of nearly zero eigenvalues will be approximately $1 - \frac{\Omega}{\pi}$; the smaller the ratio $\frac{\Omega}{\pi}$ the fewer essentially nonzero eigenvalues there will be. For other extrapolation methods, such as the PDFT, the fall-off rate may be somewhat different. For analogous methods in higher dimensions the fall-off rate may be quite different [11].

## High-resolution methods:

The bandlimited extrapolation methods we have studied are linear in the data, while the high-resolution methods are not. The high-resolution methods we have considered, such as MEM, Capon's method, the IPDFT and the eigenvector techniques, exploit the fact that the frequencies of sinusoidal components can be associated with the roots of certain polynomials obtained from eigenvectors of the autocorrelation matrix. When the roots are disturbed by phase errors or are displaced by the presence of a non-flat background, the methods that use these roots perform badly. As we mentioned earlier, there is some redundancy in the storage of information in these roots and stable processing is still possible in many cases. Not all the eigenvectors store this information and a successful method must interrogate the ones that do. Additive white noise causes MEM to fail by increasing all the eigenvalues, but does not hurt explicit eigenvector methods. Correlated noise that cannot be effectively prewhitened hurts all these methods, by making it more difficult to separate the information-bearing eigenvectors from the others. Correlation between sinusoidal components, as may occur in multipath arrivals in shallow water, causes additional difficulty, as does short data length, which corrupts the estimates of the autocorrelation values.

# Chapter 53

# A Little Probability Theory

In this chapter we review a few important results from the theory of probability that will be needed later.

**Averaging independent random variables:** Let $X_1, ..., X_N$ be $N$ independent random variables with the same mean (that is, expected value) $\mu$ and same variance $\sigma^2$. Then the *sample average*

$$\bar{X} = N^{-1} \sum_{n=1}^{N} X_n$$

has $\mu$ for its mean and $\sigma^2/N$ for its variance.

**Exercise 1:** Prove these two assertions.

**Maximum likelihood estimation- an example:** Let $\theta$ in the interval $[0, 1]$ be the unknown probability of success on one trial of a binomial distribution (a coin flip, for example), so that the probability of $x$ successes in $n$ trials is $L(\theta, x, n) = \frac{n!}{x!(n-x)!}\theta^x(1-\theta)^{n-x}$, for $x = 0, 1, ..., n$. If we have observed $n$ trials and have recorded $x$ successes we can estimate $\theta$ by selecting that $\hat{\theta}$ for which $L(\theta, x, n)$ is maximized as a function of $\theta$. This estimator is called the *maximum likelihood* estimator.

**Exercise 2:** Show that, for the binomial case described above, the maximum likelihood estimate of $\theta$ is $\hat{\theta} = x/n$.

**The Poisson distribution:** A random variable $X$ taking on only nonnegative integer values is said to have the *Poisson distribution* with parameter

$\lambda > 0$ if, for each nonnegative integer $k$, the probability $p_k$ that $X$ will take on the value $k$ is given by

$$p_k = e^{-\lambda}\lambda^k/k!.$$

**Exercise 3:** Show that the sequence $\{p_k\}_{k=0}^{\infty}$ sums to one.

**Exercise 4:** Show that the expected value $E(X)$ is $\lambda$, where the expected value in this case is

$$E(X) = \sum_{k=0}^{\infty} kp_k.$$

**Exercise 5:** Show that the variance of $X$ is also $\lambda$, where the variance of $X$ in this case is

$$\mathrm{var}(X) = \sum_{k=0}^{\infty}(k-\lambda)^2 p_k.$$

**Sums of independent Poisson random variables:** Let $Z_1, ..., Z_N$ be independent Poisson random variables with expected value $E(Z_n) = \lambda_n$. Let $\mathbf{Z}$ be the random vector with $Z_n$ as its entries, $\lambda$ the vector whose entries are the $\lambda_n$ and $\lambda_+ = \sum_{n=1}^{N}\lambda_n$. Then the probability function for $\mathbf{Z}$ is

$$f(\mathbf{Z}|\lambda) = \prod_{n=1}^{N}\lambda_n^{z_n}\exp(-\lambda_n)/z_n! = \exp(-\lambda_+)\prod_{n=1}^{N}\lambda_n^{z_n}/z_n!\ . \tag{53.1}$$

Now let $Y = \sum_{n=1}^{N} Z_n$. Then, the probability function for $Y$ is

$$\begin{aligned}\mathrm{Prob}(Y = y) &= \mathrm{Prob}(Z_1 + ... + Z_N = y)\\ &= \sum_{z_1+...z_N=y}\exp(-\lambda_+)\prod_{n=1}^{N}\lambda_n^{z_n}/z_n!\ .\end{aligned} \tag{53.2}$$

But, as we shall show shortly, we have

$$\sum_{z_1+...z_N=y}\exp(-\lambda_+)\prod_{n=1}^{N}\lambda_n^{z_n}/z_n! = \exp(-\lambda_+)\lambda_+^y/y!\,. \tag{53.3}$$

Therefore, $Y$ is a Poisson random variable with $E(Y) = \lambda_+$.

If we observe an instance of $y$, we then can consider the conditional distribution $f(\mathbf{Z}|\lambda, y)$ of $\{Z_1, ..., Z_N\}$, subject to $y = Z_1 + ... + Z_N$. We have

$$f(\mathbf{Z}|\lambda, y) = \frac{y!}{z_1!...z_N!}(\frac{\lambda_1}{\lambda_+})^{z_1}...(\frac{\lambda_N}{\lambda_+})^{z_N}. \tag{53.4}$$

This is a *multinomial distribution*. Given $y$ and $\lambda$ the conditional expected value of $Z_n$ is then $E(Z_n|\lambda, y) = y\lambda_n/\lambda_+$. To see why (53.3) is true, we discuss the multinomial distribution a bit.

**The multinomial distribution:** When we expand the quantity $(a_1 + ... + a_N)^y$ we obtain a sum of terms, each of the form $a_1^{z_1}...a_N^{z_N}$, with $z_1 + ... + z_N = y$. How many terms of the same form are there? There are $N$ variables. We are to select $z_n$ of type $n$, for each $n = 1, ..., N$, to get $y = z_1 + ... + z_N$ factors. Imagine $y$ blank spaces, to be filled in by various factor types as we do the selection. We select $z_1$ of these blanks and mark them $a_1$, for type one. We can do that in $\binom{y}{z_1}$ ways. We then select $z_2$ of the remaining blank spaces and enter $a_2$ in them; we can do this in $\binom{y-z_1}{z_2}$ ways. Continuing this way we find that we can select the $N$ factor types in

$$\binom{y}{z_1}\binom{y-z_1}{z_2}...\binom{y-(z_1+...+z_{N-2})}{z_{N-1}} \tag{53.5}$$

ways, or in

$$\frac{y!}{z_1!(y-z_1)!}...\frac{(y-(z_1+...+z_{N-2}))!}{z_{N-1}!(y-(z_1+...+z_{N-1}))!} = \frac{y!}{z_1!...z_N!}. \tag{53.6}$$

This tells us in how many different sequences the factor types can be selected. Applying this we get the multinomial theorem:

$$(a_1 + ... + a_N)^y = \sum_{z_1+...+z_N=y} \frac{y!}{z_1!...z_N!} a_1^{z_1}...a_N^{z_N}. \tag{53.7}$$

Select $a_n = \lambda_n/\lambda_+$. Then

$$\begin{aligned} 1 = 1^y &= (\frac{\lambda_1}{\lambda_+} + ... + \frac{\lambda_N}{\lambda_+})^y \\ &= \sum_{z_1+...+z_N=y} \frac{y!}{z_1!...z_N!}(\frac{\lambda_1}{\lambda_+})^{z_1}...(\frac{\lambda_N}{\lambda_+})^{z_N}. \end{aligned} \tag{53.8}$$

From this we get

$$\sum_{z_1+...z_N=y} \exp(-\lambda_+) \prod_{n=1}^{N} \lambda_n^{z_n}/z_n! = \exp(-\lambda_+)\lambda_+^y/y! \ . \tag{53.9}$$

**Gaussian random variables:** A real-valued random variable $X$ is called *Gaussian* or *normal* with mean $\mu$ and variance $\sigma^2$ if its probabilty density function (pdf) is

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp(-\frac{(x-\mu)^2}{2\sigma^2}). \tag{53.10}$$

In the statistical literature a normal random variable is *standard* if its mean is $\mu = 0$ and its variance is $\sigma^2 = 1$.

Suppose now that $Z_1, ..., Z_N$ are independent standard normal random variables. Then their joint pdf is the function

$$f(z_1, ..., z_N) = \prod_{n=1}^{N} \frac{1}{\sqrt{2\pi}} \exp(-\frac{1}{2} z_n^2) = \frac{1}{(\sqrt{2\pi})^N} \exp(-\frac{1}{2}(z_1^2 + ... + z_N^2)).$$

By taking linear combinations of these random variables we can obtain a new set of normal random variables that are no longer independent. For each $m = 1, ..., M$ let

$$X_m = \sum\nolimits_{n=1}^{N} A_{mn} Z_n.$$

Then $E(X_m) = 0$.

The *covariance matrix* associated with the $X_m$ is the matrix $R$ with entries $R_{mn} = E(X_m X_n)$, $m, n = 1, 2, ..., M$. We have

$$E(X_m X_n) = \sum\nolimits_{k=1}^{N} A_{mk} \sum\nolimits_{j=1}^{N} A_{nj} E(Z_k Z_j).$$

Since the $Z_n$ are independent with mean zero, we have $E(Z_k Z_j) = 0$ for $k \neq j$ and $E(Z_k^2) = 1$. Therefore,

$$E(X_m X_n) = \sum\nolimits_{k=1}^{N} A_{mk} A_{nk},$$

and the covariance matrix is $R = AA^T$.

Writing $\mathbf{X} = (X_1, ..., X_M)^T$ and $\mathbf{Z} = (Z_1, ..., Z_N)^T$ we have $\mathbf{X} = A\mathbf{Z}$, where $A$ is the $M$ by $N$ matrix with entries $A_{mn}$. Using the standard formulas for changing variables, we find that the joint pdf for the random variables $X_1, ..., X_M$ is

$$f(x_1, ..., x_M) = \frac{1}{\sqrt{\det(R)}} \frac{1}{(\sqrt{2\pi})^N} \exp(-\frac{1}{2} \mathbf{x}^T R^{-1} \mathbf{x}),$$

with $\mathbf{x} = (x_1, ..., x_N)^T$. For the remainder of this chapter we limit the discussion to the case of $M = N = 2$ and use the notation $X_1 = X$, $X_2 = Y$ and $f(x_1, x_2) = f(x, y)$. We also let $\rho = E(XY)/\sigma_1\sigma_2$.

The two-dimensional FT of the function $f(x, y)$, the characteristic function of the Gaussian random vector $\mathbf{X}$, is

$$F(\alpha, \beta) = \exp(-\frac{1}{2}(\sigma_1^2\alpha^2 + \sigma_2^2\beta^2 + 2\sigma_1\sigma_2\rho\alpha\beta)).$$

**Exercise 6:** Use partial derivatives of $F(\alpha, \beta)$ to show that $E(X^2Y^2) = 2\sigma_1^2\sigma_2^2\rho^2$.

**Exercise 7:** Show that $E(X^2Y^2) = E(X^2)E(Y^2) + 2E(XY)^2$.

Let $X$ and $Y$ be independent real Gaussian random variables with means $\mu_x$ and $\mu_y$, respectively, and common variance $\sigma^2$. Then $W = X+iY$ is a *complex Gaussian random variable* with mean $\mu_w = E(W) = \mu_x + i\mu_y$ and variance $\sigma_w^2 = 2\sigma^2$.

The results of Exercise 7 extend to complex Gaussian random variables $W$ and $V$. In the complex case we have

$$E(|V|^2|W|^2) = E(|V|^2)E(|W|^2) + |E(V\overline{W})|^2.$$

This is important in optical image processing, where it is called the *Hanbury-Brown Twiss effect* and provides the basis for intensity interferometry [95]. The main point is that we can obtain magnitude information about $E(V\overline{W})$, but not phase information, by measuring the correlation between the magnitudes of $V$ and $W$; that is, we learn something about $E(V\overline{W})$ from intensity measurements. Since we have only the magnitude of $E(V\overline{W})$ we then have a *phase problem*.

# Chapter 54

# Bayesian Methods

We know that to get information out we need to put information in; how to do it is the problem. One approach that is quite popular within the image reconstruction community is the use of statistical Bayesian methods and maximum *a posteriori* (MAP) estimation.

**Conditional probabilities:** Suppose that $A$ and $B$ are two events with positive probabilities $P(A)$ and $P(B)$, respectively. The *conditional probability* of $B$, given $A$, is defined to be $P(B|A) = P(A \cap B)/P(A)$. It follows that Bayes' Rule holds:

$$P(A|B) = P(B|A)P(A)/P(B).$$

To illustrate the use of this rule we consider the following example.

**An example of Bayes' Rule:** Suppose that, in a certain town, ten percent of the adults over fifty have diabetes. The town doctor correctly diagnoses those with diabetes as having the disease ninety-five percent of the time. In two percent of the cases he incorrectly diagnoses those not having the disease as having it. Let $D$ mean that the patient has diabetes, $N$ that the patient does not have the disease, $A$ mean a diagnosis of diabetes is made and $B$ a diagnosis of no diabetes is made. The probability that he will diagnose a given adult as having diabetes is given by the rule of total probability:

$$P(A) = P(A|D)P(D) + P(A|N)P(N).$$

In this example we obtain $P(A) = 0.113$. Now suppose a patient receives a diagnosis of diabetes. What is the probability that this diagnosis is correct? In other words, what is $P(D|A)$? For this we use Bayes' Rule:

$$P(D|A) = P(A|D)P(D)/P(A),$$

which turns out to be 0.84.

**Using prior probabilities:** Nothing so far is controversial. The fun begins when we attempt to broaden the use of Bayes' Rule to ascribe *a priori* probabilities to quantities that are not random. The example used originally by Thomas Bayes in the eighteenth century is as follows. Imagine a billiard table with a line drawn across it parallel to its shorter side, cutting the table into two rectangular regions, the nearer called A and the farther B. Balls are tossed onto the table, coming to rest in either of the two regions. Suppose we are told only that after $N$ such tosses $n$ of the balls ended up in region A. What is the probability that the next ball will end up in region A?

At first it would seem that we cannot answer this question unless we are told the probability of any ball ending up in region A; Bayes argues differently, however. Let $A$ be the event that a ball comes to rest in region A and let $P(A) = x$ be the unknown probability of coming to rest in region A; we may as well consider $x$ to be the relative area of region A, although this is not necessary. Let $D$ be the event that $n$ out of $N$ balls end up in A. Then

$$P(D|x) = \binom{N}{n} x^n (1-x)^{N-n}.$$

Bayes then adopts the view that the horizontal line on the table was randomly positioned so that the unknown $x$ can be treated as a random variable. Using Bayes' Rule we have

$$P(x|D) = P(D|x)P(x)/P(D),$$

where $P(x)$ is the probability density function (pdf) of the random variable $x$, which Bayes takes to be uniform over the interval $[0, 1]$. Therefore we have

$$P(x|D) = c\binom{N}{n} x^n (1-x)^{N-n},$$

where $c$ is chosen so as to make $P(x|D)$ a pdf.

**Exercise 1:** Use integration by parts to show that

$$\binom{N}{n} \int_0^1 x^n (1-x)^{N-n} dx = 1/(N+1),$$

and

$$\binom{N+1}{n+1} \int_0^1 x^{n+1} (1-x)^{N-n} dx = 1/(N+2)$$

for $n = 0, 1, ..., N$.

From the exercise we can conclude that $c = N + 1$. Therefore we have the pdf $P(x|D)$. Now we want to estimate $x$ itself. One way to do this is to calculate the expected value of this pdf, which, according to the exercise, is $(n+1)/(N+2)$. So even though we do not know $x$, we can reasonably say $(n+1)/(N+2)$ is the probability that the next ball will end up in region A, given the behavior of the previous $N$ balls.

There is a second way to estimate $x$; we can find that value of $x$ for which the pdf reaches its maximum. A quick calculation shows this value to be $n/N$. This estimate of $x$ is not the same as the one we calculated using the expected value but they are close for large $N$.

What is controversial here is the decision to treat the positioning of the line as a random act, with the resulting probability $x$ a random variable, as well as the specification of the pdf governing $x$. Even if $x$ were a random variable, we do not necessarily know its pdf. Bayes takes the pdf to be uniform over $[0, 1]$ more as an expression of ignorance than of knowledge. It is this broader use of prior probabilities that is generally known as *Bayesian methods* and not the use of Bayes' Rule itself.

**Maximum** *a posteriori* **estimation:** Bayesian methods provide us with an alternative to maximum likelihood parameter estimation. Suppose that a random variable (or vector) $Z$ has the pdf $f(z;\theta)$, where $\theta$ is a parameter. When this pdf is viewed as a function of $\theta$, not of $z$, it is called the *likelihood function*. Having observed an instance of $Z$, call it $z$, we can estimate the parameter $\theta$ by selecting that value for which the likelihood function $f(z;\theta)$ has its maximum. This is the *maximum likelihood* (ML) estimator. Alternatively, suppose we treat $\theta$ itself as one value of a random variable $\Theta$ having its own pdf, say $g(\theta)$. Then Bayes' Rule says that the conditional pdf of $\Theta$, given $z$, is

$$g(\theta|z) = f(z;\theta)g(\theta)/f(z),$$

where

$$f(z) = \int f(z;\theta)g(\theta)d\theta.$$

The maximum *a posteriori* (MAP) estimate of $\theta$ is the one for which the function $g(\theta|z)$ is maximized. Taking logs and ignoring terms that do not involve $\theta$, we find that the MAP estimate of $\theta$ maximizes the function $\log f(z;\theta) + \log g(\theta)$.

Because the ML estimate maximizes $\log f(z;\theta)$ the MAP estimate is viewed as involving a *penalty term* $\log g(\theta)$ missing in the ML approach. This penalty function is based on the prior pdf $g(\theta)$. We choose $g(\theta)$ in a way that expresses our prior knowledge of the parameter $\theta$.

**MAP reconstruction of images:** In emission tomography the parameter $\theta$ is actually a vectorized image that we wish to reconstruct and the

observed data constitute $z$. Our prior knowledge about $\theta$ may be that the true image is near some prior estimate, say $\rho$, of the correct answer, in which case $g(\theta)$ is selected to peak at $\rho$ [133]. Frequently our prior knowledge of $\theta$ is that the image it represents is nearly constant locally, except for edges. Then $g(\theta)$ is designed to weight more heavily the locally constant images and less heavily the others [99, 103, 134, 107, 137].

**Penalty function methods:** The so-called *penalty function* that appears in the MAP approach comes from a prior pdf for $\theta$. This suggests more general methods that involve a penalty function term that does not necessarily emerge from Bayes' Rule [29]. Such methods are well known in optimization. We are free to estimate $\theta$ as the maximizer of a suitable objective function whether or not that function is a posterior probability. Using penalty function methods permits us to avoid the controversies that accompany Bayesian methods.

# Chapter 55

# Correlation

The *covariance* between two complex-valued random variables $x$ and $y$ is

$$\mathrm{cov}_{xy} = E((x - E(x))(\overline{y - E(y)}))$$

and the *correlation coefficient* is

$$\rho_{xy} = \mathrm{cov}_{xy} / \sqrt{E(|x - E(x)|^2)} \sqrt{E(|y - E(y)|^2)}.$$

The two random variables are said to be *uncorrelated* if and only if $\rho_{xy} = 0$. The *covariance matrix* of a random vector $\mathbf{v}$ is the matrix $Q$ whose entries are the covariances of all the pairs of entries of $\mathbf{v}$. The vector $\mathbf{v}$ is said to be *uncorrelated* if $Q$ is diagonal; otherwise we call $\mathbf{v}$ *correlated*. If the expected value of each of the entries of $\mathbf{v}$ is zero we also have $Q = E(\mathbf{v}\mathbf{v}^\dagger)$. We saw in our discussion of the BLUE that when the noise vector $\mathbf{v}$ is correlated we need to employ the covariance matrix to obtain the best linear unbiased estimator.

We can obtain an $N$ by 1 correlated noise vector $\mathbf{v}$ by selecting a positive integer $K$, an arbitrary $N$ by $K$ matrix $C$, $K$ independent standard normal random variables $z_1, ..., z_K$, that is, their means are zero and their variances are one, and defining $\mathbf{v} = C\mathbf{z}$. Then we have $E(\mathbf{v}) = \mathbf{0}$ and $E(\mathbf{v}\mathbf{v}^\dagger) = CC^\dagger = Q$. In fact, for the Gaussian case this is the only way to obtain a correlated Gaussian random vector. The matrix $C$ producing the covariance matrix $Q$ is not unique.

We can obtain an $N$ by 1 noise vector $\mathbf{v}$ with any given $N$ by $N$ covariance matrix $Q$ using the eigenvalue/eigenvector decomposition of $Q$. In order for $Q$ be be a covariance matrix it is necessary and sufficient that it be Hermitian and nonnegative-definite; that is, $Q^\dagger = Q$ and the eigenvalues of $Q$ are nonnegative. Then, taking $U$ to be the matrix whose columns are the orthonormal eigenvectors of $Q$ and $L$ the diagonal matrix whose diagonal entries are $\lambda_n$, $n = 1, ..., N$, the eigenvalues of $Q$, we have $Q = ULU^\dagger$.

For convenience, we assume that $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_N > 0$. Let $\mathbf{z}$ be a random $N$ by 1 vector whose entries are independent standard normal random variables and let $C = U\sqrt{L}U^\dagger$, the hermitian square root of $Q$. Then $\mathbf{v} = C\mathbf{z}$ has $Q$ for its covariance matrix.

If we write this $\mathbf{v}$ as

$$\mathbf{v} = (U\sqrt{L}U^\dagger)\mathbf{z} = U(\sqrt{L}U^\dagger\mathbf{z}) = U\mathbf{p}$$

then $\mathbf{p} = \sqrt{L}U^\dagger\mathbf{z}$ is uncorrelated; $E(\mathbf{p}\mathbf{p}^\dagger) = L$.

**Principal components:** We can write the entries of the vector $\mathbf{v} = U\mathbf{p}$ as

$$\mathbf{v}_n = \sum_{m=1}^{N} \mathbf{u}_n^m \mathbf{p}_m \tag{55.1}$$

where $\mathbf{u}^m$ is the eigenvector of $Q$ associated with eigenvalue $\lambda_m$. Since the variance of $\mathbf{p}_m$ is $\lambda_m$ equation (55.1) decomposes the vector $\mathbf{v}$ into components of decreasing strength. The terms in the sum corresponding to the smaller indices describe most of $\mathbf{v}$; they are the *principal components* of $\mathbf{v}$. Each $\mathbf{p}_m$ is a linear combination of the entries of $\mathbf{v}$ and *principal component analysis* consists of finding these uncorrelated linear combinations that best describe the correlated entries of $\mathbf{v}$. The representation $\mathbf{v} = U\mathbf{p}$ expresses $\mathbf{v}$ as a linear combination of orthonormal vectors with uncorrelated coefficients. This is analogous to the *Karhunen-Loève expansion* for stochastic processes [4].

Principal component analysis has as its goal the approximation of the covariance matrix $Q = E(\mathbf{v}\mathbf{v}^\dagger)$ by nonnegative-definite matrices of lower rank. A related area is *factor analysis*, which attempts to describe the $N$ by $N$ covariance matrix $Q$ as $Q = AA^\dagger + D$, where $A$ is some $N$ by $J$ matrix, for some $J < N$, and $D$ is diagonal. Factor analysis attempts to account for the correlated components of $Q$ using the lower rank matrix $AA^\dagger$. Underlying this is a model for the random vector $\mathbf{v}$:

$$\mathbf{v} = A\mathbf{x} + \mathbf{w},$$

where both $\mathbf{x}$ and $\mathbf{w}$ are uncorrelated. The entries of the random vector $\mathbf{x}$ are the *common factors* that affect each entry of $\mathbf{v}$ while those of $\mathbf{w}$ are the *special factors*, each associated with a single entry of $\mathbf{v}$. Factor analysis plays an increasingly prominent role in signal and image processing [23], as well as in the social sciences.

In [171] Gil Strang points out that, from a linear algebra standpoint, factor analysis raises some questions. As his example below shows, the representation of $Q$ as $Q = AA^\dagger + D$ is not unique. The matrix $Q$ does not uniquely determine the size of the matrix $A$:

$$Q = \begin{bmatrix} 1 & .74 & .24 & .24 \\ .74 & 1 & .24 & .24 \\ .24 & .24 & 1 & .74 \\ .24 & .24 & .74 & 1 \end{bmatrix} = \begin{bmatrix} .7 & .5 \\ .7 & .5 \\ .7 & -.5 \\ .7 & -.5 \end{bmatrix} \begin{bmatrix} .7 & .7 & .7 & .7 \\ .5 & .5 & -.5 & -.5 \end{bmatrix} + .26I$$

and

$$Q = \begin{bmatrix} .6 & \sqrt{.38} & 0 \\ .6 & \sqrt{.38} & 0 \\ .4 & 0 & \sqrt{.58} \\ .4 & 0 & \sqrt{.58} \end{bmatrix} \begin{bmatrix} .6 & .6 & .4 & .4 \\ \sqrt{.38} & \sqrt{.38} & 0 & 0 \\ 0 & 0 & \sqrt{.58} & \sqrt{.58} \end{bmatrix} + .26I.$$

It is also possible to represent $Q$ with different diagonal components $D$.

# Chapter 56

# Signal Detection and Estimation

In this chapter we consider the problem of deciding whether or not a particular signal is present in the measured data; this is the *detection* problem. The underlying framework for the detection problem is optimal estimation and statistical hypothesis testing [98].

## The general model of signal in additive noise:

The basic model used in detection is that of a signal in additive noise. The complex data vector is $\mathbf{x} = (x_1, x_2, ..., x_N)^T$. We assume that there are two possibilities:

**Case 1: noise only**

$$x_n = z_n,\ n = 1, ..., N,$$

or

**Case 2: signal in noise**

$$x_n = \gamma s_n + z_n,$$

where $\mathbf{z} = (z_1, z_2, ..., z_N)^T$ is a complex vector whose entries $z_n$ are values of random variables that we call *noise*, about which we have only statistical information (that is to say, information about the average behavior), $\mathbf{s} = (s_1, s_2, ..., s_N)^T$ is a complex signal vector that we may known exactly, or at least for which we have a specific parametric model and $\gamma$ is a scalar that may be viewed either as deterministic or random (but unknown, in either case). Unless otherwise stated, we shall assume that $\gamma$ is deterministic.

The *detection problem* is to decide which case we are in, based on some calculation performed on the data $\mathbf{x}$. Since Case 1 can be viewed as a special case of Case 2 in which the value of $\gamma$ is zero, the detection problem is closely related to the problem of estimating $\gamma$, which we discussed in the chapter dealing with the best linear unbiased estimator, the BLUE.

We shall assume throughout that the entries of $\mathbf{z}$ correspond to random variables with means equal to zero. What the variances are and whether or not these random variables are mutually correlated will be discussed below. In all cases we shall assume that this information has been determined previously and is available to us in the form of the covariance matrix $Q = E(\mathbf{z}\mathbf{z}^\dagger)$ of the vector $\mathbf{z}$; the symbol $E$ denotes expected value, so the entries of $Q$ are the quantities $Q_{mn} = E(z_m\overline{z}_n)$. The diagonal entries of $Q$ are $Q_{nn} = \sigma_n^2$, the variance of $z_n$.

Note that we have adopted the common practice of using the same symbols, $z_n$, when speaking about the random variables and about the specific values of these random variables that are present in our data. The context should make it clear to which we are referring.

In case 2 we say that the *signal power* is equal to $|\gamma|^2 \frac{1}{N}\sum_{n=1}^N |s_n|^2 = \frac{1}{N}|\gamma|^2\mathbf{s}^\dagger\mathbf{s}$ and the *noise power* is $\frac{1}{N}\sum_{n=1}^N \sigma_n^2 = \frac{1}{N}tr(Q)$, where $tr(Q)$ is the trace of the matrix Q, that is, the sum of its diagonal terms; therefore the noise power is the average of the variances $\sigma_n^2$. The *input signal-to-noise ratio* ($\text{SNR}_{in}$) is the ratio of the signal power to that of the noise, prior to processing the data; that is,

$$\text{SNR}_{\text{in}} = \frac{1}{N}|\gamma|^2\mathbf{s}^\dagger\mathbf{s}/\frac{1}{N}tr(Q) = |\gamma|^2\mathbf{s}^\dagger\mathbf{s}/tr(Q).$$

## Optimal linear filtering for detection:

In each case to be considered below, our detector will take the form of a linear estimate of $\gamma$; that is, we shall compute the estimate $\hat{\gamma}$ given by

$$\hat{\gamma} = \sum_{n=1}^N \overline{b}_n x_n = \mathbf{b}^\dagger\mathbf{x},$$

where $\mathbf{b} = (b_1, b_2, ..., b_N)^T$ is a vector to be determined. The objective is to use what we know about the situation to select the optimal $\mathbf{b}$, which will depend on $\mathbf{s}$ and $Q$.

For any given vector $\mathbf{b}$, the quantity

$$\hat{\gamma} = \mathbf{b}^\dagger\mathbf{x} = \gamma\mathbf{b}^\dagger\mathbf{s} + \mathbf{b}^\dagger\mathbf{z}$$

is a random variable whose mean value is equal to $\gamma\mathbf{b}^\dagger\mathbf{s}$ and whose variance is

$$var(\hat{\gamma}) = E(|\mathbf{b}^\dagger\mathbf{z}|^2) = E(\mathbf{b}^\dagger\mathbf{z}\mathbf{z}^\dagger\mathbf{b}) = \mathbf{b}^\dagger E(\mathbf{z}\mathbf{z}^\dagger)\mathbf{b} = \mathbf{b}^\dagger Q\mathbf{b}.$$

Therefore, the *output signal-to-noise ratio* ($\text{SNR}_{out}$) is defined to be

$$\text{SNR}_{\text{out}} = |\gamma \mathbf{b}^\dagger \mathbf{s}|^2 / \mathbf{b}^\dagger Q \mathbf{b}.$$

The advantage we obtain from processing the data is called the *gain* associated with $\mathbf{b}$ and is defined to be the ratio of the $\text{SNR}_{out}$ to $\text{SNR}_{in}$; that is

$$\text{gain}(\mathbf{b}) = \frac{|\gamma \mathbf{b}^\dagger \mathbf{s}|^2/(\mathbf{b}^\dagger Q \mathbf{b})}{|\gamma|^2(\mathbf{s}^\dagger \mathbf{s})/tr(Q)} = \frac{|\mathbf{b}^\dagger \mathbf{s}|^2\, tr(Q)}{(\mathbf{b}^\dagger Q \mathbf{b})(\mathbf{s}^\dagger \mathbf{s})}.$$

The best $\mathbf{b}$ to use will be the one for which gain($\mathbf{b}$) is the largest. So, ignoring the terms in the gain formula that do not involve $\mathbf{b}$, we see that the problem becomes *maximize* $\frac{|\mathbf{b}^\dagger \mathbf{s}|^2}{\mathbf{b}^\dagger Q \mathbf{b}}$, *for fixed signal vector* $\mathbf{s}$ *and fixed noise covariance matrix* $Q$.

The Cauchy inequality plays a major role in optimal filtering and detection:

**Cauchy's inequality:** for any vectors $\mathbf{a}$ and $\mathbf{b}$ we have

$$|\mathbf{a}^\dagger \mathbf{b}|^2 \leq (\mathbf{a}^\dagger \mathbf{a})(\mathbf{b}^\dagger \mathbf{b}),$$

with equality if and only if $\mathbf{a}$ is proportional to $\mathbf{b}$, that is, there is a scalar $\beta$ such that $\mathbf{b} = \beta \mathbf{a}$.

**Exercise 1:** Use Cauchy's inequality to show that, for any fixed vector $\mathbf{a}$, the choice $\mathbf{b} = \beta \mathbf{a}$ maximizes the quantity $|\mathbf{b}^\dagger \mathbf{a}|^2/\mathbf{b}^\dagger \mathbf{b}$, for any constant $\beta$.

**Exercise 2:** Use the definition of the covariance matrix $Q$ to show that $Q$ is Hermitian and that, for any vector $\mathbf{y}$, $\mathbf{y}^\dagger Q \mathbf{y} \geq 0$. Therefore $Q$ is a nonnegative definite matrix and, using its eigenvector decomposition, can be written as $Q = CC^\dagger$, for some invertible square matrix $C$.

**Exercise 3:** Consider now the problem of maximizing $|\mathbf{b}^\dagger \mathbf{s}|^2/\mathbf{b}^\dagger Q \mathbf{b}$. Using the two previous exercises, show that the solution is $\mathbf{b} = \beta Q^{-1}\mathbf{s}$, for some arbitrary constant $\beta$.

We can now use the results of these exercises to continue our discussion. We choose the constant $\beta = 1/(\mathbf{s}^\dagger Q^{-1}\mathbf{s})$ so that the optimal $\mathbf{b}$ has $\mathbf{b}^\dagger \mathbf{s} = 1$; that is, the **optimal filter** $\mathbf{b}$ is

$$\mathbf{b} = (1/(\mathbf{s}^\dagger Q^{-1}\mathbf{s}))Q^{-1}\mathbf{s}$$

and the *optimal estimate* of $\gamma$ is

$$\hat{\gamma} = \mathbf{b}^\dagger \mathbf{x} = (1/(\mathbf{s}^\dagger Q^{-1}\mathbf{s}))(\mathbf{s}^\dagger Q^{-1}\mathbf{x}).$$

The random variable $\hat{\gamma}$ has mean equal to $\gamma \mathbf{b}^\dagger \mathbf{s} = \gamma$ and variance equal to $1/(\mathbf{s}^\dagger Q^{-1}\mathbf{s})$. Therefore, the output signal power is $|\gamma|^2$, the output noise power is $1/(\mathbf{s}^\dagger Q^{-1}\mathbf{s})$ and so the *output signal-to-noise ratio* ($\text{SNR}_{out}$) is

$$\text{SNR}_{\text{out}} = |\gamma|^2(\mathbf{s}^\dagger Q^{-1}\mathbf{s}).$$

The gain associated with the optimal vector $\mathbf{b}$ is then

$$\text{maximum gain} = \frac{(\mathbf{s}^\dagger Q^{-1}\mathbf{s})\, tr(Q)}{(\mathbf{s}^\dagger \mathbf{s})}.$$

The calculation of the vector $C^{-1}\mathbf{x}$ is sometimes called *prewhitening* since $C^{-1}\mathbf{x} = \gamma C^{-1}\mathbf{s} + C^{-1}\mathbf{z}$ and the new noise vector, $C^{-1}\mathbf{z}$, has the identity matrix for its covariance matrix. The new signal vector is $C^{-1}\mathbf{s}$. The filtering operation that gives $\hat{\gamma} = \mathbf{b}^\dagger \mathbf{x}$ can be written as

$$\hat{\gamma} = (1/(\mathbf{s}^\dagger Q^{-1}\mathbf{s}))(C^{-1}\mathbf{s})^\dagger C^{-1}\mathbf{x};$$

the term $(C^{-1}\mathbf{s})^\dagger C^{-1}\mathbf{x}$ is described by saying that we *prewhiten, then do a matched filter*. Now we consider some special cases of noise.

## The case of white noise:

We say that the noise is *white noise* if the covariance matrix is $Q = \sigma^2 I$, where $I$ denotes the identity matrix that is one on the main diagonal and zero elsewhere and $\sigma > 0$ is the common standard deviation of the $z_n$. This means that the $z_n$ are mutually uncorrelated (independent, in the Gaussian case) and share a common variance.

In this case the optimal vector $\mathbf{b}$ is $\mathbf{b} = \frac{1}{(\mathbf{s}^\dagger \mathbf{s})}\mathbf{s}$ and the gain is $N$. Notice that $\hat{\gamma}$ now involves only a matched filter. We consider now some special cases of the signal vectors $\mathbf{s}$.

**Constant signal:** Suppose that the vector $\mathbf{s}$ is constant, that is, $\mathbf{s} = \mathbf{1} = (1, 1, ..., 1)^T$. Then we have

$$\hat{\gamma} = \frac{1}{N}\sum_{n=1}^{N} x_n.$$

This is the same result we found in our discussion of the BLUE, when we estimated the mean value and the noise was white.

**Sinusoidal signal - known frequency:** Suppose

$$\mathbf{s} = \mathbf{e}(\omega_0) = (\exp(-i\omega_0), \exp(-2i\omega_0), ..., \exp(-Ni\omega_0))^T,$$

where $\omega_0$ denotes a known frequency in $[-\pi, \pi)$. Then $\mathbf{b} = \frac{1}{N}\mathbf{e}(\omega_0)$ and

$$\hat{\gamma} = \frac{1}{N}\sum_{n=1}^{N} x_n \exp(in\omega_0);$$

so we see yet another occurrence of the DFT.

**Sinusoidal signal - unknown frequency:** If we do not know the value of the signal frequency $\omega_0$ a reasonable thing to do is to calculate the $\hat{\gamma}$ for each (actually, finitely many) of the possible frequencies within $[-\pi, \pi)$ and base the detection decision on the largest value; that is, we calculate the DFT as a function of the variable $\omega$. If there is only a single $\omega_0$ for which there is a sinusoidal signal present in the data, the values of $\hat{\gamma}$ obtained at frequencies other than $\omega_0$ provide estimates of the noise power $\sigma^2$, against which the value of $\hat{\gamma}$ for $\omega_0$ can be compared.

## The case of correlated noise:

We say that the noise is *correlated* if the covariance matrix is $Q$ is not a multiple of the identity matrix. This means either that the $z_n$ are mutually correlated (dependent, in the Gaussian case) or that they are uncorrelated, but have different variances.

In this case, as we saw above, the optimal vector $\mathbf{b}$ is

$$\mathbf{b} = \frac{1}{(\mathbf{s}^\dagger Q^{-1}\mathbf{s})} Q^{-1}\mathbf{s}$$

and the gain is

$$\text{maximum gain} = \frac{(\mathbf{s}^\dagger Q^{-1}\mathbf{s})\, tr(Q)}{(\mathbf{s}^\dagger \mathbf{s})}.$$

How large or small the gain is depends on how the signal vector $\mathbf{s}$ relates to the matrix $Q$.

For sinusoidal signals, the quantity $\mathbf{s}^\dagger\mathbf{s}$ is the same, for all values of the parameter $\omega$; this is not always the case, however. In passive detection of sources in acoustic array processing, for example, the signal vectors arise from models of the acoustic medium involved. For far-field sources in an (acoustically) isotropic deep ocean, planewave models for $\mathbf{s}$ will have the property that $\mathbf{s}^\dagger\mathbf{s}$ does not change with source location. However, for near-field or shallow-water environments, this is usually no longer the case.

It follows from an earlier exercise that the quantity $\frac{\mathbf{s}^\dagger Q^{-1}\mathbf{s}}{\mathbf{s}^\dagger\mathbf{s}}$ achieves its maximum value when $\mathbf{s}$ is an eigenvector of $Q$ associated with its smallest eigenvalue, $\lambda_N$; in this case, we are saying that the signal vector does not look very much like a typical noise vector. The maximum gain is then

$\lambda_N^{-1} tr(Q)$. Since $tr(Q)$ equals the sum of its eigenvalues, multiplying by $tr(Q)$ serves to normalize the gain, so that we cannot get larger gain simply by having all the eigenvalues of $Q$ small.

On the other hand, if $\mathbf{s}$ should be an eigenvector of $Q$ associated with its largest eigenvalue, say $\lambda_1$, then the maximum gain is $\lambda_1^{-1} tr(Q)$. If the noise is signal-like, that is, has one dominant eigenvalue, then $tr(Q)$ is approximately $\lambda_1$ and the maximum gain is around one, so we have lost the maximum gain of $N$ we were able to get in the white noise case. This makes sense, in that it says that we cannot significantly improve our ability to discriminate between signal and noise by taking more samples, if the signal and noise are very similar.

**Constant signal with unequal-variance uncorrelated noise:** Suppose that the vector $\mathbf{s}$ is constant, that is, $\mathbf{s} = \mathbf{1} = (1, 1, ..., 1)^T$. Suppose also that the noise covariance matrix is $Q = \text{diag}\{\sigma_1, ..., \sigma_N\}$.

In this case the optimal vector $\mathbf{b}$ has entries

$$b_m = \frac{1}{(\sum_{n=1}^N \sigma_n^{-1})} \sigma_m^{-1},$$

for $m = 1, ..., N$, and we have

$$\hat{\gamma} = \frac{1}{(\sum_{n=1}^N \sigma_n^{-1})} \sum_{m=1}^N \sigma_m^{-1} x_m.$$

This is the BLUE estimate of $\gamma$ in this case.

**Sinusoidal signal - known frequency, in correlated noise:** Suppose

$$\mathbf{s} = \mathbf{e}(\omega_0) = (\exp(-i\omega_0), \exp(-2i\omega_0), ..., \exp(-Ni\omega_0))^T,$$

where $\omega_0$ denotes a known frequency in $[-\pi, \pi)$. In this case the optimal vector $\mathbf{b}$ is

$$\mathbf{b} = \frac{1}{\mathbf{e}(\omega_0)^\dagger Q^{-1} \mathbf{e}(\omega_0)} Q^{-1} \mathbf{e}(\omega_0)$$

and the gain is

$$\text{maximum gain} = \frac{1}{N} [\mathbf{e}(\omega_0)^\dagger Q^{-1} \mathbf{e}(\omega_0)] tr(Q).$$

How large or small the gain is depends on the quantity $q(\omega_0)$, where

$$q(\omega) = \mathbf{e}(\omega)^\dagger Q^{-1} \mathbf{e}(\omega).$$

The function $1/q(\omega)$ can be viewed as a sort of noise power spectrum, describing how the noise power appears when decomposed over the various

frequencies in $[-\pi, \pi)$. The maximum gain will be large if this *noise power spectrum* is relatively small near $\omega = \omega_0$; however, when the noise is similar to the signal, that is, when the noise power spectrum is relatively large near $\omega = \omega_0$, the maximum gain can be small. In this case the noise power spectrum plays a role analogous to that played by the eigenvalues of $Q$ earlier.

To see more clearly why it is that the function $1/q(\omega)$ can be viewed as a sort of noise power spectrum, consider what we get when we apply the optimal filter associated with $\omega$ to data containing only noise. The average output should tell us how much power there is in the component of the noise that resembles $\mathbf{e}(\omega)$; this is essentially what is meant by a noise power spectrum. The result is $\mathbf{b}^\dagger \mathbf{z} = (1/q(\omega))\mathbf{e}(\omega)^\dagger Q^{-1}\mathbf{z}$. The expected value of $|\mathbf{b}^\dagger \mathbf{z}|^2$ is then $1/q(\omega)$.

**Sinusoidal signal - unknown frequency:** Again, if we do not know the value of the signal frequency $\omega_0$ a reasonable thing to do is to calculate the $\hat{\gamma}$ for each (actually, finitely many) of the possible frequencies within $[-\pi, \pi)$ and base the detection decision on the largest value. For each $\omega$ the corresponding value of $\hat{\gamma}$ is

$$\hat{\gamma}(\omega) = [1/(\mathbf{e}(\omega)^\dagger Q^{-1}\mathbf{e}(\omega))] \sum_{n=1}^{N} a_n \exp(in\omega),$$

where $\mathbf{a} = (a_1, a_2, ..., a_N)^T$ satisfies the linear system $Q\mathbf{a} = \mathbf{x}$ or $\mathbf{a} = Q^{-1}\mathbf{x}$. It is interesting to note the similarity between this estimation procedure and the PDFT discussed in earlier notes; to see the connection view $[1/(\mathbf{e}(\omega)^\dagger Q^{-1}\mathbf{e}(\omega))]$ in the role of $P(\omega)$ and $Q$ its corresponding matrix of Fourier transform values. The analogy breaks down when we notice that $Q$ need not be Toeplitz, as in the PDFT case; however, the similarity is intriguing.

# Chapter 57

# Random Signal Detection

We consider now the detection and estimation problem for the case in which the signal components have random aspects as well.

## Random amplitude sinusoid in noise:

A somewhat more general model for sinusoids in additive noise is the following. The complex data vector is $\mathbf{x} = (x_1, x_2, ..., x_N)^T$. We assume that there are two possibilities:

**Case 1: noise only**

$$x_n = z_n,\ n = 1, ..., N,$$

or

**Case 2: signal in noise**

$$x_n = \gamma s_n + z_n,$$

where $\gamma = |\gamma| \exp(i\theta)$ is an unknown value of a complex random variable whose magnitude $|\gamma|$ and phase $\theta$ are mutually independent and independent of the noise. In this case the mean value of $\gamma$ can be zero, if $\theta$ is distributed uniformly over $[-\pi, \pi)$. The presence of a nonzero signal component is detected through the increase in the variance, not through a nonzero mean value, as above. The calculations are basically the same as the earlier ones and we shall not consider this case further.

## Multiple independent sinusoids in noise:

We mention briefly the case in which there may be more than one sinusoid present. For this case a random model is typically used, in which the

magnitudes and phases of the different sinusoids are taken to be mutually independent. Statistical hypothesis testing theory tells us that we should detect in two steps now:

**1:** perform a maximum likelihood estimation of the number and location (in frequency space) of the sinusoidal components; then

**2:** use the optimal linear filtering to estimate their respective coefficients, the $\gamma$'s.

The first step is computationally intractible and various suboptimal, but computationally efficient, alternatives are commonly used. These alternative methods can involve the eigenvector- or singular value decomposition of certain matrices formed from the data vector $\mathbf{x}$, and so are nonlinear procedures. How well we can detect two or more separate signals will, of course, depend on how distinct their $\mathbf{s}$ vectors are, how distinct each is from the noise, how accurate our knowledge of the noise correlation matrix $Q$ is, how accurate our model of the $\mathbf{s}$ is and on the value of $N$; this is the *resolution problem*. Our ability to resolve will also depend on the accuracy of the measurements, therefore on the hardware used to collect the measurements.

## Data-adaptive high resolution methods:

In all of the discussion so far, we have assumed that the noise correlation matrix $Q$ was available to use in forming the optimal filter $\mathbf{b}$. The $Q$ may depend on data previously obtained or may simply be the result of a model chosen to describe the physical situation. In some applications, such as sonar array processing, the $Q$ may vary from minute to minute; it would be helpful if we could obtain as good an estimate as possible of the current value of $Q$, but this would require measurements, at the present moment, of the noise without the embedded signal, which is impossible. One approach, due to Capon [56], is a *data-adaptive high resolution detection*; it has been used in the case in which there are potentially more than one signal present, to achieve higher resolution than that obtainable by the methods we have discussed so far.

**Data-adaptive high resolution methods- sinusoidal signals**

The idea behind these methods is to use the data vector $\mathbf{x}$ to estimate the noise correlation matrix. Since the vector $\mathbf{x}$ may also contain signals, it would seem that we would be lumping signals in with noise and designing a filter $\mathbf{b}$ to suppress everything. The constraint $\mathbf{b}^\dagger\mathbf{e}(\omega) = 1$ saves us, however.

Suppose that there are two signals present: then the vector $\mathbf{x}$ has components

$$x_n = \gamma_1 \exp(-in\omega_1) + \gamma_2 \exp(-in\omega_2) + z_n,$$

for $n = 1, ..., N$. When we are trying to detect $\mathbf{e}(\omega_1)$ it is fine if the $\mathbf{e}(\omega_2)$ component is viewed as noise, and vice versa. High resolution depends on what the output of our filter is when we look at a frequency $\omega$ that is between $\omega_1$ and $\omega_2$; now it is advantageous that the signal components are lumped in with the noise.

To obtain a substitute for $Q$ we partition the $N$ by 1 data vector $\mathbf{x}$ into $K$ smaller $M$ by 1 vectors, denoted $\mathbf{y}^k$, for $k = 1, ..., K$ and $N = MK$. Specifically, we let

$$y_m^k = x_{(k-1)M+m},\ m = 1, ..., M,$$

for $k = 1, 2, ..., K$. We then define the $M$ by $M$ matrix $R$ as follows:

$$R_{jm} = \frac{1}{K} \sum_{k=1}^{K} y_j^k \, \overline{y}_m^k,$$

for $j$, $m = 1, 2, ..., M$. The matrix $R$ is then Hermitian and nonnegative definite. The signal components involving $\mathbf{e}(\omega_1)$ and $\mathbf{e}(\omega_2)$ are transformed into shorter components of the form

$$\tilde{\mathbf{e}}(\omega) = (\exp(-i\omega), ..., \exp(-iM\omega))^T.$$

To obtain our data-adaptive estimate of the $\gamma$ of the potential signal component $\tilde{\mathbf{e}}(\omega)$ we apply the optimal filtering, as before, but to each of the vectors $\mathbf{y}^k$ separately, using $R$ instead of $Q$ and using $\tilde{\mathbf{e}}(\omega)$ instead of $\mathbf{e}(\omega)$. We then average the squared magnitudes of the resulting estimates over $k = 1, ..., K$, to obtain our estimate of the $|\gamma|^2$ associated with $\omega$.

**Capon's data-adaptive estimator:**

$$|\hat{\gamma}(\omega)|^2 = 1/(\tilde{\mathbf{e}}(\omega)^\dagger R^{-1}(\tilde{\mathbf{e}}(\omega)).$$

**Exercise 1:** (or better, Research Project 1.) What is going on here? Why is this method 'high resolution'? What does $R$ look like? What are its eigenvalues and eigenvectors? Can we apply it to signals other than sinusoids? Is it important that the signal coefficients (the $\gamma$'s) be random? What can go wrong? How can it be fixed?

# Chapter 58

# Parameter Estimation in Reconstruction

In its most general formulation our problem is simple. We have a vector of measured data $\mathbf{y} = (y_1, ..., y_I)^T$. Related to the data in some way is a vector $\mathbf{x} = (x_1, ..., x_J)^T$ whose entries are parameters we wish to determine. To solve the problem we need to describe the relationship between $\mathbf{y}$ and $\mathbf{x}$ and then use this description to solve for $\mathbf{x}$. As always, the devil is in the details.

The problem as stated is so general as to include problems that lie outside our main area of interest, such as drawing inferences from census data. While we do not need to exclude such problems, to which many of the techniques discussed in this book indeed apply, we shall focus here on applications in which the relationship between data and parameters involves a physical model describing some form of remote sensing or imaging. The vector $\mathbf{x}$ will often represent a vectorization of a discretized two-dimensional distribution; that is, $\mathbf{x}$ will be a vectorized image. The data vector $\mathbf{y}$ in such cases may also be a vectorized image, such as a blurred version of $\mathbf{x}$, or may simply be measurements, such as projections, related to $\mathbf{x}$. On occasion we shall formulate our problem in terms of finding a continuous distribution, as in our discussion of the Radon transform in tomography. But for the most part it is sufficient to assume that a discretization has taken place and that the unknowns are entries of a finite vector $\mathbf{x}$.

In all of the applications of interest the data is noisy and the relationship between the data and the parameters imperfectly known. Even in the absence of these errors the measurements may not be sufficient to specify a unique solution. There will always be a trade-off between the complexity of the description of the relationship and the ease of solving for the desired $\mathbf{x}$.

Because the measurements involve noise the relationship of the data to the parameters must include randomness. We shall find it useful to consider our problem as statistical parameter estimation. While this choice may seem overly restrictive it is general enough for our purposes and is, in fact, a fairly popular choice in the literature of signal processing, image reconstruction and remote sensing.

**Statistical parameter estimation:** Suppose that $\mathbf{Y}$ is a random vector whose probability density function (pdf) $f(\mathbf{y};\mathbf{x})$ is a function of the vector variable $\mathbf{y}$ and is a member of a family of pdf parametrized by the vector variable $\mathbf{x}$. Our data is one instance of $\mathbf{Y}$, that is, one particular value of the variable $\mathbf{y}$, which we also denote by $\mathbf{y}$. We want to estimate the correct value of the variable $\mathbf{x}$, which we shall also denote by $\mathbf{x}$. This notation is standard and the dual use of the symbols $\mathbf{y}$ and $\mathbf{x}$ should not cause confusion. Given the particular $\mathbf{y}$ we can estimate the correct $\mathbf{x}$ by viewing $f(\mathbf{y};\mathbf{x})$ as a function of the second variable, with the first variable held fixed. This function of the parameters only is called the *likelihood function*. A *maximum likelihood* (ML) estimate of the parameter vector $\mathbf{x}$ is any value of the second variable for which the function is maximized. We consider several examples.

**Example 1: Estimating a Gaussian mean:** Let $Y_1, ..., Y_I$ be $I$ independent Gaussian (or normal) random variables with known variance $\sigma^2 = 1$ and unknown common mean $\mu$. Let $\mathbf{Y} = (Y_1, ..., Y_I)^T$. The parameter $x$ we wish to estimate is the mean $x = \mu$. Then the random vector $\mathbf{Y}$ has the pdf

$$f(\mathbf{y};x) = (2\pi)^{-I/2} \exp(-\frac{1}{2} \sum\nolimits_{i=1}^{I} (y_i - x)^2).$$

Holding $\mathbf{y}$ fixed and maximizing over $x$ is equivalent to minimizing

$$\sum\nolimits_{i=1}^{I} (y_i - x)^2$$

as a function of $x$. The ML estimate is the arithmetic mean of the data,

$$x_{ML} = \frac{1}{I} \sum_{i=1}^{I} y_i.$$

Notice that $E(\mathbf{Y})$, the expected value of $\mathbf{Y}$, is the vector $\mathbf{x}$ all of whose entries are $x = \mu$. The ML estimate is the least squares solution of the overdetermined system of equations $\mathbf{y} = E(\mathbf{Y})$, that is,

$$y_i = x$$

for $i = 1, ..., I$.

The least squares solution of a system of equations $A\mathbf{x} = \mathbf{b}$ is the vector that minimizes the Euclidean distance between $A\mathbf{x}$ and $\mathbf{b}$; that is, it minimizes the Euclidean norm of their difference, $||A\mathbf{x} - \mathbf{b}||^2$, where, for any two vectors $\mathbf{a}$ and $\mathbf{b}$ we define

$$||\mathbf{a} - \mathbf{b}||^2 = \sum_{i=1}^{I} (a_i - b_i)^2.$$

As we shall see in the next example, another important measure of distance is the *Kullback-Leibler* (KL) distance between two nonnegative vectors $\mathbf{c}$ and $\mathbf{d}$, given by

$$KL(\mathbf{c}, \mathbf{d}) = \sum_{i=1}^{I} c_i \log(c_i/d_i) + d_i - c_i.$$

**Example 2: Estimating a Poisson mean** Let $Y_1, ..., Y_I$ be $I$ independent Poisson random variables with unknown common mean $\lambda$, which is the parameter $x$ we wish to estimate. Let $\mathbf{Y} = (Y_1, ..., Y_I)^T$. Then the probability function of $\mathbf{Y}$ is

$$f(\mathbf{y}; x) = \prod_{i=1}^{I} \exp(-x) x^{y_i}/(y_i)!.$$

Holding $\mathbf{y}$ fixed and maximizing this likelihood function over positive values of $x$ is equivalent to minimizing the Kullback-Leibler distance between the nonnegative vector $\mathbf{y}$ and the vector $\mathbf{x}$ whose entries are all equal to $x$, given by

$$KL(\mathbf{y}, \mathbf{x}) = \sum_{i=1}^{I} y_i \log(y_i/x) + x - y_i.$$

The ML estimator is easily seen to be the arithmetic mean of the data,

$$x_{ML} = \frac{1}{I} \sum_{i=1}^{I} y_i.$$

The vector $\mathbf{x}$ is again $E(\mathbf{Y})$, so the ML estimate is once again obtained by finding an approximate solution of the overdetermined system of equations $\mathbf{y} = E(\mathbf{Y})$. In the previous example the approximation was in the least squares sense, whereas here it is in the minimum KL sense; the ML estimate is the arithmetic mean in both cases because the parameter to be estimated is one-dimensional.

**Example 3: Estimating a uniform mean** Suppose now that $Y_1, ..., Y_I$ are independent random variables uniformly distributed over the interval

$[0, 2x]$. The parameter to be determined in their common mean, $x$. The random vector $\mathbf{Y} = (Y_1, ..., Y_I)^T$ has the pdf

$$f(\mathbf{y}; x) = x^{-I}, \text{for } 2x \geq m,$$

$$f(\mathbf{y}; x) = 0\,, \text{otherwise}\,,$$

where $m$ is the maximum of the $y_i$. For fixed vector $\mathbf{y}$ the ML estimate of $x$ is $m/2$. The expected value of $\mathbf{Y}$ is $E(\mathbf{Y}) = \mathbf{x}$ whose entries are all equal to $x$. In this case the ML estimator is not obtained by finding an approximate solution to the overdetermined system $\mathbf{y} = E(\mathbf{Y})$.

Since we can always write

$$\mathbf{y} = E(\mathbf{Y}) + (\mathbf{y} - E(\mathbf{Y}))$$

we can model $\mathbf{y}$ as the sum of $E(\mathbf{Y})$ and mean-zero error or noise. Since $f(\mathbf{y}; \mathbf{x})$ depends on $\mathbf{x}$ so does $E(\mathbf{Y})$. Therefore it makes some sense to consider estimating our parameter vector $\mathbf{x}$ using an approximate solution for the system of equations

$$\mathbf{y} = E(\mathbf{Y}).$$

As the first two examples (as well as many others) illustrate, this is what the ML approach often amounts to, while the third example shows that this is not always the case, however. Still to be determined, though, is the metric with respect to which the approximation is to be performed. As the Gaussian and Poisson examples showed, the ML formalism can provide that metric. In those overly simple cases it did not seem to matter which metric we used, but it does matter.

**Example 4: Image restoration** A standard model for image restoration is the following:

$$\mathbf{y} = A\mathbf{x} + \mathbf{z},$$

where $\mathbf{y}$ is the blurred image, $A$ is an $I$ by $J$ matrix describing the linear imaging system, $\mathbf{x}$ is the desired vectorized restored image and $\mathbf{z}$ is (possibly correlated) mean-zero additive Gaussian noise. The noise covariance matrix is $Q = E(\mathbf{z}\mathbf{z}^T)$. Then $E(\mathbf{Y}) = A\mathbf{x}$ and the pdf is

$$f(\mathbf{y}; \mathbf{x}) = c \exp(-(\mathbf{y} - A\mathbf{x})^T Q^{-1} (\mathbf{y} - A\mathbf{x})),$$

where $c$ is a constant that does not involve $\mathbf{x}$. Holding $\mathbf{y}$ fixed and maximizing $f(\mathbf{y}; \mathbf{x})$ with respect to $\mathbf{x}$ is equivalent to minimizing

$$(\mathbf{y} - A\mathbf{x})^T Q^{-1} (\mathbf{y} - A\mathbf{x}).$$

Therefore the ML solution is obtained by finding a weighted least squares approximate solution of the overdetermined linear system $\mathbf{y} = E(\mathbf{Y})$, with

the weights coming from the matrix $Q^{-1}$. When the noise terms are uncorrelated and have the same variance this reduces to the least squares solution.

**Example 5: Poisson mixtures** The model of a Poisson mixture is commonly used in emission tomography and elsewhere. Let $P$ be an $I$ by $J$ matrix with nonnegative entries and let $\mathbf{x} = (x_1, ..., x_J)^T$ be a vector of nonnegative parameters. Let $Y_1, ..., Y_I$ be independent Poisson random variables with positive means

$$E(Y_i) = \sum_{j=1}^{J} P_{ij} x_j = (P\mathbf{x})_i.$$

The probability function for the random vector $\mathbf{Y}$ is then

$$f(\mathbf{y}; \mathbf{x}) = c \prod_{i=1}^{I} \exp(-(P\mathbf{x})_i)((P\mathbf{x})_i)^{y_i},$$

where $c$ is a constant not involving $\mathbf{x}$. Maximizing this function of $\mathbf{x}$ for fixed $\mathbf{y}$ is equivalent to minimizing the KL distance $KL(\mathbf{y}, P\mathbf{x})$ over nonnegative $\mathbf{x}$. The expected value of the random vector $\mathbf{Y}$ is $E(\mathbf{Y}) = P\mathbf{x}$ and once again we see that the ML estimate is a nonnegative approximate solution of the system of (linear) equations $\mathbf{y} = E(\mathbf{Y})$, with the approximation in the KL sense. The system $\mathbf{y} = P\mathbf{x}$ may not be overdetermined; there may even be exact solutions. But we require in addition that $\mathbf{x} \geq 0$ and there need not be a nonnegative solution to $\mathbf{y} = P\mathbf{x}$. We see from this example that constrained optimization plays a role in solving our problems.

In the previous two examples the expected value $E(\mathbf{Y})$ was linear in the vector $\mathbf{x}$. This is a convenient and commonly employed model but does not always apply, as we shall see in our discussion of transmission tomography.

The ML approach is not always the best approach. As we have seen, the ML estimate is often found by solving, at least approximately, the system of equations $\mathbf{y} = E(\mathbf{Y})$. Since noise is always present, this system of equations is rarely a correct statement of the situation. It is possible to overfit the mean to the noisy data, in which case the resulting $\mathbf{x}$ can be useless. In such cases Bayesian methods and maximum *a posteriori* estimation, as well as other forms of regularization and penalty function techniques, can help. Other approaches involve stopping iterative algorithms prior to convergence.

In most applications the data is limited and it is helpful to include prior information about the parameter vector $\mathbf{x}$ to be estimated. In the Poisson mixture problem above the vector $\mathbf{x}$ must have nonnegative entries. In certain applications, such as transmission tomography, we might have upper bounds on suitable values of the entries of $\mathbf{x}$.

From a mathematical standpoint we are interested in the convergence of iterative algorithms, while in many applications we want usable estimates in a reasonable amount of time, often obtained by running an iterative algorithm for only a few iterations. Algorithms designed to minimize the same cost function can behave quite differently during the early iterations. Iterative algorithms, such as block-iterative or incremental methods, that can provide decent answers quickly will be important.

Formulating the problem as one of statistical parameter estimation and then applying likelihood maximization is by no means the end of the story. In the Poisson mixture problem we are told to minimize the KL distance $KL(\mathbf{y}, P\mathbf{x})$ with respect to $\mathbf{x} \geq 0$, but we are not told how to do this. Even in the linear image restoration example we still need an algorithm for finding the weighted least squares solution of the (possibly) overdetermined system $\mathbf{y} = A\mathbf{x}$. If there happen to be multiple exact solutions we still would need a criterion (and an algorithm) for selecting one out the many possibilities. Keeping in mind that these systems involve thousands of equations and thousands of unknowns in most cases, we see that practical considerations, such as storage and computation time, will be important. With few exceptions the algorithms we shall consider here are iterative ones.

The main problems of image reconstruction are deriving an accurate model for the data collection, determining appropriate cost functions to be minimized and obtaining suitable algorithms for this minimization. There are, of course, general methods for minimization, such as steepest descent methods, that can be applied to any problem. Because many of the minimization problems encountered here will involve restrictions on the desired solution, such as nonnegativity, we find that methods tailored to the specific problem are often preferred.

In developing algorithms it helps to have some guiding principles or paradigms. One such paradigm is *fixed point iteration*. Suppose that we wish to minimize the real-valued cost function $F(\mathbf{x})$. In the absence of constraints this usually means that we want its gradient to vanish, that is, we want $f(\mathbf{x}) = \nabla F(\mathbf{x}) = 0$. Equivalently, we want an $\mathbf{x}$ which, for any invertible matrix $G$, satisfies

$$\mathbf{x} = \mathbf{x} + G^{-1} f(\mathbf{x});$$

that is, we want a fixed point of the operator

$$T(\mathbf{x}) = \mathbf{x} + G^{-1} f(\mathbf{x}).$$

An obvious way to find fixed points is to compute the sequence of iterates $\{\mathbf{x}^{k+1} = T(\mathbf{x}^k)\}$. The function $f(\mathbf{x})$ is determined, but we are free to select the matrix $G$. The objective is to find a $G$ that is easily inverted and for which the iteration converges.

A second paradigm for image reconstruction is *convex feasibility*. The parameter vector $\mathbf{x}$ to be estimated often is known to satisfy certain constraints which can be imposed by requiring that $\mathbf{x}$ be a member of each of several closed convex sets, $C_m$, $m = 1, ..., M$. Finding a member of the intersection of convex sets is called the *convex feasibility problem* (CFP). The *projection onto convex sets* (POCS) method is one way to derive an algorithm to solve the CFP. Several of the algorithms we shall consider later are best derived using *alternating minimization* methods, which is POCS with $M = 2$. These algorithms are also fixed point iteration schemes, combining the two paradigms. Sometimes the algorithms are designed so that the constraints are satisfied not only by the limit vector, but by each of the iterates $\mathbf{x}^k$; these methods are *interior point algorithms*.

# Chapter 59

# Emission Tomography

In positron emission tomography (PET) and single photon emission tomography (SPECT) the patient swallows, inhales or is injected with chemicals to which radioactive material has been chemically attached. The chemicals are designed to accumulate in that specific region of the body we wish to image. For example, we may be looking for tumors in the abdomen, weakness in the heart wall or evidence of brain activity in a selected region. The patient is placed on a table surrounded by detectors that count the number of emitted photons. On the basis of where the various counts were obtained, we wish to determine the concentration of radioactivity at various locations throughout the region of interest within the patient.

In SPECT the radionuclide emits single photons, which then travel through the body of the patient and, in some fraction of the cases, are detected. Detections in SPECT correspond to individual sensor locations outside the body. The data is SPECT are the photon counts at each of the finitely many detector locations.

In PET the situation is different. The radionuclide emits individual positrons, which travel, on average, between 4 mm and 2.5 cm (depending on their kinetic energy) before encountering an electron. The resulting annihilation releases two gamma-ray photons that then proceed in essentially opposite directions. Detection in the PET case means the recording of two photons at nearly the same time at two different detectors. The locations of these two detectors then provide the end points of the line segment passing, more or less, through the site of the original positron emission. Therefore each possible pair of detectors determines a *line of response* (LOR). Because there are so many such LOR the odds are good that no LOR is recorded more than once and most are never recorded. When a LOR is recorded it is assumed that a positron was emitted somewhere along that line. The PET data consists of the list of LOR that are recorded. Because the two photons detected at either end of the LOR are not detected at exactly the

same time the time difference can be used in *time of flight* PET to further localize the site of the emission to a smaller segment of perhaps 8 cm in length.

In what follows we use the term *detector* in whichever sense is appropriate for the modality under discussion. We begin by *discretizing* the problem; that is, we imagine the region of interest within the patient to consist of finitely many tiny squares, called *pixels* for two dimensional processing or cubes, called *voxels* for three dimensional processing. In what follows we shall not distinguish the two cases, but as a linguistic shorthand, we shall refer to 'pixels' indexed by $j = 1, ..., J$. The detectors are indexed by $i = 1, ..., I$, the count obtained at detector $i$ is denoted $y_i$ and the vector $\mathbf{y} = (y_1, ..., y_I)^T$ is our data. In practice, for the fully 3D case, $I$ and $J$ can be several hundred thousand.

We imagine that each pixel $j$ has its own level of concentration of radioactivity and these concentration levels are what we want to determine. Proportional to these concentration levels are the average rates of emission of photons; the average rate for $j$ we denote by $x_j$. The goal is to determine the vector $\mathbf{x} = (x_1, ..., x_J)^T$ from $\mathbf{y}$.

To achieve our goal we must construct a model that relates $\mathbf{y}$ to $\mathbf{x}$. The standard way to do this is to adopt the model of *independent Poisson emitters*. For $i = 1, ..., I$ and $j = 1, ..., J$ denote by $Z_{ij}$ the random variable whose value is to be the number of photons detected at detector $i$ during the scanning time that were emitted from pixel $j$. We assume that the members of the collection $\{Z_{ij} | i = 1, ..., I, j = 1, ..., J\}$ are independent. In keeping with standard practice in modelling radioactivity, we also assume the $Z_{ij}$ are Poisson distributed.

We assume that $Z_{ij}$ is a Poisson random variable whose mean value (and variance) is $\lambda_{ij} = P_{ij}x_j$. Here the $x_j \geq 0$ is the average rate of emission from pixel $j$, as discussed above, and $P_{ij} \geq 0$ is the probability that a photon emitted from pixel $j$ will be detected at detector $i$. We then define the random variables $Y_i = \sum_{j=1}^{J} Z_{ij}$, the total counts to be recorded at detector $i$; our actual count $y_i$ is then the observed value of the random variable $Y_i$. Note that the actual value of the individual $Z_{ij}$ are not observable.

So far the problem looks like a fairly standard parameter estimation problem of the sort studied in beginning statistics. There is one problem, however; we do not know what the $P_{ij}$ are. These values will vary from one patient to the next, since whether or not a photon makes it from a given pixel to a given detector depends on the geometric relationship between detector $i$ and pixel $j$, as well as what is in the patient's body between these two locations. If there are ribs or skull getting in the way, the probability of making it goes down. If there are just lungs, the probability goes up. There are additional complications when we try to image a beating heart. One way or another, we decide on our values of the $P_{ij}$.

# Chapter 60

# The EMML Algorithm

In our discussion of emission tomography we saw that the photon count data can reasonably be viewed as a linear superposition or mixture of finitely many independent Poisson random variables, whose mean values we wish to estimate. The *expectation maximization maximum likelihood method*, called the EM algorithm, is a general statistical procedure for iterative parameter estimation [82]. What we shall call the EMML method is the algorithm obtained when we apply the general EM algorithm to the particular problem posed by emission tomography [132], [133], [179]. As we shall see, the EMML can be used more generally to find approximate solutions of nonnegative systems of linear equations. The likelihood function we maximize here is closely related to a certain *cross-entropy* distance, leading us to a short discussion of entropy-maximizing methods.

Let $\{Z_{ij}, i = 1, ..., I, j = 1, ..., J\}$ be independent Poisson random variables, with $E(Z_{ij}) = P_{ij}x_j \geq 0$, where $P = [P_{ij}]$ is a matrix with nonnegative entries and $\mathbf{x} = (x_1, ..., x_J)^T$ is a column vector with nonnegative entries. Let $Y_i = \sum_{j=1}^{J} Z_{ij}, i = 1, ..., I$. Then the $\{Y_i, i = 1, ..., I\}$ are independent Poisson random variables, with $E(Y_i) = P\mathbf{x}_i = (P\mathbf{x})_i = \sum_{j=1}^{J} P_{ij}x_j$. For the sake of notational convenience we assume that the problem is normalized so that $\sum_i P_{ij} = 1$, for $j = 1, ..., J$; here $\sum_i = \sum_{i=1}^{I}$. The log likelihood function $LL_{\mathbf{y}}(\mathbf{x})$ now has the form

$$LL_{\mathbf{y}}(\mathbf{x}) = \sum\nolimits_i y_i \log(P\mathbf{x}_i) - P\mathbf{x}_i - \log(y_i!). \tag{60.1}$$

According to the Karush-Kuhn-Tucker theorem [155], at a maximizer $\hat{x}$ of $LL_{\mathbf{y}}(\mathbf{x})$ the gradient must have the properties

$$\nabla LL_{\mathbf{y}}(\hat{\mathbf{x}})_j = \sum\nolimits_i [\frac{y_i}{P\hat{\mathbf{x}}_i} - 1]P_{ij} \leq 0, \; j = 1, ..., J, \tag{60.2}$$

and

$$\nabla LL_{\mathbf{y}}(\hat{\mathbf{x}})_j = \sum\nolimits_i [\frac{y_i}{P\hat{\mathbf{x}}_i} - 1]P_{ij} = 0, \tag{60.3}$$

for all $j$ such that $\hat{x}_j > 0$. A closed form expression for the solution $\hat{\mathbf{x}}$ is not available and an iterative procedure is needed.

If we had observed the vector $\mathbf{z} = \{z_{ij}|i = 1, ..., I, j = 1, ..., J\}$, then we could maximize the log likelihood function $LL_{\mathbf{z}}(\mathbf{x})$, which has the form

$$LL_{\mathbf{z}}(\mathbf{x}) = \sum\nolimits_i \sum\nolimits_j z_{ij} \log(P_{ij}x_j) - P_{ij}x_j - \log(z_{ij}!). \tag{60.4}$$

The maximizing $\mathbf{x}$ can be obtained in closed form as

$$x_j = \sum\nolimits_i z_{ij}, \tag{60.5}$$

recalling that $\sum_i P_{ij} = 1, j = 1, ..., J$.

**The EM algorithm:** the general EM algorithm [82] is the following two-step iterative procedure. Having obtained $\mathbf{x}^k$, let $z_{ij}^k$ be the conditional expected value of $Z_{ij}$, conditioned on $\mathbf{x}^k$ and the data $\mathbf{y}$. Now we maximize $LL_{\mathbf{z}^k}(\mathbf{x})$ to get $\mathbf{x}^{k+1}$. Now increment $k$ to $k + 1$ and repeat the two steps.

Now we consider the EM algorithm as it applies in the Poisson case. Since $Z_{ij}$ is $P_{ij}x_j^k$-Poisson and the sum $\sum_j Z_{ij} = y_i$, we know that the conditional expected value is

$$z_{ij}^k = P_{ij}x_j^k \frac{y_i}{P\mathbf{x}_i^k}. \tag{60.6}$$

Now we maximize $LL_{\mathbf{z}^k}(\mathbf{x})$ to get $\mathbf{x}^{k+1}$; using (60.5), we have that

$$x_j^{k+1} = x_j^k \sum\nolimits_i P_{ij} \frac{y_i}{Px_i^k}, \tag{60.7}$$

for $j = 1, ..., J$. We begin with $x^0 > 0$ and proceed iteratively, as above. Then the sequence $\mathbf{x}^k$ converges to a maximizer of $LL_{\mathbf{y}}(\mathbf{x})$. We refer to this specific instance of the EM algorithm as the **EMML** algorithm.

For $a > 0$ and $b > 0$ let the *Kullback-Leibler* or *cross-entropy* distance from $a$ to $b$ be defined by

$$KL(a, b) = a \log \frac{a}{b} + b - a \geq 0,$$

with $KL(0, b) = b$ and $KL(a, 0) = +\infty$. For vectors $\mathbf{a} = (a_1, ..., a_N)^T$ and $\mathbf{b} = (b_1, ..., b_N)^T$ with nonnegative entries define

$$KL(\mathbf{a}, \mathbf{b}) = \sum\nolimits_{n=1}^{N} KL(a_n, b_n).$$

If $\mathbf{1} = (1, 1, ..., 1)^T$ then

$$KL(\mathbf{a}, \mathbf{1}) = N + \sum\nolimits_{n=1}^{N} a_n \log a_n - a_n;$$

the sum

$$-\sum\nolimits_{n=1}^{N} a_n \log a_n - a_n$$

is sometimes called the *Shannon entropy* of the vector $\mathbf{a}$. The quantity

$$-KL(\mathbf{1}, \mathbf{b}) = \sum\nolimits_{n=1}^{N} \log b_n - b_n$$

is sometimes called the *Burg entropy* of the vector $\mathbf{b}$.

The negative of the likelihood function above is, except for terms not involving the variable $\mathbf{x}$, equal to the quantity $KL(\mathbf{y}, P\mathbf{x})$. The following convergence theorem for the EMML algorithm is due to Csiszár and Tusnády [76].

**Theorem 60.1** *For any positive starting vector $\mathbf{x}^0$ the EMML sequence $\mathbf{x}^k$ converges to a nonnegative minimizer $\mathbf{x}^\infty$ of $KL(\mathbf{y}, P\mathbf{x})$. If the linear system of equations $\mathbf{y} = P\mathbf{x}$ has nonnegative solutions, then $\mathbf{y} = P\mathbf{x}^\infty$. For any nonnegative minimizer $\hat{\mathbf{x}}$ of $KL(\mathbf{y}, P\mathbf{x})$, we have $KL(\hat{\mathbf{x}}, \mathbf{x}^\infty) < +\infty$, so the support of the vector $\mathbf{x}^\infty$ must be maximal with respect to all nonnegative minimizers of $KL(\mathbf{y}, P\mathbf{x})$.*

In the inconsistent case, in which the system $\mathbf{y} = P\mathbf{x}$ has no nonnegative solutions, the nonnegative minimizer of $KL(\mathbf{y}, P\mathbf{x})$ is almost always unique, regardless of the relative sizes of $I$ and $J$, as the following theorem shows [29]. Say that the matrix $P$ has the 'full rank property' (FRP) if $P$ and every submatrix obtained from $P$ by deleting columns have full rank.

**Theorem 60.2** *Let $P$ have the FRP and let $\mathbf{y} = P\mathbf{x}$ have no nonnegative solution. Then there is a subset $S$ of $\{j = 1, ..., J\}$, having cardinality at most $I - 1$, with the property that any nonnegative minimizer $\hat{\mathbf{x}}$ of $KL(\mathbf{y}, P\mathbf{x})$ has positive entries, $\hat{x}_j > 0$, only if $j \in S$. Consequently, $\hat{\mathbf{x}}$ is unique.*

## Maximum entropy solutions:

Suppose that the system $\mathbf{y} = P\mathbf{x}$ has nonnegative solutions. We sometimes seek the solution having the maximum Shannon entropy; that is, we want to maximize $KL(\mathbf{x}, \mathbf{1})$, subject to $\mathbf{y} = P\mathbf{x}$. Although the EMML algorithm gives a nonnegative solution it will not generally be the maximum Shannon entropy solution. On the other hand, the *simultaneous multiplicative* ART (SMART) algorithm does give the maximum Shannon entropy solution.

The SMART is an iterative algorithm with the following iterative step:

$$x_j^{k+1} = x_j^k \exp[\sum\nolimits_i P_{ij} \log \frac{y_i}{P\mathbf{x}_i^k}],$$

for $j = 1, ..., J$. When there are nonnegative solutions to $\mathbf{y} = P\mathbf{x}$ the SMART converges to that solution minimizing $KL(\mathbf{x}, \mathbf{x}^0)$, where $\mathbf{x}^0 > 0$ is the starting vector; if $\mathbf{x}^0 = \mathbf{1}$ then we get the maximum Shannon entropy solution. If there are no nonnegative solutions of $\mathbf{y} = P\mathbf{x}$ then the SMART converges to the minimizer of $KL(P\mathbf{x}, \mathbf{y})$ for which $KL(\mathbf{x}, \mathbf{x}^0)$ is minimized.

## Transforming from a general linear system to a nonnegative one

Suppose that $H\mathbf{c} = \mathbf{d}$ is an arbitrary (real) system of linear equations, with the matrix $H = [H_{ij}]$. Rescaling the equations if necessary, we may assume that for each $j$ the column sum $\sum_i H_{ij}$ is nonzero; note that if a particular rescaling of one equation to make the first column sum nonzero causes another column sum to become zero, we simply choose a different rescaling. Since there are finitely many columns to worry about, we can always succeed in making all the column sums nonzero. Now redefine $H$ and $\mathbf{c}$ as follows: replace $H_{kj}$ with $G_{kj} = \frac{H_{kj}}{\sum_i H_{ij}}$ and $c_j$ with $g_j = c_j \sum_i H_{ij}$; the product $H\mathbf{c}$ is equal to $G\mathbf{g}$ and the new matrix $G$ has column sums equal to one. The system $G\mathbf{g} = \mathbf{d}$ still holds, but now we know that $\sum_i d_i = d_+ = \sum_j g_j = g_+$. Let $U$ be the matrix whose entries are all one and let $t \geq 0$ be large enough so that $B = G + tU$ has all nonnegative entries. Then $B\mathbf{g} = G\mathbf{g} + (tg_+)\mathbf{1}$, where $\mathbf{1}$ is the vector whose entries are all one. So the new system of equations to solve is $B\mathbf{g} = \mathbf{d} + (td_+)\mathbf{1} = \mathbf{y}$.

In the algorithms of interest to us we often made the further assumption that the column sums of the matrix are all one. To achieve this, we make one additional renormalization: replace $B_{kj}$ with $P_{kj} = \frac{B_{kj}}{\sum_i B_{ij}}$ and $g_j$ with $x_j = g_j \sum_i B_{ij}$; the product $B\mathbf{g}$ is equal to $P\mathbf{x}$ and the new matrix $P$ is nonnegative and has column sums equal to one.

# Chapter 61

# A Tale of Two Algorithms

The *expectation maximization maximum likelihood method* (EMML) discussed in the previous chapter has been the subject of much attention in the medical imaging literature over the past decade. Statisticians like it because it is based on the well studied principle of likelihood maximization for parameter estimation. Physicists like it because, unlike its competition, filtered backprojection, it permits the inclusion of sophisticated models of the physical situation. Mathematicians like it because it can be derived from iterative optimization theory. Physicians like it because the images are better than those produced by other means. No method is perfect, however, and the EMML suffers from sensitivity to noise and slow rate of convergence. Research is ongoing to find faster and less sensitive versions of this algorithm.

Another class of iterative algorithms were introduced into medical imaging by Gordon *et al* in [102]. These include the *algebraic reconstruction technique* (ART) and its multiplicative version, MART. These methods were derived by viewing image reconstruction as solving systems of linear equations, possibly subject to constraints, such as positivity. The *simultaneous* MART (SMART) [80], [162] is a variant of MART that uses all the data at each step of the iteration.

Although the EMML and SMART algorithms have quite different histories and are not typically considered together they are closely related [29], [30]. In this chapter we examine these two algorithms in tandem, following [31]. Forging a link between the EMML and SMART led to a better understanding of both of these algorithms and to new results. The proof of convergence of the SMART in the inconsistent case [29] was based on the analogous proof for the EMML [179], while discovery of the faster version of the EMML, the *rescaled block-iterative* EMML (RBI-EMML) [32] came from studying the analogous block-iterative version of SMART [62]. The proofs we give here are elementary and rely mainly on easily established

properties of the cross-entropy.

For $a > 0$ and $b > 0$ define the cross-entropy or Kullback-Leibler distance

$$KL(a, b) = a \log(\frac{a}{b}) + b - a.$$

Let $KL(a, 0) = +\infty$ and $KL(0, b) = b$. For nonnegative vectors $\mathbf{x}$ and $\mathbf{z}$ define $KL(\mathbf{x}, \mathbf{z})$ component-wise:

$$KL(\mathbf{x}, \mathbf{z}) = \sum_{j=1}^{J} KL(x_j, z_j).$$

Note that the KL distance has the property $KL(c\mathbf{x}, c\mathbf{z}) = cKL(\mathbf{x}, \mathbf{z})$ for all positive scalars $c$.

**Exercise 1:** Let $z_+ = \sum_{j=1}^{J} z_j > 0$. Then

$$KL(\mathbf{x}, \mathbf{z}) = KL(x_+, z_+) + KL(\mathbf{x}, (x_+/z_+)\mathbf{z}). \tag{61.1}$$

As we shall see, the KL distance mimics the ordinary Euclidean distance in several ways that make it particularly useful in designing optimization algorithms.

Let $P$ be an $I$ by $J$ matrix with entries $P_{ij} \geq 0$, such that, for each $j = 1, ..., J$, we have $s_j = \sum_{i=1}^{I} P_{ij} > 0$. Let $\mathbf{y} = (y_1, ..., y_I)^T$ with $y_i > 0$ for each $i$. We shall assume throughout this chapter that $s_j = 1$ for each $j$. If this is not the case initially, we replace $x_j$ with $x_j s_j$ and $P_{ij}$ with $P_{ij}/s_j$; the quantities $(P\mathbf{x})_i$ are unchanged.

For each nonnegative vector $\mathbf{x}$ for which $(P\mathbf{x})_i = \sum_{j=1}^{J} P_{ij} x_j > 0$ let $r(\mathbf{x}) = \{r(\mathbf{x})_{ij}\}$ and $q(\mathbf{x}) = \{q(\mathbf{x})_{ij}\}$ be the $I$ by $J$ arrays with entries

$$r(\mathbf{x})_{ij} = x_j P_{ij} \frac{y_i}{(P\mathbf{x})_i}$$

and

$$q(\mathbf{x})_{ij} = x_j P_{ij}.$$

The KL distances

$$KL(r(\mathbf{x}), q(\mathbf{z})) = \sum_{i=1}^{I} \sum_{j=i}^{J} KL(r(\mathbf{x})_{ij}, q(\mathbf{z})_{ij})$$

and

$$KL(q(\mathbf{x}), r(\mathbf{z})) = \sum_{i=1}^{I} \sum_{j=1}^{J} KL(q(\mathbf{x})_{ij}, r(\mathbf{z})_{ij})$$

will play important roles in the discussion that follows. Note that if there is nonnegative $\mathbf{x}$ with $r(\mathbf{x}) = q(\mathbf{x})$ then $\mathbf{y} = P\mathbf{x}$.

**Some Pythagorean identities involving the KL distance:** The iterative algorithms we discuss in this chapter are derived using the principle of *alternating minimization*, according to which the distances $KL(r(\mathbf{x}), q(\mathbf{z}))$ and $KL(q(\mathbf{x}), r(\mathbf{z}))$ are minimized, first with respect to the variable $\mathbf{x}$ and then with respect to the variable $\mathbf{z}$. Although the KL distance is not Euclidean, and, in particular, not even symmetric, there are analogues of Pythagoras' theorem that play important roles in the convergence proofs.

**Exercise 2:** Establish the following *Pythagorean identities*:

$$KL(r(\mathbf{x}), q(\mathbf{z})) = KL(r(\mathbf{z}), q(\mathbf{z})) + KL(r(\mathbf{x}), r(\mathbf{z})); \tag{61.2}$$

$$KL(r(\mathbf{x}), q(\mathbf{z})) = KL(r(\mathbf{x}), q(\mathbf{x}')) + KL(\mathbf{x}', \mathbf{z}), \tag{61.3}$$

for

$$x'_j = x_j \sum_{i=1}^{I} P_{ij} \frac{y_i}{(P\mathbf{x})_i}; \tag{61.4}$$

$$KL(q(\mathbf{x}), r(\mathbf{z})) = KL(q(\mathbf{x}), r(\mathbf{x})) + KL(\mathbf{x}, \mathbf{z}) - KL(P\mathbf{x}, P\mathbf{z}); \tag{61.5}$$

$$KL(q(\mathbf{x}), r(\mathbf{z})) = KL(q(\mathbf{z}''), r(\mathbf{z})) + KL(\mathbf{x}, \mathbf{z}''), \tag{61.6}$$

for

$$z''_j = z_j \exp(\sum_{i=1}^{I} P_{ij} \log \frac{y_i}{(P\mathbf{z})_i}). \tag{61.7}$$

Note that it follows from equation (61.1) that $KL(\mathbf{x}, \mathbf{z}) - KL(P\mathbf{x}, P\mathbf{z}) \geq 0$.

**The two algorithms:** The algorithms we shall consider are the *expectation maximization maximum likelihood* method (EMML) and the *simultaneous multiplicative algebraic reconstruction technique* (SMART). When $\mathbf{y} = P\mathbf{x}$ has nonnegative solutions both algorithms produce such a solution. In general, the EMML gives a nonnegative minimizer of $KL(\mathbf{y}, P\mathbf{x})$, while the SMART minimizes $KL(P\mathbf{x}, \mathbf{y})$ over nonnegative $\mathbf{x}$.

For both algorithms we begin with an arbitrary positive vector $\mathbf{x}^0$. The iterative step for the EMML method is

**EMML:**

$$x_j^{k+1} = (\mathbf{x}^k)'_j = x_j^k \sum_{i=1}^{I} P_{ij} \frac{y_i}{(P\mathbf{x}^k)_i}. \tag{61.8}$$

The iterative step for the SMART is

**SMART:**

$$x_j^{m+1} = (\mathbf{x}^m)''_j = x_j^m \exp\Big(\sum_{i=1}^{I} P_{ij} \log \frac{y_i}{(P\mathbf{x}^m)_i}\Big). \tag{61.9}$$

Note that, to avoid confusion, we use $k$ for the iteration number of the EMML and $m$ for the SMART.

**Exercise 3:** Show that, for $\{\mathbf{x}^k\}$ given by equation (61.8), $\{KL(\mathbf{y}, P\mathbf{x}^k)\}$ is decreasing and $\{KL(\mathbf{x}^{k+1}, \mathbf{x}^k)\} \to 0$. Show that, for $\{\mathbf{x}^m\}$ given by equation (61.9), $\{KL(P\mathbf{x}^m, \mathbf{y})\}$ is decreasing and $\{KL(\mathbf{x}^m, \mathbf{x}^{m+1})\} \to 0$.

**Hints:** Use $KL(r(\mathbf{x}), q(\mathbf{x})) = KL(\mathbf{y}, P\mathbf{x})$, $KL(q(\mathbf{x}), r(\mathbf{x})) = KL(P\mathbf{x}, \mathbf{y})$ and the Pythagorean identities.

**Exercise 4:** Show that the EMML sequence $\{\mathbf{x}^k\}$ is bounded by showing

$$\sum_{j=1}^{J} x_j^k = \sum_{i=1}^{I} y_i.$$

Show that the SMART sequence $\{\mathbf{x}^m\}$ is bounded by showing that

$$\sum_{j=1}^{J} x_j^m \leq \sum_{i=1}^{I} y_i.$$

**Exercise 5:** Show that $(\mathbf{x}^*)' = \mathbf{x}^*$ for any cluster point $\mathbf{x}^*$ of the EMML sequence $\{\mathbf{x}^k\}$ and that $(\mathbf{x}^*)'' = \mathbf{x}^*$ for any cluster point $\mathbf{x}^*$ of the SMART sequence $\{\mathbf{x}^m\}$.

**Hint:** Use the facts that $\{KL(\mathbf{x}^{k+1}, \mathbf{x}^k)\} \to 0$ and $\{KL(\mathbf{x}^m, \mathbf{x}^{m+1})\} \to 0$.

**Exercise 6:** Let $\hat{\mathbf{x}}$ and $\tilde{\mathbf{x}}$ minimize $KL(\mathbf{y}, P\mathbf{x})$ and $KL(P\mathbf{x}, \mathbf{y})$, respectively, over all $\mathbf{x} \geq \mathbf{0}$. Then $(\hat{\mathbf{x}})' = \hat{\mathbf{x}}$ and $(\tilde{\mathbf{x}})'' = \tilde{\mathbf{x}}$.

**Hints:** Apply Pythagorean identities to $KL(r(\hat{\mathbf{x}}), q(\hat{\mathbf{x}}))$ and $KL(q(\tilde{\mathbf{x}}), r(\tilde{\mathbf{x}}))$.

Note that, because of convexity properties of the KL distance, even if the minimizers $\hat{\mathbf{x}}$ and $\tilde{\mathbf{x}}$ are not unique, the vectors $P\hat{\mathbf{x}}$ and $P\tilde{\mathbf{x}}$ are unique.

**Exercise 7:** For the EMML sequence $\{\mathbf{x}^k\}$ with cluster point $\mathbf{x}^*$ and $\hat{\mathbf{x}}$ as above we have the *double inequality*

$$KL(\hat{\mathbf{x}}, \mathbf{x}^k) \geq KL(r(\hat{\mathbf{x}}), r(\mathbf{x}^k)) \geq KL(\hat{\mathbf{x}}, \mathbf{x}^{k+1}), \tag{61.10}$$

from which we conclude that the sequence $\{KL(\hat{\mathbf{x}}, \mathbf{x}^k)\}$ is decreasing and $KL(\hat{\mathbf{x}}, \mathbf{x}^*) < +\infty$.

**Hint:** For the first inequality calculate $KL(r(\hat{\mathbf{x}}), q(\mathbf{x}^k))$ two ways. For the second one, use $(\mathbf{x})'_j = \sum_{i=1}^{I} r(\mathbf{x})_{ij}$ and Exercise 1.

**Exercise 8:** For the SMART sequence $\{\mathbf{x}^m\}$ with cluster point $\mathbf{x}^*$ and $\tilde{\mathbf{x}}$ as above we have

$$KL(\tilde{\mathbf{x}}, \mathbf{x}^m) - KL(\tilde{\mathbf{x}}, \mathbf{x}^{m+1}) = KL(P\mathbf{x}^{m+1}, \mathbf{y}) - KL(P\tilde{\mathbf{x}}, \mathbf{y}) +$$

$$KL(P\tilde{\mathbf{x}}, P\mathbf{x}^m) + KL(\mathbf{x}^{m+1}, \mathbf{x}^m) - KL(P\mathbf{x}^{m+1}, P\mathbf{x}^m), \tag{61.11}$$

from which we conclude that the sequence $\{KL(\tilde{\mathbf{x}}, \mathbf{x}^m)\}$ is decreasing, $KL(P\tilde{\mathbf{x}}, P\mathbf{x}^*) = 0$ and $KL(\tilde{\mathbf{x}}, \mathbf{x}^*) < +\infty$.

**Hint:** Expand $KL(q(\tilde{\mathbf{x}}), r(\mathbf{x}^m))$ using the Pythagorean identities.

**Exercise 9:** For $\mathbf{x}^*$ a cluster point of the EMML sequence $\{\mathbf{x}^k\}$ we have $KL(\mathbf{y}, P\mathbf{x}^*) = KL(\mathbf{y}, P\hat{\mathbf{x}})$. Therefore $\mathbf{x}^*$ is a nonnegative minimizer of $KL(\mathbf{y}, P\mathbf{x})$. Consequently, the sequence $\{KL(\mathbf{x}^*, \mathbf{x}^k)\}$ converges to zero, and so $\{\mathbf{x}^k\} \to \mathbf{x}^*$.

**Hint:** Use the double inequality (61.10) and $KL(r(\hat{\mathbf{x}}), q(\mathbf{x}^*))$.

**Exercise 10:** For $\mathbf{x}^*$ a cluster point of the SMART sequence $\{\mathbf{x}^m\}$ we have $KL(P\mathbf{x}^*, \mathbf{y}) = KL(P\tilde{\mathbf{x}}, \mathbf{y})$. Therefore $\mathbf{x}^*$ is a nonnegative minimizer of $KL(P\mathbf{x}, \mathbf{y})$. Consequently, the sequence $\{KL(\mathbf{x}^*, \mathbf{x}^m)\}$ converges to zero, and so $\{\mathbf{x}^m\} \to \mathbf{x}^*$. Moreover,

$$KL(\tilde{\mathbf{x}}, \mathbf{x}^0) \geq KL(\mathbf{x}^*, \mathbf{x}^0)$$

for all $\tilde{\mathbf{x}}$ as above.

**Hints:** Use Exercise 8. For the final assertion use the fact that the difference $KL(\tilde{\mathbf{x}}, \mathbf{x}^m) - KL(\tilde{\mathbf{x}}, \mathbf{x}^{m+1})$ is independent of the choice of $\tilde{\mathbf{x}}$, since it depends only on $P\mathbf{x}^* = P\tilde{\mathbf{x}}$. Now sum over the index $m$.

Both the EMML and the SMART algorithms are slow to converge. For that reason attention has shifted, in recent years, to *block iterative* versions of these algorithms. We take up that topic in the next chapter.

# Chapter 62

# List-mode EMML in PET imaging

We saw earlier in our brief discussion of positron emission tomography (PET) that a detection in PET is the nearly simultaneous recording of photon arrival at two separate detector locations. The detection is then associated with the line segment having these two locations as end points and it is assumed that the original positron emission occurred somewhere along that line segment. Such line segments are called *lines of response* (LOR).

In the case of SPECT we know in advance the finite set of detector locations at which photon arrivals can be detected. The data is then the number of such arrivals recorded at each of these locations. In the case of PET we maintain a list of the LOR associated with detections. We have a choice to make now. For each pair of end points $\mathbf{x}_1 = (x_1, y_1, z_1)$ and $\mathbf{x}_2 = (x_2, y_2, z_2)$ there is a LOR $\lambda(\mathbf{x}_1, \mathbf{x}_2)$ connecting these two points. We can identify a very large, but finite, set of locations capable of serving as the end points of LOR, in which case we posit *a priori* a very large, but finite, set $\{\lambda_i,\, i = 1, ..., I\}$ containing these LOR. On the other hand, we can imagine a continuum of possible LOR.

In the first (finite) case we must specify the nonnegative quantities $P_{ij}$, the probability that a positron emission at voxel $j$ will be detected and associated with LOR $\lambda_i$. Then the sum

$$s_j = \sum_{i=1}^{I} P_{ij}$$

is the probability that an emission at voxel $j$ will be detected.

In the second (continuously infinite) case we have to specify, for each voxel $j$, a probability density function (pdf) $f_j(\lambda)$ describing the random

distribution of LOR due to emissions at voxel $j$. In this second case the distributions $f_j$ over the space of all LOR $\lambda(\mathbf{x}_1, \mathbf{x}_2)$ can be viewed as a distribution over the space of all pairs of end points $(\mathbf{x}_1, \mathbf{x}_2)$. In addition we must specify the probability $g(\lambda)$ that a photon pair travelling along LOR $\lambda$ will be detected.

The first choice, the finite case, is the one adopted by Huesman *et al* [119], while Barrett *et al* make the second choice, the continuum model [7, 153]. In either case the data consists of a list of the LOR associated with an emission, rather than counts, hence the term *list-mode*. We suppose that $N$ LOR are on the list. Regardless of which case we are in, we denote these LOR by $\{\lambda_n,\, n = 1, ..., N\}$.

In all of the papers just cited the EMML algorithm is chosen for the reconstruction. For list-mode processing the EMML iterative step is the following:

**List-mode EMML:**

$$x_j^{k+1} = d_j^{-1} x_j^k \sum_{n=1}^{N} P_{nj} \frac{1}{(P\mathbf{x}^k)_n}, \tag{62.1}$$

where $d_j$ is the probability of detecting an emission from voxel $j$. In the finite case $d_j = s_j$. In the continuum case $P_{nj}$ is the value of the pdf $f_j$ at the $n$th LOR on the list, that is, $P_{nj} = f_j(\lambda_n)$ and

$$d_j = \int f_j(\lambda) g(\lambda) d\lambda.$$

In the finite case the EMML algorithm is a special case of the algorithm used in SPECT. In the second case, however, there is some modification necessary. The issue here is the role of the term $d_j$ and its relation to the $P_{nj}$. Because the $P_{nj}$ are values of a pdf they can take on any positive values and are not restricted to lie within $[0, 1]$. The $d_j$ is not the sum of the $P_{nj}$ over the index $n$. Convergence of the EMML algorithm in the second, continuum case does not follow from results concerning the finite case. Nevertheless, the EMML algorithm in the continuum case can be shown to converge to a maximizer of the likelihood [38].

We can convert the quantities $P_{nj}$ into probabilities by dividing each one by the sum

$$t_j = \sum_{n=1}^{N} P_{nj}.$$

Let $R$ be the matrix with entries $R_{nj} = P_{nj}/t_j$. To use the EMML algorithm as given in equation (62.1) we need only the relative probabilities represented by the $R_{nj}$, along with the overall sensitivity coefficients $d_j$;

we do not need to specify the $f_j$ explicitly. Indeed, we can rewrite equation (62.1) as

$$z_j^{k+1} = d_j^{-1} t_j z_j^k \sum_{n=1}^{N} R_{nj} \frac{1}{(R\mathbf{z}^k)_n} \tag{62.2}$$

for $z_j^k = t_j x_j^k$.

Suppose, after the list has been created, we treat the $N$ LOR on the list as the only ones that could have been there, in effect putting us into the first (finite) case, with $N$ replacing $I$ now. Since $\sum_{n=1}^{N} R_{nj} = 1$ for each $j$, we are implicitly assuming that with probability one all emissions are detected. The parameters we seek now are $w_j = x_j d_j$, the *detected intensity* at voxel $j$. The iterative step of the EMML algorithm is then

$$w_j^{k+1} = w_j^k \sum_{n=1}^{N} R_{nj} \frac{1}{(R\mathbf{w}^k)_n}. \tag{62.3}$$

This iteration converges to a nonnegative minimizer of the KL distance $KL(\mathbf{u}, R\mathbf{w})$, where $\mathbf{u}$ is the vector whose entries are all one.

# Chapter 63

# Maximum *a posteriori* estimation

The EMML iterative algorithm maximizes the likelihood function for the case in which the entries of the data vector $\mathbf{y} = (y_1, ..., y_I)^T$ are assumed to be samples of independent Poisson random variables with mean values $(P\mathbf{x})_i$; here $P$ is an $I$ by $J$ matrix with nonnegative entries and $\mathbf{x} = (x_1, ..., x_J)^T$ is the vector of nonnegative parameters to be estimated. Equivalently, it minimizes the Kullback-Leibler distance $KL(\mathbf{y}, P(\mathbf{x}))$. This situation arises in single photon emission tomography, where the $y_i$ are the number of photons counted at each detector $i$, $\mathbf{x}$ is the vectorized image to be reconstructed and its entries $x_j$ are (proportional to) the radionuclide intensity levels at each voxel $j$. When the signal-to-noise ratio is low, which is almost always the case in medical applications, maximizing likelihood can lead to unacceptably noisy reconstructions, particularly when $J$ is larger than $I$. One way to remedy this problem is simply to halt the EMML algorithm after a few iterations, to avoid over-fitting the $\mathbf{x}$ to the noisy data. A more mathematically sophisticated remedy is to employ a Bayesian approach and seek a maximum *a posteriori* (MAP) estimate of $\mathbf{x}$.

In the Bayesian approach we view $\mathbf{x}$ as an instance of a random vector having a probability density function $f(\mathbf{x})$. Instead of maximizing the likelihood given the data we now maximize the posterior likelihood, given both the data and the prior distribution for $\mathbf{x}$. This is equivalent to minimizing

$$F(\mathbf{x}) = KL(\mathbf{y}, P(\mathbf{x})) - \log f(\mathbf{x}). \tag{63.1}$$

As we saw earlier, the EMML algorithm is an example of an optimization method based on alternating minimization of a function of two vector variables. The alternating minimization works this way: let $\mathbf{x}$ and $\mathbf{z}$ be vector variables and $H(\mathbf{x}, \mathbf{z}) > 0$. If we fix $\mathbf{z}$ and minimize $H(\mathbf{x}, \mathbf{z})$ with

respect to $\mathbf{x}$ we find that the solution is $\mathbf{x} = \mathbf{z}$, the vector we fixed; that is, $H(\mathbf{x}, \mathbf{z}) \geq H(\mathbf{z}, \mathbf{z})$ always. If we fix $\mathbf{x}$ and minimize $H(\mathbf{x}, \mathbf{z})$ with respect to $\mathbf{z}$ we get something new; call it $T\mathbf{x}$. The EMML algorithm has the iterative step $\mathbf{x}^{k+1} = T\mathbf{x}^k$.

Obviously, we can't use an arbitrary function $H$; it must be related to $KL(\mathbf{y}, P\mathbf{x})$ that we wish to minimize and we must be able to obtain each intermediate optimizer in closed form. The clever step is to select $H(\mathbf{x}, \mathbf{z})$ so that $H(\mathbf{x}, \mathbf{x}) = KL(\mathbf{y}, P\mathbf{x})$, for any $\mathbf{x}$. Now see what we have so far:

$$KL(\mathbf{y}, P\mathbf{x}^k) = H(\mathbf{x}^k, \mathbf{x}^k) \geq H(\mathbf{x}^k, \mathbf{x}^{k+1}) \geq H(\mathbf{x}^{k+1}, \mathbf{x}^{k+1}) = KL(y, P\mathbf{x}^{k+1}).$$

That tells us that the algorithm makes $KL(\mathbf{y}, P\mathbf{x}^k)$ decrease with each iteration. The proof doesn't stop here, but at least it is now plausible that the EMML iteration could minimize $KL(\mathbf{y}, P\mathbf{x})$.

The function $H(\mathbf{x}, \mathbf{z})$ used in the EMML case is the KL distance

$$H(\mathbf{x}, \mathbf{z}) = KL(r(\mathbf{x}), q(\mathbf{z})) = \sum_{i=1}^{I} \sum_{j=i}^{J} KL(r(\mathbf{x})_{ij}, q(\mathbf{z})_{ij}). \tag{63.2}$$

With $\mathbf{x} = \mathbf{x}^k$ fixed, we minimize with respect to $\mathbf{z}$ to obtain the next EMML iterate $\mathbf{x}^{k+1}$. As before, we define, for each nonnegative vector $\mathbf{x}$ for which $(P\mathbf{x})_i = \sum_{j=1}^{J} P_{ij} x_j > 0$, the arrays $r(\mathbf{x}) = \{r(\mathbf{x})_{ij}\}$ and $q(\mathbf{x}) = \{q(\mathbf{x})_{ij}\}$ with entries

$$r(\mathbf{x})_{ij} = x_j P_{ij} \frac{y_i}{(P\mathbf{x})_i}$$

and

$$q(\mathbf{x})_{ij} = x_j P_{ij}.$$

Having selected the prior pdf $f(\mathbf{x})$ we want an iterative algorithm to minimize the function $F(\mathbf{x})$ in equation (63.1). It would be a great help if we could mimic the alternating minimization formulation and obtain $\mathbf{x}^{k+1}$ by minimizing

$$KL(r(\mathbf{x}^k), q(\mathbf{z})) - \log f(\mathbf{z}) \tag{63.3}$$

with respect to $\mathbf{z}$. Unfortunately, to be able to express each new $\mathbf{x}^{k+1}$ in closed form we need to choose $f(\mathbf{x})$ carefully.

**The Gamma prior distribution for x:** In [133] Lange *et al* suggest viewing the entries $x_j$ as samples of independent gamma-distributed random variables. A gamma-distributed random variable $x$ takes positive values and has for its pdf the *gamma distribution* defined for positive $x$ by

$$\gamma(x) = \frac{1}{\Gamma(\alpha)} (\frac{\alpha}{\beta})^{\alpha} x^{\alpha-1} e^{-\alpha x/\beta},$$

where $\alpha$ and $\beta$ are positive parameters and $\Gamma$ denotes the gamma function. The mean of such a gamma-distributed random variable is then $\mu = \beta$ and the variance is $\sigma^2 = \beta^2/\alpha$.

**Exercise 1:** Show that if the entries $z_j$ of $\mathbf{z}$ are viewed as independent and gamma-distributed with means $\mu_j$ and variances $\sigma_j^2$ then minimizing (63.3) with respect to $\mathbf{z}$ is equivalent to minimizing the function

$$KL(r(\mathbf{x}^k), q(\mathbf{z})) + \sum_{j=1}^{J} \delta_j KL(\gamma_j, z_j), \tag{63.4}$$

for

$$\delta_j = \frac{\mu_j}{\sigma_j^2},\ \gamma_j = \frac{\mu_j^2 - \sigma_j^2}{\mu_j},$$

under the assumption that the latter term is positive. Show further that the resulting $x^{k+1}$ has entries given in closed form by

$$x_j^{k+1} = \frac{\delta_j}{\delta_j + s_j}\gamma_j + \frac{1}{\delta_j + s_j} x_j^k \sum_{i=1}^{I} P_{ij} y_i / (P\mathbf{x}^k)_i, \tag{63.5}$$

where $s_j = \sum_{i=1}^{I} P_{ij}$.

We see from equation (63.5) that the MAP iteration using the gamma priors generates a sequence of estimates each entry of which is a convex combination or weighted arithmetic mean of the result of one EMML step and the prior estimate $\gamma_j$. Convergence of the resulting iterative sequence is established in [133]; see also [29].

**The one-step-late alternative:** It may well happen that we do not wish to use the gamma priors model and prefer some other $f(\mathbf{x})$. Because we will not be able to find a closed form expression for the $\mathbf{z}$ minimizing the function in equation (63.3) we need some other way to proceed with the alternating minimization. Green [103] has offered the *one-step-late* (OSL) alternative. When we try to minimize the function in (63.3) by setting the gradient to zero we replace the variable $\mathbf{z}$ that occurs in the gradient of the term $-\log f(\mathbf{z})$ with $\mathbf{x}^k$, the previously calculated iterate. Then we can solve for $\mathbf{z}$ in closed form to obtain the new $\mathbf{x}^{k+1}$. Unfortunately, negative entries can result and convergence is not guaranteed. There is a sizable literature on the use of MAP methods for this problem. In [37] an interior point algorithm (IPA) is presented that avoids the OSL issue. In [146] the IPA is used to regularize transmission tomographic images.

**Regularizing the SMART:** In the presence of noisy data the SMART algorithm suffers from the same problem that afflicts the EMML, overfitting

to noisy data resulting in an unacceptably noisy image. As we saw earlier, there is a close connection between the EMML and SMART algorithms. This suggests that a regularization method for SMART can be developed along the lines of the MAP with gamma priors used for EMML. Since the SMART is obtained by minimizing the function

$$KL(q(\mathbf{z}), r(\mathbf{x}^k))$$

with respect to $\mathbf{z}$ to obtain $\mathbf{x}^{k+1}$ it seems reasonable to attempt to derive a regularized SMART iterative scheme by minimizing

$$KL(q(\mathbf{z}), r(\mathbf{x}^k)) + \sum_{j=1}^{J} \delta_j KL(z_j, \gamma_j), \tag{63.6}$$

for selected positive parameters $\delta_j$ and $\gamma_j$.

**Exercise 2:** Show that the $z_j$ minimizing the function in (63.6) can be expressed in closed form and that the resulting $\mathbf{x}^{k+1}$ has entries that satisfy

$$\log x_j^{k+1} = \frac{\delta_j}{\delta_j + s_j} \log \gamma_j + \frac{1}{\delta_j + s_j} x_j^k \sum_{i=1}^{I} P_{ij} \log[y_i/(P\mathbf{x}^k)_i]. \tag{63.7}$$

In [29] it was shown that this iterative sequence converges to a minimizer of the function

$$KL(P\mathbf{x}, \mathbf{y}) + \sum_{j=1}^{J} \delta_j KL(x_j, \gamma_j).$$

It is useful to note that although it may be possible to rederive this minimization problem within the framework of Bayesian MAP estimation by carefully selecting a prior pdf for the vector $\mathbf{x}$ we have not done so. The MAP approach is a special case of regularization through the use of penalty functions. Those penalty functions need not arise through a Bayesian formulation of the parameter estimation problem.

**De Pierro's surrogate function method:** In [83] De Pierro presents a modified EMML algorithm that includes regularization in the form of a penalty function. His objective is the same as ours was in the case of regularized SMART: to embed the penalty term in the alternating minimization framework in such a way as to make it possible to obtain the next iterate in closed form. Because his *surrogate function* method has been used subsequently by others to obtain penalized likelihood algorithms [64] we consider his approach in some detail.

Let $\mathbf{x}$ and $\mathbf{z}$ be vector variables and $H(\mathbf{x}, \mathbf{z}) > 0$. Mimicking the behavior of the function $H(\mathbf{x}, \mathbf{z})$ used in equation (63.2) we require that if we fix

$\mathbf{z}$ and minimize $H(\mathbf{x}, \mathbf{z})$ with respect to $\mathbf{x}$ the solution should be $\mathbf{x} = \mathbf{z}$, the vector we fixed; that is, $H(\mathbf{x}, \mathbf{z}) \geq H(\mathbf{z}, \mathbf{z})$ always. If we fix $\mathbf{x}$ and minimize $H(\mathbf{x}, \mathbf{z})$ with respect to $\mathbf{z}$ we should get something new; call it $T\mathbf{x}$. As with the EMML, the algorithm will have the iterative step $\mathbf{x}^{k+1} = T\mathbf{x}^k$.

Summarizing, we see that we need a function $H(\mathbf{x}, \mathbf{z})$ with the properties 1) $H(\mathbf{x}, \mathbf{z}) \geq H(\mathbf{z}, \mathbf{z})$ for all $\mathbf{x}$ and $\mathbf{z}$; 2) $H(\mathbf{x}, \mathbf{x})$ is the function $F(\mathbf{x})$ we wish to minimize; and 3) minimizing $H(\mathbf{x}, \mathbf{z})$ with respect to $\mathbf{z}$ for fixed $\mathbf{x}$ is easy.

The function to be minimized is

$$F(\mathbf{x}) = KL(\mathbf{y}, P(\mathbf{x})) + g(\mathbf{x}),$$

where $g(\mathbf{x}) \geq 0$ is some penalty function. De Pierro uses penalty functions $g(\mathbf{x})$ of the form

$$g(\mathbf{x}) = \sum_{l=1}^{p} f_l(\langle \mathbf{s}_l, \mathbf{x} \rangle).$$

Let us define the matrix $S$ to have for its $l$th row the vector $\mathbf{s}_l$. Then $\langle \mathbf{s}_l, \mathbf{x} \rangle = (S\mathbf{x})_l$, the $l$th entry of the vector $S\mathbf{x}$. Therefore,

$$g(\mathbf{x}) = \sum_{l=1}^{p} f_l((S\mathbf{x})_l).$$

Let $\lambda_{jl} > 0$ with $\sum_{j=1}^{J} \lambda_{jl} = 1$, for each $l$.

Assume that the functions $f_l$ are convex. Therefore, for each $l$, we have

$$f_l((S\mathbf{x})_l) = f_l(\sum_{j=1}^{J} S_{jl}x_j) = f_l(\sum_{j=1}^{J} \lambda_{jl}(S_{jl}/\lambda_{jl})\mathbf{x}_j)$$

$$\leq \sum_{j=1}^{J} \lambda_{jl} f_l((S_{jl}/\lambda_{jl})x_j).$$

Therefore

$$g(\mathbf{x}) \leq \sum_{l=1}^{p} \sum_{j} \lambda_{jl} f_l((S_{jl}/\lambda_{jl})x_j).$$

So we have replaced $g(\mathbf{x})$ with a related function in which the $x_j$ occur separately, rather than just in the combinations $(S\mathbf{x})_l$. But we aren't quite done yet.

We would like to take for De Pierro's $H(\mathbf{x}, \mathbf{z})$ the function used in the EMML algorithm, plus the function

$$\sum\nolimits_{l=1}^{p} \sum_{j=1}^{J} \lambda_{jl} f_l((S_{jl}/\lambda_{jl})z_j).$$

But there is one slight problem: we need $H(\mathbf{z}, \mathbf{z}) = F(\mathbf{z})$, which we don't have yet. De Pierro's clever trick is to replace $f_l((S_{jl}/\lambda_{jl})z_j)$ with

$$f_l((S_{jl}/\lambda_{jl})z_j - (S_{jl}/\lambda_{jl})x_j + (S\mathbf{x})_l).$$

So De Pierro's function $H(\mathbf{x}, \mathbf{z})$ is the sum of the $H(\mathbf{x}, \mathbf{z})$ used in the EMML case and the function

$$\sum\nolimits_{l=1}^{p} \sum_{j=1}^{J} \lambda_{jl} f_l((S_{jl}/\lambda_{jl})z_j - (S_{jl}/\lambda_{jl})x_j + (S\mathbf{x})_l).$$

Now he has the three properties he needs. Once he has computed $\mathbf{x}^k$ he minimizes $H(\mathbf{x}^k, \mathbf{z})$ by taking the gradient and solving the equations for the correct $\mathbf{z} = T\mathbf{x}^k = \mathbf{x}^{k+1}$. For the choices of $f_l$ he discusses these intermediate calculations can either be done in closed form (the quadratic case) or with a simple Newton-Raphson iteration (the logcosh case).

# Chapter 64

# Block-iterative algorithms

Iterative methods for reconstructing images have been studied for decades. Because many of these methods, such as the EMML, are slow to converge, particularly for the large data sets typical of modern imaging, there has been growing interest in block-iterative (also called ordered subset) methods for image reconstruction, due largely to the accelerated convergence some of these methods provide. A brief overview of the use of iterative reconstrction methods in medical imaging is given in [135]. The block-iterative methods of interest to us here can be derived as incremental optimization procedures, in which the cost function $h(\mathbf{x})$ to be minimized can be decomposed as a sum of simpler functions, $h(\mathbf{x}) = \sum_{i=1}^{I} h_i(\mathbf{x})$, and the iterative procedure involves the gradients of only a few of the $h_i(\mathbf{x})$ at each step.

Our topic is the reconstruction of a discrete image from finite data pertaining to that image. Because realistic models relating the data to the image pixels (or voxels) typically preclude closed form solutions, we shall focus here on iterative algorithms. For reasons to be presented shortly, the algorithms we shall consider are optimization methods, in which we seek to maximize or minimize some function over the set of feasible images, that is, those satisfying whatever constraints, such as nonnegativity, we have imposed .

When the data is essentially noise-free, but insufficient to determine a unique image, one may choose that feasible image consistent with the data, for which some function, such as entropy, is maximized, or some measure of image roughness or distance to a prior estimate of the image is minimized. When the data is noisy, there may be no feasible image consistent with the data. In such cases, one may choose to minimize a function that measures deviation from data consistency, with or without an additional regularizing term.

In typical image reconstruction situations both the data set and the

number of pixels or voxels to be determined are large; in addition, time considerations are important. The overall objective is the practical one of producing a useful reconstructed image quickly, rather than the more theoretical one of finding the solution of an optimization problem. Therefore iterative methods that produce fairly accurate reconstructed images in a short time are desired. For such practical reasons there has been growing interest in certain *block-iterative* or *ordered subset* methods [109], [32], [118], which provide the topic of this chapter.

Block-iterative methods are called *incremental* methods in the optimization literature [14]. The basic idea is as follows. Suppose that we wish to minimize a function $h: R^J \to (-\infty, +\infty)$. Iterative gradient methods would require us to calculate the gradient of $h$ at each step. If $h$ is the sum of a large number of simpler functions $h_i$ whose gradients are easier to calculate, so

$$h(\mathbf{x}) = \sum_{i=1}^{I} h_i(\mathbf{x}), \tag{64.1}$$

then at the $k$-th step we would need to compute

$$\nabla h(\mathbf{x}^k) = \sum_{i=1}^{I} \nabla h_i(\mathbf{x}^k). \tag{64.2}$$

For example, consider the least squares problem of finding a minimizer of the function $h(\mathbf{x}) = \frac{1}{2}||A\mathbf{x} - \mathbf{b}||^2$, where $A$ is any real $I$ by $J$ matrix. The gradient of $h$ is $\nabla h(\mathbf{x}) = A^T(A\mathbf{x} - \mathbf{b})$. We can put $h$ into the form of equation (64.1) using $h_i(\mathbf{x}) = \frac{1}{2}((A\mathbf{x})_i - b_i)^2$, which has for its gradient $\nabla h_i(\mathbf{x}) = ((A\mathbf{x})_i - b_i)\mathbf{a}^i$, where $\mathbf{a}^i$ is the $i$-th column of the matrix $A^T$.

To avoid computing the large sum in (64.2), we might consider using only those gradients $\nabla h_i$ whose indices $i$ belong to some predetermined *block* $B_n$, where $n$ depends on $k$; we assume throughout this paper that $\{B_1, ..., B_N\}$ denotes a partition of the set $\{i = 1, ..., I\}$ into disjoint subsets. We then proceed *incrementally*, using only these partial gradients to determine the direction to the next iterate. Stated this way, block-iterative methods appear to reduce computation at each step; but if the price we pay is to increase the number of steps needed to produce a good reconstructed image, we have gained nothing. Several of the block-iterative methods we shall discuss here do not require an increased number of steps, hence provide considerable time reduction in the reconstruction process.

When there is only a single block, that is $N = 1$, we say that the method is *simultaneous*. When each block contains only a single $i$, so there are $I$ blocks, we call the method *sequential* or *successive*; for problems involving the solution of matrix equations sequential methods have also been called *row-action* methods [58].

Because most of the functions we encounter in image reconstruction can be decomposed as in equation (64.1), obtaining block-iterative versions of iterative optimization algorithms is usually not difficult; but this is not enough. In order for a block-iterative method to be useful it must satisfy certain requirements. These requirements pertain to acceleration of convergence, as well as to the manner in which the method handles noise in the data.

Block-iterative methods are not new and the literature on the subject is extensive; see the book by Censor and Zenios [63] and the references therein.

# Chapter 65

# More on the ART

In this chapter we take a longer look at the algebraic reconstruction technique (ART). Both ART and its multiplicative version, MART, have block-iterative and simultaneous counterparts, which we shall discuss in subsequent chapters.

The ART is a procedure for solving the system of linear equations $A\mathbf{x} = \mathbf{b}$. Let $A$ be an $M$ by $N$ real matrix and for $m = 1, ..., M$ let $B_m = \{\mathbf{x}|(A\mathbf{x})_m = b_m\}$, where $b_m$ denotes the $m$-th entry of the vector $\mathbf{b}$. For notational convenience we shall assume in this chapter that $A$ has been normalized so that each of its rows has euclidean length one. Any solution of $A\mathbf{x} = \mathbf{b}$ lies in the intersection of the $B_m$; if the system is inconsistent then the intersection is empty. The Kaczmarz algorithm [122] for solving the system of linear equations $A\mathbf{x} = \mathbf{b}$ has the iterative step

$$\mathbf{x}_n^{k+1} = \mathbf{x}_n^k + A_{m(k)n}(b_{m(k)} - (A\mathbf{x}^k)_{m(k)}), \tag{65.1}$$

for $n = 1, ..., N$, $k = 0, 1, ...$ and $m(k) = k(\mathrm{mod}\, M) + 1$. This algorithm was rediscovered, in the context of medical imaging, by Gordon, Bender and Herman [102], who called it the *algebraic reconstruction technique* (ART). The ART algorithm is an example of the method of *successive orthogonal projections* (SOP) [105].

In the consistent case, in which the intersection of the hyperplanes $B_m$ is nonempty, the ART converges to that solution of $A\mathbf{x} = \mathbf{b}$ closest to the starting vector $\mathbf{x}^0$, as illustrated in Figure 65.1. The ART cannot converge in the inconsistent case, in which the intersection of the sets $B_m$ is empty, since the limit would then be a member of the (empty) intersection. Instead, the ART exhibits what is called *cyclic convergence*; that is, subsequences converge to finitely many distinct limits comprising a limit cycle [173], as illustrated in Figure 65.2. Once a member of this limit cycle is reached, further application of the algorithm results in passing from one member of the limit cycle to the next. Proving the existence of these limit

cycles is not as easy as it may seem. The proof given here is perhaps the most elementary. We assume throughout this chapter that the real $M$ by $N$ matrix $A$ has full rank and its rows have Euclidean length one.

## Some useful facts about the ART:

For $m = 1, 2, ..., M$ let $K_m = \{\mathbf{x} | (A\mathbf{x})_m = 0\}$ and $\mathbf{p}^m$ be the metric projection of $\mathbf{x} = \mathbf{0}$ onto $B_m$. Let $v_m^r = (A\mathbf{x}^{rM+m-1})_m$ and $\mathbf{v}^r = (v_1^r, ..., v_M^r)^T$, for $r = 0, 1, ....$ We begin with some basic facts.

**Exercise 1:** Establish the following facts concerning the ART.

Fact 1:

$$||\mathbf{x}^k||^2 - ||\mathbf{x}^{k+1}||^2 = (A(\mathbf{x}^k)_{m(k)})^2 - (b_{m(k)})^2.$$

Fact 2:

$$||\mathbf{x}^{rM}||^2 - ||\mathbf{x}^{(r+1)M}||^2 = ||\mathbf{v}^r||^2 - ||\mathbf{b}||^2.$$

Fact 3:

$$||\mathbf{x}^k - \mathbf{x}^{k+1}||^2 = ((A\mathbf{x}^k)_{m(k)} - b_{m(k)})^2.$$

Fact 4: There exists $B > 0$ such that, for all $r = 0, 1, ...,$ if $||\mathbf{v}^r|| \leq ||\mathbf{b}||$ then $||\mathbf{x}^{rM}|| \geq ||\mathbf{x}^{(r+1)M}|| - B$.

Fact 5: Let $\mathbf{x}^0$ and $\mathbf{y}^0$ be arbitrary and $\{\mathbf{x}^k\}$ and $\{\mathbf{y}^k\}$ the sequences generated by applying the ART algorithm. Then

$$||\mathbf{x}^0 - \mathbf{y}^0||^2 - ||\mathbf{x}^M - \mathbf{y}^M||^2 = \sum_{m=1}^{M} ((A\mathbf{x}^{m-1})_m - (A\mathbf{y}^{m-1})_m)^2.$$

## The system $A\mathbf{x} = \mathbf{b}$ is consistent:

In this subsection we give a proof of the following result.

**Theorem 65.1** *Let $A\hat{\mathbf{x}} = \mathbf{b}$ and let $\mathbf{x}^0$ be arbitrary. Let $\{\mathbf{x}^k\}$ be generated by equation (65.1). Then the sequence $\{||\hat{\mathbf{x}} - \mathbf{x}^k||\}$ is decreasing and $\{\mathbf{x}^k\}$ converges to the solution of $A\mathbf{x} = \mathbf{b}$ closest to $\mathbf{x}^0$.*

**Proof:** Let $A\hat{\mathbf{x}} = \mathbf{b}$. It follows from Fact 5 that the sequence $\{||\hat{\mathbf{x}} - \mathbf{x}^{rM}||\}$ is decreasing and the sequence $\{\mathbf{v}^r - \mathbf{b}\} \to 0$. So $\{\mathbf{x}^{rM}\}$ is bounded; let $\mathbf{x}^{*,0}$ be a cluster point. Then, for $m = 1, 2, ..., M$ let $\mathbf{x}^{*,m}$ be the successor of $\mathbf{x}^{*,m-1}$ using the ART algorithm. It follows that $(A\mathbf{x}^{*,m-1})_m = \mathbf{b}_m$ for each $m$, from which we conclude that $\mathbf{x}^{*,0} = \mathbf{x}^{*,m}$ for all $m$ and that $A\mathbf{x}^{*,0} = \mathbf{b}$. Using $\mathbf{x}^{*,0}$ in place of $\hat{\mathbf{x}}$, we have that $\{||\mathbf{x}^{*,0} - \mathbf{x}^k||\}$ is decreasing. But a subsequence converges to zero, so $\{\mathbf{x}^k\}$ converges to $\mathbf{x}^{*,0}$. By Fact 5 the difference $||\hat{\mathbf{x}} - \mathbf{x}^k||^2 - ||\hat{\mathbf{x}} - \mathbf{x}^{k+1}||^2$ is independent of which solution $\hat{\mathbf{x}}$ we pick; consequently, so is $||\hat{\mathbf{x}} - \mathbf{x}^0||^2 - ||\hat{\mathbf{x}} - \mathbf{x}^{*,0}||^2$. It follows that $\mathbf{x}^{*,0}$ is the solution closest to $\mathbf{x}^0$. This completes the proof. ∎

## The system $A\mathbf{x} = \mathbf{b}$ is inconsistent:

In the inconsistent case the sequence $\{\mathbf{x}^k\}$ will not converge, since any limit would be a solution. However, for each fixed $m \in \{1, 2, ..., M\}$, the subsequence $\{\mathbf{x}^{rM+m}\}$ converges [173]. Tanabe's proof relies heavily on results from linear algebra. The proof here is more elementary. We begin by establishing the following.

**Proposition 65.1** *The sequence $\{\mathbf{x}^{rM}\}$ is bounded.*

**Proof:** Assume that the sequence $\{\mathbf{x}^{rM}\}$ is unbounded. We first show that we can select a subsequence $\{\mathbf{x}^{r_jM}\}$ with the properties $||\mathbf{x}^{r_jM}|| \geq j$ and $||\mathbf{v}^{r_j}|| < ||\mathbf{b}||$, for $j = 1, 2, ....$

Assume that we have selected $\mathbf{x}^{r_jM}$, with the properties $||\mathbf{x}^{r_jM}|| \geq j$ and $||\mathbf{v}^{r_j}|| < ||\mathbf{b}||$; we show how to select $\mathbf{x}^{r_{j+1}M}$. Pick integer $t > 0$ such that

$$||\mathbf{x}^{tM}|| \geq ||\mathbf{x}^{r_jM}|| + B + 1,$$

where $B > 0$ is as in Fact 4. With $n + r_j = t$ let $i \geq 0$ be the smallest integer for which

$$||\mathbf{x}^{(r_j+n-i-1)M}|| < ||\mathbf{x}^{tM}|| \leq ||\mathbf{x}^{(r_j+n-i)M}||.$$

Then $||\mathbf{v}^{r_j+n-i-1}|| < ||\mathbf{b}||$. Let $\mathbf{x}^{r_{j+1}M} = \mathbf{x}^{(r_j+n-i-1)M}$. Then we have

$$||\mathbf{x}^{r_{j+1}M}|| \geq ||\mathbf{x}^{(r_j+n-i)M}|| - B \geq ||\mathbf{x}^{tM}|| - B \geq ||\mathbf{x}^{r_jM}|| + B + 1 - B \geq j + 1.$$

This gives us the desired subsequence.

For every $k = 0, 1, ...$ let $\mathbf{z}^{k+1} = \mathbf{x}^{k+1} - \mathbf{p}^{m(k)}$. Then $\mathbf{z}^{k+1} \in K_{m(k)}$. For $\mathbf{z}^{k+1} \neq 0$ let $\mathbf{u}^{k+1} = z^{k+1}/||\mathbf{z}^{k+1}||$. Since the subsequence $\{\mathbf{x}^{r_jM}\}$ is unbounded, so is $\{\mathbf{z}^{r_jM}\}$, so for sufficiently large $j$ the vectors $\mathbf{u}^{r_jM}$ are defined and on the unit sphere. Let $\mathbf{u}^{*,0}$ be a cluster point of $\{\mathbf{u}^{r_jM}\}$; replacing $\{\mathbf{x}^{r_jM}\}$ with a subsequence if necessary, assume that the sequence $\{\mathbf{u}^{r_jM}\}$ converges to $\mathbf{u}^{*,0}$. Then let $\mathbf{u}^{*,1}$ be a subsequence of $\{\mathbf{u}^{r_jM+1}\}$; again, assume the sequence $\{\mathbf{u}^{r_jM+1}\}$ converges to $\mathbf{u}^{*,1}$. Continuing in this

manner, we have $\{\mathbf{u}^{r_jM+i}\}$ converging to $\mathbf{u}^{*,i}$ for $i = 0, 1, 2, ....$ We know that $\{\mathbf{z}^{r_jM}\}$ is unbounded and since $||\mathbf{v}^{r_j}|| < ||\mathbf{b}||$, we have, by Fact 3, that $\{\mathbf{z}^{r_jM+m-1} - \mathbf{z}^{r_jM+m}\}$ is bounded for each $m$. Consequently $\{\mathbf{z}^{r_jM+m}\}$ is unbounded for each $m$.

Now we have

$$||\mathbf{z}^{r_jM+m-1} - \mathbf{z}^{r_jM+m}||$$

$$\geq ||\mathbf{z}^{r_jM+m-1}||\,||\mathbf{u}^{r_jM+m-1} - \langle \mathbf{u}^{r_jM+m-1}, \mathbf{u}^{r_jM+m}\rangle \mathbf{u}^{r_jM+m}||.$$

Since the left side is bounded and $||\mathbf{z}^{r_jM+m-1}||$ has no infinite bounded subsequence, we conclude that

$$||\mathbf{u}^{r_jM+m-1} - \langle \mathbf{u}^{r_jM+m-1}, \mathbf{u}^{r_j+M+m}\rangle \mathbf{u}^{r_jM+m}|| \to 0.$$

It follows that $\mathbf{u}^{*,0} = \mathbf{u}^{*,m}$ or $\mathbf{u}^{*,0} = -\mathbf{u}^{*,m}$ for each $m = 1, 2, ..., M$. Therefore $\mathbf{u}^{*,0}$ is in $K_m$ for each $m$; since the null space of $A$ contains only zero, this is a contradiction. This completes the proof of the proposition. ∎

Now we give a proof of the following result.

**Theorem 65.2** *Let $A$ be $M$ by $N$, with $M > N$ and $A$ with full rank. If $A\mathbf{x} = \mathbf{b}$ has no solutions, then, for any $\mathbf{x}^0$ and each fixed $m \in \{0, 1, ..., M\}$, the subsequence $\{\mathbf{x}^{rM+m}\}$ converges to a limit $\mathbf{x}^{*,m}$. Beginning the iteration in equation (65.1) at $\mathbf{x}^{*,0}$, we generate the $\mathbf{x}^{*,m}$ in turn, with $\mathbf{x}^{*,M} = \mathbf{x}^{*,0}$.*

**Proof:** Let $\mathbf{x}^{*,0}$ be a cluster point of $\{\mathbf{x}^{rM}\}$. Beginning the ART algorithm at $\mathbf{x}^{*,0}$ we obtain $\mathbf{x}^{*,i}$, for $i = 0, 1, 2, ....$ It is easily seen that

$$||\mathbf{x}^{(r-1)M} - \mathbf{x}^{rM}||^2 - ||\mathbf{x}^{rM} - \mathbf{x}^{(r+1)M}||^2$$

$$= \sum_{m=1}^{M} ((A\mathbf{x}^{(r-1)M+m-1})_m - (A\mathbf{x}^{rM+m-1})_m)^2.$$

Therefore the sequence $\{||\mathbf{x}^{(r-1)M} - \mathbf{x}^{rM}||\}$ is decreasing and

$$\{\sum_{m=1}^{M} ((A\mathbf{x}^{(r-1)M+m-1})_m - (A\mathbf{x}^{rM+m-1})_m)^2\} \to 0.$$

Therefore $(A\mathbf{x}^{*,m-1})_m = (A\mathbf{x}^{*,M+m-1})_m$ for each $m$.

For arbitrary $\mathbf{x}$ we have

$$||\mathbf{x} - \mathbf{x}^{*,0}||^2 - ||\mathbf{x} - \mathbf{x}^{*,M}||^2$$

$$= \sum_{m=1}^{M} ((A\mathbf{x})_m - (A\mathbf{x}^{*,m-1})_m)^2 - \sum_{m=1}^{M} ((A\mathbf{x})_m - b_m)^2,$$

so that

$$||\mathbf{x} - \mathbf{x}^{*,0}||^2 - ||\mathbf{x} - \mathbf{x}^{*,M}||^2 = ||\mathbf{x} - \mathbf{x}^{*,M}||^2 - ||\mathbf{x} - \mathbf{x}^{*,2M}||^2.$$

Using $\mathbf{x} = \mathbf{x}^{*,M}$ we have

$$||\mathbf{x}^{*,M} - \mathbf{x}^{*,0}|| = -||\mathbf{x}^{*,M} - \mathbf{x}^{*,2M}||,$$

from which we conclude that $\mathbf{x}^{*,0} = \mathbf{x}^{*,M}$. From Fact 5 it follows that the sequence $\{||\mathbf{x}^{*,0} - \mathbf{x}^{rM}||\}$ is decreasing; but a subsequence converges to zero, so the entire sequence converges to zero and $\{\mathbf{x}^{rM}\}$ converges to $\mathbf{x}^{*,0}$. This completes the proof. ∎

## Avoiding the limit cycle behavior:

The greater the minimum value of $||A\mathbf{x} - \mathbf{b}||^2$ the more the vectors of the LC are distinct from one another. There are several ways to avoid the LC in ART and to obtain a least squares solution. One way is the *double ART* (DART) [36]:

**The DART:** We know that any $\mathbf{b}$ can be written as $\mathbf{b} = A\hat{\mathbf{x}} + \hat{\mathbf{w}}$, where $A^T\hat{\mathbf{w}} = \mathbf{0}$ and $\hat{\mathbf{x}}$ is a minimizer of $||A\mathbf{x} - \mathbf{b}||^2$. The vector $\hat{\mathbf{w}}$ is the orthogonal projection of $\mathbf{b}$ onto the null space of the matrix transformation $A^T$. Therefore, in Step 1 of DART we apply the ART algorithm to the consistent system of linear equations $A^T\mathbf{w} = 0$, beginning with $\mathbf{w}^0 = \mathbf{b}$. The limit is $\mathbf{w}^\infty = \hat{\mathbf{w}}$, the member of the null space of $A^T$ closest to $\mathbf{b}$. In Step 2, apply ART to the consistent system of linear equations $A\mathbf{x} = \mathbf{b} - \mathbf{w}^\infty = A\hat{\mathbf{x}}$. The limit is then the minimizer of $||A\mathbf{x} - \mathbf{b}||$ closest to $\mathbf{x}^0$.

Another method for avoiding the LC is *strong underrelaxation* [60].

**Strongly underrelaxed ART:** Let $t > 0$. Replace the iterative step in ART with

$$\mathbf{x}_j^{k+1} = \mathbf{x}_j^k + tA_{ij}\frac{(b_i - (A\mathbf{x}^k)_i)}{\sum_{l=1}^J A_{il}^2}. \tag{65.2}$$

In [60] it is shown that, as $t \to 0$, the vectors of the LC approach the geometric least squares solution closest to $\mathbf{x}^0$. Bertsekas [14] uses strong underrelaxation to obtain convergence of more general incremental methods.

## Regularizing ART:

It is often the case that the entries of the vector $\mathbf{b}$ in the system $A\mathbf{x} = \mathbf{b}$ come from measurements, so are usually noisy. If the entries of $\mathbf{b}$ are noisy

but the system $A\mathbf{x} = \mathbf{b}$ remains consistent (which can easily happen in the underdetermined case, with $N > M$) the ART begun at $\mathbf{x}^0 = 0$ converges to the solution having minimum norm, but this norm can be quite large. The resulting solution is probably useless. Instead of solving $A\mathbf{x} = \mathbf{b}$ we *regularize* by minimizing, for example, the function

$$||A\mathbf{x} - \mathbf{b}||^2 + \epsilon^2||\mathbf{x}||^2, \tag{65.3}$$

for some small $\epsilon^2$. The solution to this problem is the vector $\mathbf{x}$ for which

$$(A^T A + \epsilon^2 I)\mathbf{x} = A^T\mathbf{b}. \tag{65.4}$$

However, we do not want to have to calculate $A^T A$, particularly when the matrix $A$ is large.

We discuss two methods for using ART to obtain regularized solutions of $A\mathbf{x} = \mathbf{b}$. The first one is new, the second one is due to Eggermont, Herman and Lent [88].

In our first method we use ART to solve the system of equations given in matrix form by

$$[\, A^T \quad \epsilon I \,] \begin{bmatrix} \mathbf{u} \\ \mathbf{v} \end{bmatrix} = 0.$$

We begin with $\mathbf{u}^0 = \mathbf{b}$ and $\mathbf{v}^0 = 0$. The lower component of the limit vector is then $\mathbf{v}^\infty = -\epsilon\hat{\mathbf{x}}$, where $\hat{\mathbf{x}}$ minimizes the function in (65.3).

The method of Eggermont *et al* is similar. In his method we use ART to solve the system of equations given in matrix form by

$$[\, A \quad \epsilon I \,] \begin{bmatrix} \mathbf{x} \\ \mathbf{v} \end{bmatrix} = \mathbf{b}.$$

We begin at $\mathbf{x}^0 = 0$ and $\mathbf{v}^0 = 0$. Then the limit vector has for its upper component $\mathbf{x}^\infty = \hat{\mathbf{x}}$ as before. Also $\epsilon\mathbf{v}^\infty = \mathbf{b} - A\hat{\mathbf{x}}$.

As Herman and Meyer have shown [109], the order in which the equations are accessed in ART, as well as the use of relaxation parameters, can greatly affect the speed of convergence. The main consideration is to avoid taking the equations in an order such that each equation substantially repeats the information about the image present in the previous equation. To avoid such a situation we could employ a random ordering of the equations, although more carefully designed ordering may achieve somewhat faster convergence.

There are several interesting questions we can ask about the behavior of the ART in the inconsistent case, some of which are, I believe, unanswered.

## Where is the least squares solution?

When the system $A\mathbf{x} = \mathbf{b}$ has no exact solutions we could seek instead the *least squares* solution $\hat{\mathbf{x}}$ satisfying

$$A^T A\hat{\mathbf{x}} = A^T\mathbf{b}.$$

But suppose we do not know if the system has exact solutions. We do the ART and then discover, after convergence to a limit cycle, that $A\mathbf{x} = \mathbf{b}$ has no solutions. What can we do then? Is there a simple way to compute the least squares solution from the limit cycle vectors? More generally, where is the least squares solution, in relation to the vectors of the limit cycle? The following partial answer was presented in [33].

**Theorem 65.3** *Let $M = N + 1$. If the system of equations $A\mathbf{x} = \mathbf{b}$ has no solution then the vectors of the ART limit cycle lie on a sphere in $R^N$ centered at the least squares solution.*

**Proof:** Let the vectors of the limit cycle be $\{\mathbf{z}^1, \mathbf{z}^2, ..., \mathbf{z}^M = \mathbf{z}^0\}$ and let the vector $\mathbf{c}$ have the entries $\mathbf{c}_m = (A\mathbf{z}^{m-1})_m$, for $m = 1, 2, ..., M$. We then have

$$\mathbf{z}_n^m - \mathbf{z}_n^{m-1} = A_{mn}(b_m - c_m)$$

for each $m$ and $n$. Summing over $m = 1, ..., M$ on both sides and using the fact that $\mathbf{z}^M = \mathbf{z}^0$, we get zero on the left side, for each $n$. It follows then that

$$A^T\mathbf{b} = A^T\mathbf{c}.$$

Therefore the systems $A\mathbf{x} = \mathbf{b}$ and $A\mathbf{x} = \mathbf{c}$ have the same least squares solution $\hat{\mathbf{x}}$. This means that we can write

$$\mathbf{b} = A\hat{\mathbf{x}} + \mathbf{v}$$

and

$$\mathbf{c} = A\hat{\mathbf{x}} + \mathbf{w},$$

where $A^T\mathbf{v} = A^T\mathbf{w} = \mathbf{0}$. In addition, we have

$$||\mathbf{b}||^2 = ||A\hat{\mathbf{x}}||^2 + ||\mathbf{v}||^2$$

and

$$||\mathbf{c}||^2 = ||A\hat{\mathbf{x}}||^2 + ||\mathbf{w}||^2.$$

It is easy to show that

$$||\hat{\mathbf{x}} - \mathbf{z}^m||^2 - ||\hat{\mathbf{x}} - \mathbf{z}^{m-1}||^2 = v_m^2 - w_m^2,$$

as well as

$$||\mathbf{z}^m||^2 - ||\mathbf{z}^{m-1}||^2 = b_m^2 - c_m^2.$$

for each $m$. Again summing over $m$ on both sides of the latter equation, we get zero on the left and $||\mathbf{b}||^2 - ||\mathbf{c}||^2$ on the right. It follows that $||\mathbf{v}|| = ||\mathbf{w}||$. Both $\mathbf{v}$ and $\mathbf{w}$ are in the null space of the matrix $A^T$. Since $M = N + 1$ and $A$ is assumed to have full rank, the null space of $A^T$ has dimension one. Consequently $\mathbf{v} = \mathbf{w}$ or $\mathbf{v} = -\mathbf{w}$. The first choice is

out, since that implies that $\mathbf{z}^1 = \mathbf{z}^2 = ... = \mathbf{z}^M$, which means the system $A\mathbf{x} = \mathbf{b}$ is consistent, with solution $\mathbf{z}^1$. So we must conclude that $\mathbf{v} = -\mathbf{w}$. But this says

$$||\hat{\mathbf{x}} - \mathbf{z}^m||^2 - ||\hat{\mathbf{x}} - \mathbf{z}^{m-1}||^2 = 0.$$

Since this holds for any $m$ the proof of the theorem is complete. ∎

It is curious that this result holds only sometimes when the condition $M = N+1$ is violated. An interesting question that has not been answered is: What is the radius of this sphere? As far as I know, this theorem has not been extended to the general case.

## A quick side trip to Euclidean geometry:

The theorem above has an interesting connection to a not very well known theorem in plane euclidean geometry. It is well known that the medians of a triangle are concurrent, as are the angle bisectors. The *symmedian lines*, formed by reflecting the medians in the angle bisectors, are also concurrent, their common point being the *Grebe-Lemoine point*, also called the *symmedian point* [121]. The symmedian point can be shown to be that point in the plane such that the sum of the squares of the distances from the point to the three sides of the triangle is minimized.

**Exercise 2:** Connect this result with our theorem above.

## Another look at the least squares solution:

One reason why the system of equations $A\mathbf{x} = \mathbf{b}$ can fail to have a solution when $M > N$ is that there are not enough unknowns. Suppose we augment the vector of unknowns $\mathbf{x}$ by concatenating an $M-N$ by 1 vector $\mathbf{y}$, forming the $M$ by $M$ vector $\mathbf{z} = [\mathbf{x}^T \ \mathbf{y}^T]^T$. Similarly, augment the $M$ by $N$ matrix $A$ by adding $M - N$ new columns to get $C = [A \ B]$.

**Exercise 3:** Show that if we select $B$ so that $C$ is invertible and $B^T A = 0$ then the exact solution of $C\mathbf{z} = \mathbf{b}$ is the concatenation of the least squares solutions of $A\mathbf{x} = \mathbf{b}$ and $B\mathbf{y} = \mathbf{b}$.

## Nonnegatively constrained least squares:

Consider the problem of minimizing the function $||A\mathbf{x} - \mathbf{b}||$, subject to the constraints $\mathbf{x}_n \geq 0$ for all $n$. We can solve this problem using a slight modification of the ART: at each step of the iteration, if the $n$-th entry of the vector $\mathbf{x}^{k+1}$ given by the ART is nonnegative we accept it; if it is not, we replace it with zero. Although there may be multiple solutions $\hat{\mathbf{x}}$, we know, at least, that $A\hat{\mathbf{x}}$ is the same for all solutions.

According to the Karush-Kuhn-Tucker theorem [155] the vector $A\hat{\mathbf{x}}$ must satisfy the condition

$$\sum_{m=1}^{M} A_{mn}(A\hat{\mathbf{x}}_m - b_m) = 0 \tag{65.5}$$

for all $n$ for which $\hat{\mathbf{x}}_n > 0$ for some solution $\hat{\mathbf{x}}$. Let $S$ be the set of all indices $n$ for which there exists a solution $\hat{\mathbf{x}}$ with $\hat{\mathbf{x}}_n > 0$. Then equation (65.5) must hold for all $n$ in $S$. Let $Q$ be the matrix obtained from $A$ by deleting those columns whose index $n$ is not in $S$. Then $Q^T(A\hat{\mathbf{x}} - \mathbf{b}) = 0$. If $Q$ has full rank and the cardinality of $S$ is greater than or equal to $M$, then $Q^T$ is one-to-one and $A\hat{\mathbf{x}} = \mathbf{b}$. We have proven the following result:

**Theorem 65.4** *Suppose that $A$ and every matrix $Q$ obtained from $A$ by deleting columns has full rank. Suppose there is no nonnegative solution of the system of equations $A\mathbf{x} = \mathbf{b}$. Then there is a subset $S$ of the set $\{n = 1, 2, ..., N\}$ with cardinality at most $M - 1$ such that, if $\hat{\mathbf{x}}$ is any minimizer of $||A\mathbf{x} - \mathbf{b}||$ subject to $\mathbf{x} \geq 0$, then $\hat{x}_n = 0$ for $n$ not in $S$. Therefore $\hat{\mathbf{x}}$ is unique.*

When $\hat{\mathbf{x}}$ is a vectorized two-dimensional image and $N > M$ the presence of at most $M-1$ positive pixels makes the resulting image resemble stars in the sky; for that reason this theorem and the related result for the EMML algorithm are sometimes called *night sky* theorems.

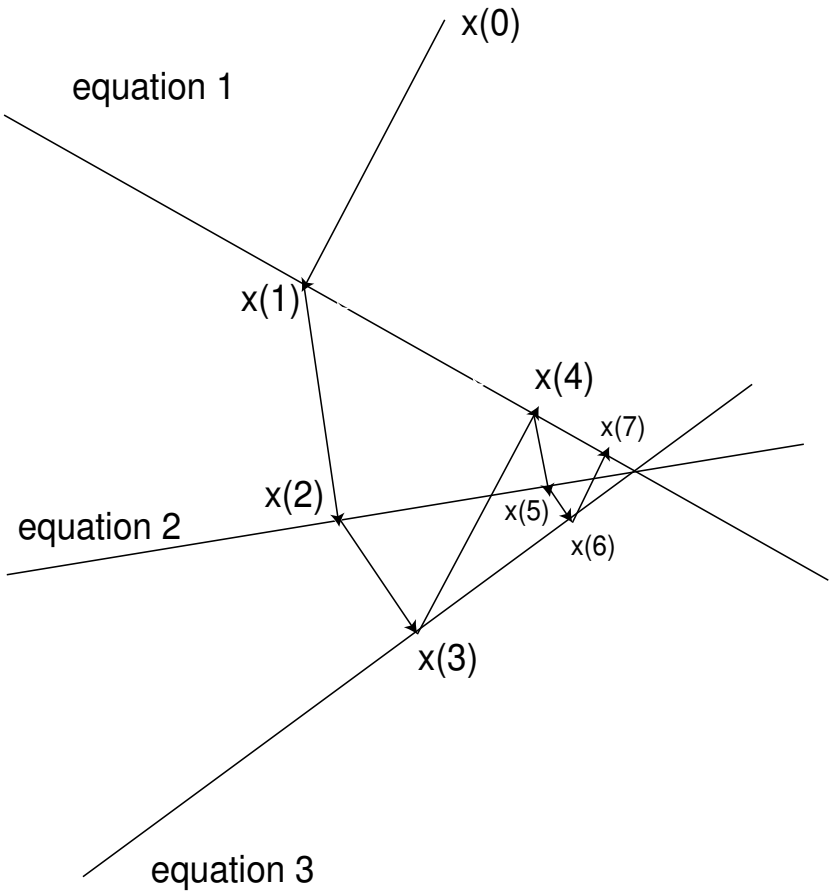



Figure 65.1: The ART algorithm in the consistent case.

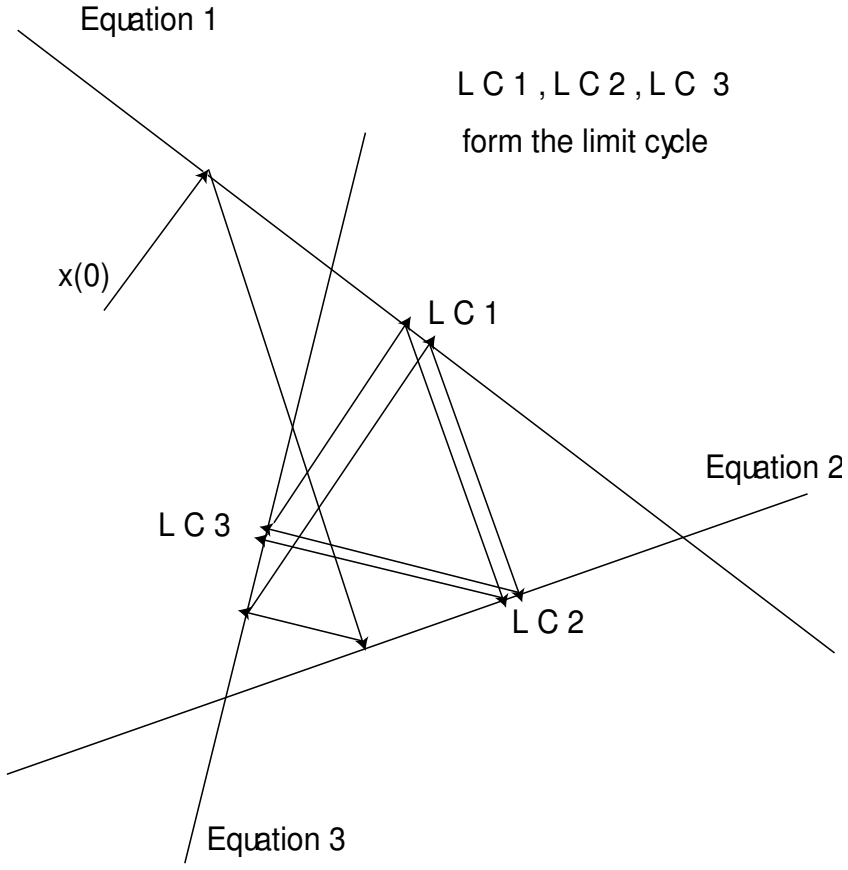



Figure 65.2: The ART algorithm in the inconsistent case.

# Chapter 66

# Methods related to the ART

The ART algorithm for solving the system $A\mathbf{x} = \mathbf{b}$ can be modified to include relaxation and regularization. There are also block-iterative and simultaneous versions of the ART. For example, we can introduce relaxation in ART using the *relaxed* ART (REART):

**The REART:**

$$x_j^{k+1} = x_j^k + \gamma_k A_{ij} \frac{(b_i - (A\mathbf{x}^k)_i)}{\sum_{l=1}^J A_{il}^2}, \tag{66.1}$$

with $\gamma_k$ positive scalars.

A simultaneous version of the ART was introduced by Cimmino [69]. It is obtained by projecting orthogonally onto each hyperplane simultaneously, then averaging the result. In closed form the Cimmino method is the following:

**Cimmino's method:** For $k = 0, 1, ...$ let

$$x_j^{k+1} = x_j^k + \frac{1}{I} \sum_{i=1}^I A_{ij} \frac{(b_i - (A\mathbf{x}^k)_i)}{\sum_{l=1}^J A_{il}^2}; \tag{66.2}$$

with

$$G_{ij} = A_{ij} / (\sum_{l=1}^J A_{il}^2)^{1/2} \tag{66.3}$$

and

$$c_i = b_i / (\sum_{l=1}^J A_{il}^2)^{1/2}, \tag{66.4}$$

the iteration in equation (66.2) becomes

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \frac{1}{I} G^T(\mathbf{c} - G\mathbf{x}^k). \tag{66.5}$$

Clearly the Cimmino method is a special case of the Landweber iterative method given in equation (37.2).

Cimmino's method can also employ relaxation: using positive relaxation parameters $\gamma_k$ in place of $\frac{1}{I}$ we get

**The relaxed Cimmino method:**

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \gamma_k G^T(\mathbf{c} - G\mathbf{x}^k). \tag{66.6}$$

The convergence proof for the relaxed Cimmino method requires that the relaxation parameters satisfy the inequality $0 < \gamma_k < 2/L$, where $L$ is the largest eigenvalue of the matrix $G^T G$. Since the trace of $GG^T$ is $I$, we know that $L \leq I$. This is a quite conservative estimate, in most cases, particularly if the matrix $A$ is sparse. Let $s_j$ be the number of nonzero entries in the $j$-th column of $A$ and let $s$ be the maximum of the $s_j$. As we showed in an earlier chapter, $L \leq s$, which says that the relaxed Cimmino method converges with $\gamma_k = \frac{1}{s}$. To illustrate, suppose that $s = I^{1/2}$. Then the factor $I^{-1}$ in Cimmino can be replaced with $I^{-1/2}$, which significantly accelerates convergence. We can obtain additional acceleration by passing to a block-iterative version of ART.

Because the computations in Cimmino can be performed simultaneously, the Cimmino method has the advantage of being parallelizable. In practice, it might be more efficient for only a subset of these computations to be performed simultaneously. In that case, block-iterative versions of ART would be more appropriate. We consider those now.

We can obtain a block-iterative version of ART (BI-ART) by partitioning the collection of hyperplanes into finitely many subsets or blocks and then projecting orthogonally onto each hyperplane in the current block and averaging the result. Then a new current block is selected and the process repeated. For $n = 1, ..., N$ let $I_n$ be the cardinality of the block $B_n$.

The *block-iterative* ART (BI-ART) has the following iterative step:

**The BI-ART:** For $k = 0, 1, ...$ and $n = n(k) = k(\mathrm{mod}N) + 1$ let

$$x_j^{k+1} = x_j^k + \frac{1}{I_n} \sum_{i \in B_n} A_{ij} \frac{(b_i - (A\mathbf{x}^k)_i)}{\sum_{l=1}^J A_{il}^2}. \tag{66.7}$$

Obtain the matrix $G_n$ from $G$ in equation (66.3) by removing the $i$-th row of $G$ for those $i$ not in $B_n$. Similarly, obtain vector $\mathbf{c}^n$ from $\mathbf{c}$ in equation (66.4). Then the iteration in equation (66.7) becomes

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \frac{1}{I_n} G_n^T(\mathbf{c}^n - G_n\mathbf{x}^k). \tag{66.8}$$

Relaxed BI-ART (RE-BI-ART) employs positive relaxation parameters $\gamma_n$ in place of $\frac{1}{I_n}$:

**The RE- BI-ART:** For $k = 0, 1, ...$ and $n = n(k) = k(\mathrm{mod}N) + 1$ let

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \gamma_n G_n^T(\mathbf{c}^n - G_n\mathbf{x}^k), \tag{66.9}$$

where $\gamma_n \in (0, 1/L_n)$ for $L_n$ the largest eigenvalue of the matrix $G_n^T G_n$. Concerning the RE-BI-ART we have the following theorem.

**Theorem 66.1** *Let $k$ and $n = n(k)$ be fixed and let $G\mathbf{x} = \mathbf{c}$. Then*

$$||\mathbf{x} - \mathbf{x}^k||^2 - ||\mathbf{x} - \mathbf{x}^{k+1}||^2 \geq \gamma_n||\mathbf{c}^n - G_n\mathbf{x}^k||^2. \tag{66.10}$$

This result follows by expanding the norms in terms of inner products and using the Cauchy inequality.

When there are solutions of $G\mathbf{x} = \mathbf{c}$ the RE-BI-ART converges to the solution closest to the starting point $\mathbf{x}^0$. The inequality in (66.10) is the key to the convergence proof. The same trace argument as used earlier tells us that $L_n \leq I_n$. If $G_n$ is sparse, we can do much better. For fixed $n$ and $j$, let $s_{nj}$ be the number of nonzero entries in the $j$-th column of the matrix $G_n$; let $s_n$ be the maximum of the $s_{nj}$. Then we have $L_n \leq s_n$, so that the factor $1/I_n$ in equation (66.7) can be replaced by the factor $1/s_n$. This can lead to significant acceleration of convergence.

Suppose, for the sake of illustration, that each column of the matrix $G$ has $s$ nonzero entries and that $r = s/I$ is the proportion of nonzero entries in any column. Suppose that $I_n = I/N$ for each $n$. If $N$ is not too large, we would expect $s_{nj}$ to be nearly equal to $rI_n = rI/N$, for each $j$ and $n$; then $s_n$ is nearly $rI/N = \frac{s}{I}\frac{I}{N} = \frac{s}{N}$. So the factor $\frac{1}{I}$ in Cimmino is replaced by $\frac{s}{N}$ in RE-BI-ART. But, unless $s_n = 0$, which means the matrix $G_n$ is the zero matrix, we have $s_n \geq 1$, regardless of the size of $N$. So the factor $1/s_n$ is never larger than one, which is the factor used in unrelaxed ART. For a given value of $s$, we need to use approximately $N = s$ blocks to have $s_n$ nearly equal to one. Therefore, the more sparse the matrix is, the fewer blocks we need to use for the factor $1/s_n$ to attain its maximum value. For very sparse matrices, few blocks are needed, allowing for a high degree of parallelization, since, within each block, the computation is simultaneous.

When there are solutions of the system $G\mathbf{x} = \mathbf{c}$ then ART, BI-ART and Cimmino methods converge to the solution of $A\mathbf{x} = \mathbf{b}$ closest to the initial vector $\mathbf{x}^0$, according to the Euclidean distance. In addition, when there are no solutions of $A\mathbf{x} = \mathbf{b}$ Cimmino converges to the geometric least squares solution, the minimizer of $||G\mathbf{x} - \mathbf{c}||$ closest to $\mathbf{x}^0$, while ART and BI-ART fail to converge. Instead, as Tanabe has shown [173], for each fixed $i$, as $m \to +\infty$, the ART subsequences $\{\mathbf{x}^{mI+i}\}$ converge to (usually $I$) distinct vectors $\mathbf{x}^{\infty,i}$; we call this set of vectors the *limit cycle* (LC). The

greater the minimum value of $||G\mathbf{x} - \mathbf{c}||^2$ the more the vectors of the LC are distinct from one another. An analogous result holds for RE-BI-ART.

In practical situations, one may use only a few iterations of an algorithm and be less concerned with the limiting vector (or vectors) than with the behavior of the iterates for small values of $k$. When the minimum value of $||A\mathbf{x} - \mathbf{b}||^2$ is not too large (that is, the measured data is not too noisy), the ART has been shown to provide usable reconstructions with very few iterations, particularly when the equations are carefully ordered and some amount of underrelaxation is used [109]. In contrast, the Cimmino method can be quite slow to converge.

It is important to note that acceleration of convergence need not require passing from a simultaneous method to a block-iterative method. The example of Cimmino's method and BI-ART in the case of a sparse matrix $A$ shows that part of the reason why Cimmino's method is slow is that it does not employ an appropriate relaxation parameter. If we know a good upper bound on the eigenvalues of $G^TG$ then we can improve Cimmino by using relaxation with better values of $\gamma_k$. If we have no *a priori* estimate, we could begin with $\gamma_k = 1/I$ and begin to lower the $\gamma_k$ as the iteration proceeds, checking for divergence. In the sparse case, as we have seen, we can get significant acceleration with relaxed Cimmino by making use of the degree of sparseness of the matrix $G$.

# Chapter 67

# The MART and related methods

Related to the ART is the *multiplicative* ART (MART), also due to Gordon, Bender and Herman [102]. While the ART applies to arbitrary systems of linear equations, the MART is restricted to a system of linear equations $\mathbf{y} = P\mathbf{x}$, in which the $I$ by $J$ matrix $P$ has nonnegative entries, the entries of $\mathbf{y}$ are positive and $\mathbf{x}$ has nonnegative entries; we shall also assume, for notational convenience, that the columns of $P$ sum to one, although that is not necessary. The MART and its block-iterative versions, BI-MART, converge to nonnegative solutions of $\mathbf{y} = P\mathbf{x}$, whenever such solutions exist. The block-iterative version involving only a single block is the simultaneous MART (SMART), which also converges to an approximate solution when no nonnegative solution of $\mathbf{y} = P\mathbf{x}$ exists.

The function minimized by the SMART is $h(\mathbf{x}) = KL(P\mathbf{x}, \mathbf{y})$; here $KL(\mathbf{u}, \mathbf{v})$ is the Kullback-Leibler (or cross-entropy) distance, defined for nonnegative vectors $\mathbf{u}$ and $\mathbf{v}$ by

$$KL(\mathbf{u}, \mathbf{v}) = \sum_{m=1}^{M} KL(u_m, v_m), \tag{67.1}$$

where $KL(a, b) = a \log \frac{a}{b} + b - a$, $KL(0, b) = b$ and $KL(a, 0) = +\infty$ for positive scalars $a$ and $b$. With $h_i(\mathbf{x}) = KL((P\mathbf{x})_i, y_i)$ we see that $h$ has the decomposition given by equation (64.1).

The MART algorithm is the following:

**The MART:** The *multiplicative algebraic reconstruction technique* (MART) [102] begins with a strictly positive vector $\mathbf{x}^0$ and has the iterative step

$$x_j^{k+1} = x_j^k \Big( \frac{y_i}{(P\mathbf{x}^k)_i} \Big)^{P_{ij}}, \tag{67.2}$$

for $j = 1, 2, ..., J$ and $i = k(\mathrm{mod}\, I)+1$. The simultaneous MART (SMART) algorithm is then

**The SMART:** The *simultaneous* MART (SMART) begins with a strictly positive vector $\mathbf{x}^0$ and has the iterative step

$$x_j^{k+1} = x_j^k \prod_{i=1}^{I} \Big(\frac{y_i}{(P\mathbf{x}^k)_i}\Big)^{P_{ij}}, \tag{67.3}$$

for $j = 1, 2, ..., J$. This algorithm was discovered independently in 1972, in statistics by Darroch and Ratcliff [80] [77] and in medical imaging by Schmidlin [162],[116]. It was discussed as a simultaneous version of MART in [62] and convergence in the inconsistent case was demonstrated in [29], where the algorithm was called the SMART.

The block-iterative SMART (BI-SMART) is as follows:

**The BI-SMART:** The *block-iterative* SMART (BI-SMART) [32] begins with a strictly positive vector $\mathbf{x}^0$ and has the iterative step

$$x_j^{k+1} = x_j^k \prod_{i\in B_n} \Big(\frac{y_i}{(P\mathbf{x}^k)_i}\Big)^{P_{ij}}, \tag{67.4}$$

for $j = 1, 2, ..., J$ and $n = k(\mathrm{mod}\, N) + 1$. Clearly, MART and SMART are special cases of the BI-SMART method. We introduce relaxation into the BI-SMART as follows:

**The relaxed BI-SMART:** The relaxed BI-SMART begins with a strictly positive vector $\mathbf{x}^0$ and has the iterative step

$$x_j^{k+1} = x_j^k \prod_{i\in B_n} \Big(\frac{y_i}{(P\mathbf{x}^k)_i}\Big)^{\gamma_k P_{ij}}, \tag{67.5}$$

for $j = 1, 2, ..., J$ and $n = k(\mathrm{mod}\, N) + 1$.

In the consistent case, that is, when there are vectors $\mathbf{x} \geq 0$ with $\mathbf{y} = P\mathbf{x}$, BI-SMART converges to the nonnegative solution that minimizes $KL(\mathbf{x}, \mathbf{x}^0)$. When there are no such nonnegative vectors, the SMART converges to the unique nonnegative minimizer of $KL(P\mathbf{x}, \mathbf{y})$ for which $KL(\mathbf{x}, \mathbf{x}^0)$ is minimized (see [29]); for $N > 1$, the BI-SMART fails to converge. What is always observed, but for which no proof exists, is that, for each fixed $i = 1, 2, ..., I$, as $m \to +\infty$, the MART subsequences $\{\mathbf{x}^{mI+i}\}$ converge to separate limit vectors, say $\mathbf{x}^{\infty,i}$. This *limit cycle* LC $= \{\mathbf{x}^{\infty,i} | i = 1, ..., I\}$ reduces to a single vector whenever there is a nonnegative solution of $\mathbf{y} = P\mathbf{x}$. The greater the minimum value of $KL(P\mathbf{x}, \mathbf{y})$ the more distinct from one another the vectors of the limit cycle are. An analogous result is observed for BI-SMART.

The MART will converge, in the consistent case, provided that $0 \leq P_{ij} \leq 1$, for all $i$ and $j$; this condition holds here since we have assumed that the columns of $P$ sum to one. Since $I$ is typically quite large, the $P_{ij}$ are likely to be a great deal smaller than one. We can accelerate the convergence of MART by rescaling the equations, obtaining what we have called the REMART.

**The REMART:** The *rescaled multiplicative algebraic reconstruction technique* (REMART) [32] begins with a strictly positive vector $\mathbf{x}^0$ and has the iterative step

$$x_j^{k+1} = x_j^k \Big( \frac{y_i}{(P\mathbf{x}^k)_i} \Big)^{m_i^{-1} P_{ij}}, \tag{67.6}$$

for $j = 1, 2, ..., J$ and $i = k(\mathrm{mod}\, I) + 1$, with $m_i = \max\{P_{ij} | j = 1, ..., J\}$.

Although the importance of the rescaling for accelerating MART is not remarked upon in papers on MART, the rescaling was often a part of actual implementations [108].

Similarly, the BI-SMART will converge, in the consistent case, provided that $0 \leq \sum_{i \in B_n} P_{ij} \leq 1$, for all $n$ and $j$; this condition holds here since we have assumed that the columns of $P$ sum to one. Since $N$ may be large, the $\sum_{i \in B_n} P_{ij}$ are likely to be a great deal smaller than one. We can accelerate the convergence of BI-SMART by rescaling the equations, obtaining what we have called the *rescaled block-iterative* SMART (RBI-SMART).

**The RBI-SMART:** The *rescaled block-iterative* SMART (RBI-SMART) [32] begins with a strictly positive vector $\mathbf{x}^0$ and has the iterative step

$$x_j^{k+1} = x_j^k \prod_{i \in B_n} \Big( \frac{y_i}{(P\mathbf{x}^k)_i} \Big)^{m_n^{-1} P_{ij}}, \tag{67.7}$$

for $j = 1, 2, ..., J$ and $n = k(\mathrm{mod}\, N) + 1$, with

$$m_n = \max\{\sum\nolimits_{i \in B_n} P_{ij} | j = 1, ..., J\}.$$

The BI-SMART and RBI-SMART converge whenever there is a common nonnegative minimizer of the functions $h_i(\mathbf{x}), i = 1, .., I$. When there is no such vector, these algorithms are always observed to produce a limit cycle just as the ART does. So far, however, there is no proof of convergence to a limit cycle for entropy-based algorithms such as these.

For $k = 0, 1, ...,$ and $n = k(\mathrm{mod}\, N) + 1$ we can see easily that $\mathbf{x}^{k+1}$ in (67.5) is the unique minimizer of the function $G_k(\mathbf{x}, \mathbf{x}^k)$ given by

$$G_k(\mathbf{x}, \mathbf{x}^k) =$$

$$KL(\mathbf{x}, \mathbf{x}^k) - \gamma_k \sum\nolimits_{i \in B_n} KL(P\mathbf{x}_i, P\mathbf{x}_i^k) + \gamma_k \sum\nolimits_{i \in B_n} KL(P\mathbf{x}_i, y_i), \tag{67.8}$$

where $P\mathbf{x}_i^k = (P\mathbf{x}^k)_i$. Let $\hat{\mathbf{x}}$ be an arbitrary nonnegative solution of $\mathbf{y} = P\mathbf{x}$. Then we can show that

$$KL(\hat{\mathbf{x}}, \mathbf{x}^k) - KL(\hat{\mathbf{x}}, \mathbf{x}^{k+1}) = G_k(\mathbf{x}^{k+1}, \mathbf{x}^k) + \gamma_k \sum\nolimits_{i \in B_n} KL(y_i, P\mathbf{x}_i^k). \quad (67.9)$$

We want to conclude that the sequence $\{KL(\hat{\mathbf{x}}, \mathbf{x}^k)\}$ is decreasing. To be sure that $G_k(\mathbf{x}^{k+1}, \mathbf{x}^k) \geq 0$ we select $\gamma_k$ so that $1/\gamma_k \geq \sum_{i \in B_n} P_{ij}$ for all $j$.

We know from equation (61.1) that

$$KL(\mathbf{x}, \mathbf{z}) = KL(x_+, z_+) + KL(\mathbf{x}, \frac{x_+}{z_+}\mathbf{z}) \quad (67.10)$$

for any nonnegative vectors $\mathbf{x}$ and $\mathbf{z}$, with $x_+$ and $z_+ > 0$ denoting the sums of the entries of vectors $\mathbf{x}$ and $\mathbf{z}$, respectively. We reason here as follows. Therefore we know that $KL(\mathbf{x}, \mathbf{z}) \geq KL(x_+, z_+)$ always. Then

$$KL(\mathbf{x}, \mathbf{z}) \geq \gamma_k \sum\nolimits_j (\sum\nolimits_{i \in B_n} P_{ij}) KL(x_j, z_j) \geq \gamma_k \sum\nolimits_{i \in B_n} KL(P\mathbf{x}_i, P\mathbf{z}_i).$$

At the same time, we see that the decrease in the distance to a solution, as described by the left side of equation (67.9), is roughly proportional to $\gamma_k$, so we want $\gamma_k$ as large as possible. This suggests taking $\gamma_k = m_n^{-1}$, for $m_n$ as above. This is the choice used in the RBI-SMART. We note finally that the right side of equation (67.9) also contains the term $\sum_{i \in B_n} KL(y_i, P\mathbf{x}_i^k)$, which we want to be large also. As in the case of relaxed BI-ART, the ordering of the blocks affects the rate of convergence.

In all of the examples we have just considered we have convergence to a solution in the consistent case, but expect limit cycles for the block-iterative methods in the inconsistent case.

In the next chapter we consider a block-iterative version of the EMML method. We show how one particular attempt to form a block-iterative version of EMML, the *ordered subset* EM (OSEM), usually fails to converge in the consistent case and we show how to obtain a corrected algorithm.

# Chapter 68

# The Block-iterative EMML method

The EMML algorithm minimizes the function $KL(\mathbf{y}, P\mathbf{x})$ over nonnegative vectors $\mathbf{x}$, where $P$ is an $I$ by $J$ matrix of nonnegative entries with column sums equal to one and $y$ is the vector with positive entries. Say we are in the *consistent case* if there is a nonnegative $\mathbf{x}$ with $\mathbf{y} = P\mathbf{x}$; otherwise, we are in the *inconsistent case*. The EMML algorithm has the following iterative step:

**The EMML:**

$$x_j^{k+1} = x_j^k \sum_{i=1}^{I} P_{ij}\Big(\frac{y_i}{(P\mathbf{x}^k)_i}\Big). \tag{68.1}$$

In the consistent case the EMML converges to a nonnegative solution of $\mathbf{y} = P\mathbf{x}$; in the inconsistent case it converges to the (almost always) unique minimizer of $KL(\mathbf{y}, P\mathbf{x})$ [29], [30], [31]. If we had not redefined $P$ and $\mathbf{x}$ so as to have the columns of $P$ sum to one, the EMML would have had the iterative step

$$x_j^{k+1} = x_j^k \Big[\sum_{i=1}^{I} P_{ij}\Big(\frac{y_i}{(P\mathbf{x}^k)_i}\Big)\Big] / \Big[\sum_{i=1}^{I} P_{ij}\Big]. \tag{68.2}$$

The *ordered subset* EM (OSEM) method was derived from equation (68.2) by replacing both sums in (68.2) with partial sums over just those $i$ in $B_n$ [118]. The OSEM has the following iterative step:

**The OSEM:**

$$x_j^{k+1} = x_j^k \Big[\sum\nolimits_{i\in B_n} P_{ij}\Big(\frac{y_i}{(P\mathbf{x}^k)_i}\Big)\Big] / \Big[\sum\nolimits_{i\in B_n} P_{ij}\Big], \tag{68.3}$$

where $n = k(\mathrm{mod}\, N) + 1$.

The OSEM is mathematically incorrect. To be specific, it fails to converge to a solution in the consistent case, except for the quite special case of subset balance. The partition is said to have the *subset balance* property if, for each fixed value of $j$, the sums $\sum_{i\in B_n} P_{ij}$ are independent of $n$. The OSEM produces, in the consistent case, limit cycles typical of the behavior of block-iterative methods in the noisy (or inconsistent) case; in the inconsistent case, it is noisier still. How distinct from one another the vectors of this limit cycle are depends on the extent to which subset balance fails, as much as on the relative noise level. Recent use of the OSEM on clinically obtained patient data has shown that OSEM can provide accurate images in a fraction of the time required for the EMML. In practice in emission tomography, subset-balance may hold approximately in certain circumstances, so may not be an unreasonable assumption, particularly when the blocks have the same size.

A corrected version of OSEM, called the *rescaled block-iterative* EMML (RBI-EMML) method, was presented in [32] (see also [33] and [34]). The RBI-EMML has the following iterative step:

**The RBI-EMML:**

$$x_j^{k+1} = (1 - m_n^{-1} \sum\nolimits_{i\in B_n} P_{ij}) x_j^k + m_n^{-1} x_j^k \sum\nolimits_{i\in B_n} P_{ij} \Big(\frac{y_i}{(P\mathbf{x}^k)_i}\Big), \quad (68.4)$$

where, as earlier, we take

$$m_n = \max_j \{\sum\nolimits_{i\in B_n} P_{ij}\}.$$

When subset balance holds, the RBI-EMML reduces to the OSEM. The RBI-EMML converges, in the consistent case, to a solution, for every choice of subsets. In the inconsistent case the RBI-EMML is always observed to produce a limit cycle, although no proof of this fact is known; how distinct from one another the vectors of the limit cycle are depends on how large the minimum value of $KL(\mathbf{y}, P\mathbf{x})$ is. In contrast, the OSEM, applied in the inconsistent case, produces a limit cycle with the differences between vectors dependent not only on the noise in the data vector $\mathbf{y}$ but also on the deviation from subset balance. This causes the OSEM to appear noisier than it should.

When we are free to choose the blocks we could, of course, design them to have the subset balanced condition, or nearly so; but we are not always free to select the blocks as we wish. When we attempt to correct for patient motion, such as respiration, in emission tomography we may want to combine into a single block data received while the patient was in a fixed position. In this case the blocks may well have different sizes and subset balance is unlikely. The OSEM can perform poorly in such cases and, as

noted in [120], the RBI-EMML is a better choice, since it does not require subset balance.

Both the OSEM and the RBI-EMML appear noisier than EMML in the inconsistent case, early in the iteration, for another reason. In the inconsistent case, the ML solution can have at most $I-1$ nonzero entries (for almost all matrices $P$)[29]; if there are more unknowns than equations ($J > I$) then this means the ML solution will have zero entries and these tend to be sprinkled throughout the image. Fast methods such as OSEM and RBI-EMML get near this poor ML solution sooner than the EMML algorithm does, so they look noisier.

There is another reason why block-iterative reconstructions can appear noisier than their simultaneous counterparts. The individual vectors in the limit cycle have their own noise component; if we averaged over the vectors of the limit cycle to get the final result, instead of simply taking the last vector computed, the noise would be somewhat smoothed.

The RBI-EMML algorithm converges in the consistent case to a nonnegative solution of the linear system $\mathbf{y} = P\mathbf{x}$. As with ART, strong underrelaxation can be used to achieve convergence in the inconsistent case. Such a method, called the *row-action maximum likelihood algorithm* (RAMLA), was discovered independently by Browne and De Pierro [22]. The RAMLA has the following iterative step:

**The RAMLA:**

$$x_j^{k+1} = (1 - \lambda_k \sum\nolimits_{i \in B_n} P_{ij}) x_j^k + \lambda_k x_j^k \sum\nolimits_{i \in B_n} P_{ij} \Big( \frac{y_i}{(P\mathbf{x}^k)_i} \Big), \qquad (68.5)$$

where the positive relaxation parameters $\lambda_k$ converge to zero and $\sum_{k=0}^{+\infty} \lambda_k = +\infty$.

Before leaving this section, we point out that when there are $N = I$ blocks, so that each $B_n$ contains a single value of $i$, the RBI-EMML algorithm provides an analogue of the REMART in equation (67.6):

**The RBI-EMML for N=I:** for $k = 0, 1, ...$ and $i = k(\mathrm{mod}\, I) + 1$ let

$$x_j^{k+1} = (1 - m_i^{-1} P_{ij}) x_j^k + m_i^{-1} x_j^k P_{ij} \Big( \frac{y_i}{(P\mathbf{x}^k)_i} \Big), \qquad (68.6)$$

where $m_i = \max_j \{P_{ij}\}$.

The RBI-EMML has been applied recently to hyperspectral imaging [142]. In this application radar imaging from satellites is used to generate a picture of the ground. Because of the distance between the satellite and the ground a single image pixel can cover an area about 30 meters square. It is desirable to decompose such a pixel into constituent parts, to determine, for example, how much is grass, how much is water, etc. The signal received provides a power spectrum associated with the pixel, with each constituent

part contributing its own distinctive spectrum in proportion to its presence in the pixel. If the pixel is largely water, then the power spectrum is mainly that associated with water. If the pixel is half grass and half water then the power spectrum is a mixture of the power spectra of grass and of water. The received power spectrum is taken to be a mixture of known spectra associated with potential constituent parts. The RBI-EMML is then used to determine the proportion of each actually present within the received power spectrum.

# Chapter 69

# A general iterative algorithm

As we have seen, the bandlimited extrapolation procedure of Gerchberg-Papoulis, the SART of Anderson and Kak, Cimmino's algorithm and the Landweber and projected Landweber iterations are all particular cases of the CQ algorithm for the split feasibility problem. In this chapter we shall see that the CQ algorithm is itself a particular case of a much more general method, the Krasnoselskii/Mann (KM) [140] approach to finding fixed points for nonexpansive operators. The KM algorithm also includes the ART as a particular case. The discussion here is an abbreviated version of [40].

**Fixed point iterative methods:** The iterative methods we shall consider have the form

$$\mathbf{x}^{k+1} = T\mathbf{x}^k, \tag{69.1}$$

for $k = 0, 1, ...$, where $T$ is a linear or nonlinear continuous operator on a real (possibly infinite dimensional) Hilbert space $\mathcal{H}$ and $\mathbf{x}^0$ is an arbitrary starting vector. For any operator $T$ on $\mathcal{H}$ the *fixed point set* of $T$ is

$$\text{Fix}(T) = \{\mathbf{z} | \, T\mathbf{z} = \mathbf{z}\}.$$

If the iterative sequence defined by equation (69.1) converges then the limit is a member of $\text{Fix}(T)$.

In the algorithms of interest here the operator $T$ is selected so that the set $\text{Fix}(T)$ contains those vectors $\mathbf{z}$ that possess the properties we desire in a solution to the original signal processing or image reconstruction problem; finding a fixed point of the iteration leads to a solution of our problem.

Our concern here is with properties of the operator $T$ sufficient to guarantee convergence, for arbitrary $\mathbf{x}$, of the sequence $\{T^k\mathbf{x}\}$ whenever fixed points of $T$ exist. Most studies of iterative fixed point algorithms begin with the class of nonexpansive operators and we shall do the same.

**Nonexpansive operators:** A (possibly nonlinear) operator $N$ on $\mathcal{H}$ is called *nonexpansive* (ne) if, for all $\mathbf{x}$ and $\mathbf{y}$ in $\mathcal{H}$,

$$||N\mathbf{x} - N\mathbf{y}|| \leq ||\mathbf{x} - \mathbf{y}||.$$

The identity map $I\mathbf{x} = \mathbf{x}$ for all $\mathbf{x}$ is clearly ne; more generally, for any fixed vector $\mathbf{w}$ in $\mathcal{H}$ the maps $N\mathbf{x} = \mathbf{x} + \mathbf{w}$ and $N\mathbf{x} = -\mathbf{x} + \mathbf{w}$ are ne. As the example $N\mathbf{x} = -\mathbf{x}$ shows, convergence of the sequence $\{N^k\mathbf{x}\}$ is not guaranteed for ne operators, even when $\mathrm{Fix}(N)$ is nonempty.

**The Krasnoselskii/Mann approach:** The Krasnoselskii/Mann (KM) [140] approach to finding fixed points of a ne operator $N$ is quite simple, yet remarkably useful. Given a ne operator $N$, let

$$T = (1 - \alpha)I + \alpha N$$

for some $\alpha \in (0, 1)$. The operator $T$ is then said to be *averaged* (av). The Krasnoselskii/Mann theorem discussed below tells us that the sequence defined by equation (69.1) then converges (weakly) to a fixed point of $N$ whenever such points exist. The metric projection $P_C$ onto a convex set $C$ is av, as is the operator $(I - \gamma\nabla f)$ if $\nabla f$ is Lipschitz continuous and the parameter $\gamma$ is appropriately chosen; the product of finitely many av operators is av, so the operators $P_{C_2}P_{C_1}$ and $P_C(I - \gamma\nabla f)$ are also av. Consequently, fixed points of such operators are limits of the sequence defined by equation (69.1).

**Averaged operators:** As we have seen, the fact that a ne operator $N$ has fixed points is not sufficient to guarantee convergence of the orbit sequence $\{N^k\mathbf{x}\}$; additional conditions are needed. An operator $S$ on $\mathcal{H}$ is said to be a *strict contraction* (sc) if there is $\sigma \in (0, 1)$ such that, for all $\mathbf{x}$ and $\mathbf{y}$ in $\mathcal{H}$,

$$||S\mathbf{x} - S\mathbf{y}|| \leq \sigma||\mathbf{x} - \mathbf{y}||.$$

The well known Banach-Picard theorem [87] assures us that the operator $S$ has a unique fixed point, to which the orbit sequence $\{S^k\mathbf{x}\}$ converges, for any starting point $\mathbf{x}$. Requiring the operator to be a strict contraction is quite restrictive; most of the operators we are interested in here have multiple fixed points, so are not sc. The Krasnoselskii/Mann theorem suggests strongly that we should concentrate on averaged operators. We have the following result.

**Theorem 69.1** *Let $T$ be an av operator on $\mathcal{H}$ and let Fix(T) be nonempty. Then the orbit sequence $\{T^k\mathbf{x}\}$ converges weakly to a member of Fix(T), for any $\mathbf{x}$.*

We shall include a proof of this theorem, for the finite dimensional case.

Recall that the CQ algorithm has the iterative step

$$\mathbf{x}^{k+1} = P_C(\mathbf{x}^k - \gamma A^T(I - P_Q)A\mathbf{x}^k), \tag{69.2}$$

where $\gamma \in (0, 2/\rho(A^TA))$, for $\rho(A^TA)$ the spectral radius of the matrix $A^TA$, which is also its largest eigenvalue. The CQ algorithm converges to a solution of the SFP, for any starting vector $\mathbf{x}^0$, whenever the SFP has solutions. When the SFP has no solutions, the CQ algorithm converges to a minimizer of the function

$$f(\mathbf{x}) = \frac{1}{2}||P_QA\mathbf{x} - A\mathbf{x}||^2$$

over the set $C$, provided such constrained minimizers exist. This result is a consequence of Theorem 69.1: the function $f(\mathbf{x})$ is convex and differentiable. Its gradient operator $\nabla f(\mathbf{x}) = A^T(I - P_Q)A\mathbf{x}$ can be shown to be $\lambda$-Lipschitz continuous for $\lambda = \rho(A^TA)$, from which it follows that the operator

$$T(\mathbf{x}) = P_C(\mathbf{x} - \gamma A^T(I - P_Q)A\mathbf{x})$$

is averaged for $\gamma \in (0, 2/\rho(A^TA))$.

**Proof of the KM theorem:** The following identity relates an operator $T$ to its complement $G = I - T$:

$$||\mathbf{x} - \mathbf{y}||^2 - ||T\mathbf{x} - T\mathbf{y}||^2 = 2\langle G\mathbf{x} - G\mathbf{y}, \mathbf{x} - \mathbf{y}\rangle - ||G\mathbf{x} - G\mathbf{y}||^2. \tag{69.3}$$

Let $\mathbf{z}$ be a fixed point of the nonexpansive operator $N$ and let $\alpha \in (0, 1)$. Let $T = (1 - \alpha)I + \alpha N$, so the iterative step becomes

$$\mathbf{x}^{k+1} = T\mathbf{x}^k = (1 - \alpha)\mathbf{x}^k + \alpha N\mathbf{x}^k. \tag{69.4}$$

The identity in equation (69.3) is the key to proving Theorem 69.1.

Using $T\mathbf{z} = \mathbf{z}$ and $(I - T)\mathbf{z} = 0$ and setting $G = I - T$ we have

$$||\mathbf{z} - \mathbf{x}^k||^2 - ||T\mathbf{z} - \mathbf{x}^{k+1}||^2 = 2\langle G\mathbf{z} - G\mathbf{x}^k, \mathbf{z} - \mathbf{x}^k\rangle - ||G\mathbf{z} - G\mathbf{x}^k||^2$$

so that

$$||\mathbf{z} - \mathbf{x}^k||^2 - ||\mathbf{z} - \mathbf{x}^{k+1}||^2 \geq (\frac{1}{\alpha} - 1)||\mathbf{x}^k - \mathbf{x}^{k+1}||^2. \tag{69.5}$$

Consequently the sequence $\{\mathbf{x}^k\}$ is bounded, the sequence $\{||\mathbf{z} - \mathbf{x}^k||\}$ is decreasing and the sequence $\{||\mathbf{x}^k - \mathbf{x}^{k+1}||\}$ converges to zero. Let $\mathbf{x}^*$ be a cluster point of $\{\mathbf{x}^k\}$. Then we have $T\mathbf{x}^* = \mathbf{x}^*$, so we may use $\mathbf{x}^*$ in place of the arbitrary fixed point $\mathbf{z}$. It follows then that the sequence $\{||\mathbf{x}^* - \mathbf{x}^k||\}$ is decreasing; since a subsequence converges to zero, the entire sequence converges to zero and the proof is complete.

# Chapter 70

# The Wave Equation

In this chapter and the next we demonstrate how the problem of Fourier transform estimation from sampled data arises in the processing of measurements obtained by sampling electromagnetic or acoustic field fluctuations, as in radar or sonar.

In many areas of remote sensing what we measure are the fluctuations in time of an electromagnetic or acoustic field. Such fields are described mathematically as solutions of certain partial differential equations, such as the *wave equation*. A function $u(x, y, z, t)$ is said to satisfy the *three-dimensional wave equation* if

$$u_{tt} = c^2(u_{xx} + u_{yy} + u_{zz}) = c^2\nabla^2 u,$$

where $u_{tt}$ denotes the second partial derivative of $u$ with respect to the time variable $t$ twice and $c > 0$ is the (constant) speed of propagation. More complicated versions of the wave equation permit the speed of propagation $c$ to vary with the spatial variables $x, y, z$, but we shall not consider that here.

We use the method of *separation of variables* at this point, to get some idea about the nature of solutions of the wave equation. Assume, for the moment, that the solution $u(t, x, y, z)$ has the simple form

$$u(t, x, y, z) = f(t)g(x, y, z).$$

Inserting this separated form into the wave equation we get

$$f''(t)g(x, y, z) = c^2 f(t)\nabla^2 g(x, y, z)$$

or

$$f''(t)/f(t) = c^2\nabla^2 g(x, y, z)/g(x, y, z).$$

The function on the left is independent of the spatial variables, while the one on the right is independent of the time variable; consequently, they

must both equal the same constant, which we denote $-\omega^2$. From this we have two separate equations,

$$f''(t) + \omega^2 f(t) = 0, \tag{70.1}$$

and

$$\nabla^2 g(x,y,z) + \frac{\omega^2}{c^2} g(x,y,z) = 0. \tag{70.2}$$

The equation (70.2) is the *Helmholtz equation*.

Equation (70.1) has for its solutions the functions $f(t) = \cos(\omega t)$ and $\sin(\omega t)$, or, in complex form, the complex exponential functions $f(t) = e^{i\omega t}$ and $f(t) = e^{-i\omega t}$. Functions $u(t,x,y,z) = f(t)g(x,y,z)$ with such time dependence are called *time-harmonic* solutions.

In three-dimensional spherical coordinates with $r = \sqrt{x^2+y^2+z^2}$ a radial function $u(r,t)$ satisfies the wave equation if

$$u_{tt} = c^2(u_{rr} + \frac{2}{r}u_r).$$

**Exercise 1:** Show that the radial function $u(r,t) = \frac{1}{r}h(r-ct)$ satisfies the wave equation for any twice differentiable function $h$.

Radial solutions to the wave equation have the property that at any fixed time the value of $u$ is the same for all the points on a sphere centered at the origin; the curves of constant value of $u$ are these spheres, for each fixed time.

Suppose at time $t = 0$ the function $h(r,0)$ is zero except for $r$ near zero; that is, initially, there is a localized disturbance centered at the origin. As time passes that disturbance spreads out spherically. When the radius of a sphere is very large, the surface of the sphere appears planar, to an observer on that surface, who is said then to be in the *far field*. This motivates the study of solutions of the wave equation that are constant on planes; the so-called *planewave solutions*.

**Exercise 2:** Let $\mathbf{s} = (x,y,z)$ and $u(\mathbf{s},t) = u(x,y,z,t) = e^{i\omega t}e^{i\mathbf{k}\cdot\mathbf{s}}$. Show that $u$ satisfies the wave equation $u_{tt} = c^2\nabla^2 u$ for any real vector $\mathbf{k}$, so long as $||\mathbf{k}||^2 = \omega^2/c^2$. This solution is a planewave associated with frequency $\omega$ and *wavevector* $\mathbf{k}$; at any fixed time the function $u(\mathbf{s},t)$ is constant on any plane in three dimensional space having $\mathbf{k}$ as a normal vector.

# Chapter 71

# Array Processing

In radar and sonar the field $u(\mathbf{s}, t)$ being sampled is usually viewed as a discrete or continuous superposition of planewave solutions with various amplitudes, frequencies and wavevectors. We sample the field at various spatial locations $\mathbf{s}_m$, $m = 1, ..., M$, for $t$ in some finite interval of time. We simplify the situation a bit now by assuming that all the planewave solutions are associated with the same frequency, $\omega$. If not, we perform an FFT on the functions of time received at each sensor location $\mathbf{s}_m$ and keep only the value associated with the desired frequency $\omega$.

In the continuous superposition model the field is

$$u(\mathbf{s}, t) = e^{i\omega t} \int f(\mathbf{k}) e^{i\mathbf{k}\cdot\mathbf{s}} d\mathbf{k}.$$

Our measurements at the sensor locations $\mathbf{s}_m$ give us the values

$$F(\mathbf{s}_m) = \int f(\mathbf{k}) e^{i\mathbf{k}\cdot\mathbf{s}_m} d\mathbf{k},$$

for $m = 1, ..., M$. The data are then Fourier transform values of the complex function $f(\mathbf{k})$; $f(\mathbf{k})$ is defined for all three-dimensional real vectors $\mathbf{k}$, but is zero, in theory, at least, for those $\mathbf{k}$ whose squared length $||\mathbf{k}||^2$ is not equal to $\omega^2/c^2$. Our goal is then to estimate $f(\mathbf{k})$ from finitely many values of its Fourier transform. Since each $\mathbf{k}$ is a normal vector for its planewave field component, determining the value of $f(\mathbf{k})$ will tell us the strength of the planewave component coming from the direction $\mathbf{k}$.

The collection of sensors at the spatial locations $\mathbf{s}_m$, $m = 1, ..., M$, is called *an array* and the size of the array, in units of the wavelength $\lambda = 2\pi c/\omega$, is called the *aperture* of the array. Generally the larger the aperture the better, but what is a large aperture for one value of $\omega$ will be a smaller aperture for a lower frequency. The book by Haykin [106] is a useful reference, as is the review paper by Wright, Pridham and Kay [183].

In some applications the sensor locations are essentially arbitrary, while in others their locations are carefully chosen. Sometimes, the sensors are collinear, as in sonar towed arrays. Let's look more closely at the collinear case.

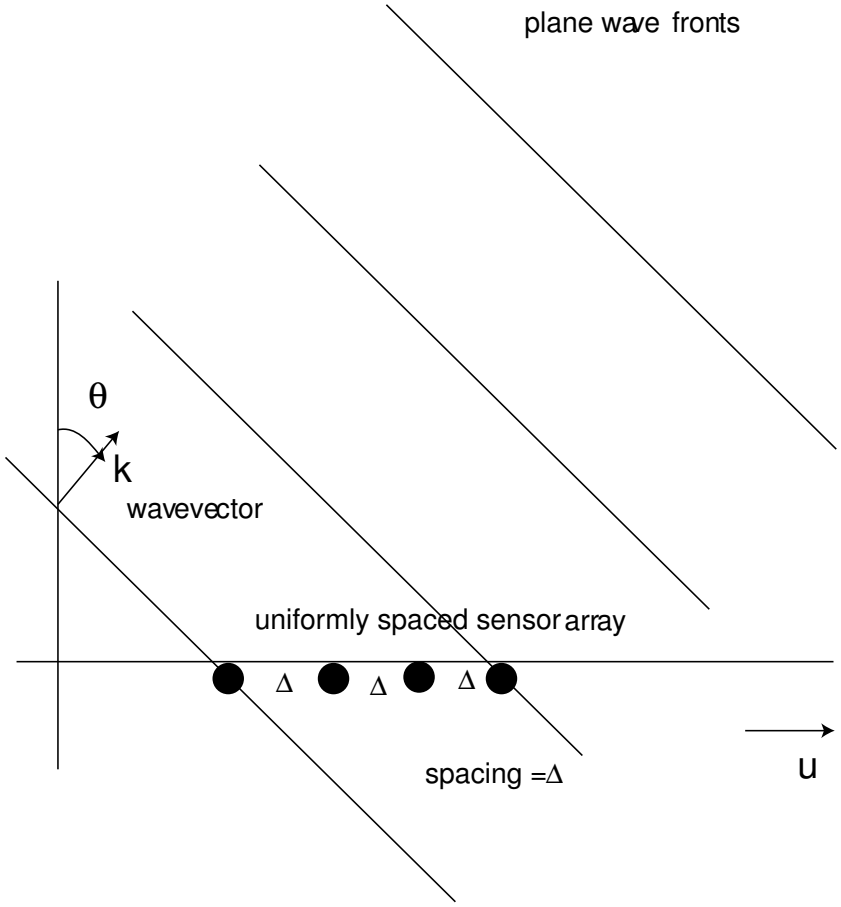



Figure 71.1: A uniform line array sensing a planewave field.

We assume now that the sensors are equispaced along the $x$-axis, at locations $(m\Delta, 0, 0)$, $m = 1, ..., M$, where $\Delta > 0$ is the sensor spacing; such an arrangement is called a *uniform line array*; this setup is illustrated in Figure 71.1. Our data is then

$$F_m = F(\mathbf{s}_m) = F((m\Delta, 0, 0)) = \int f(\mathbf{k}) e^{im\Delta \mathbf{k} \cdot (1,0,0)} d\mathbf{k}.$$

Since $\mathbf{k} \cdot (1,0,0) = \frac{\omega}{c}\cos\theta$, for $\theta$ the angle between the vector $\mathbf{k}$ and the $x$-axis, we see that there is some ambiguity now; we cannot distinguish the cone of vectors that have the same $\theta$. It is common then to assume that the wavevectors $\mathbf{k}$ have no $z$-component and that $\theta$ is the angle between two vectors in the $x, y$-plane, the so-called *angle of arrival*. The *wavenumber* variable $k = \frac{\omega}{c}\cos\theta$ lies in the interval $[-\frac{\omega}{c}, \frac{\omega}{c}]$ and we imagine that $f(\mathbf{k})$ is now $f(k)$, defined for $|k| \leq \frac{\omega}{c}$. The Fourier transform of $f(k)$ is $F(s)$, a function of a single real variable $s$. Our data is then viewed as the values $F(m\Delta)$, for $m = 1, ..., M$. Since the function $f(k)$ is zero for $|k| > \frac{\omega}{c}$ the Nyquist spacing in $s$ is $\frac{\pi c}{\omega}$, which is $\frac{\lambda}{2}$, where $\lambda = \frac{2\pi c}{\omega}$ is the wavelength.

To avoid aliasing, which now means mistaking one direction of arrival for another, we need to select $\Delta \leq \frac{\lambda}{2}$. When we have oversampled, so that $\Delta < \frac{\lambda}{2}$, the interval $[-\frac{\omega}{c}, \frac{\omega}{c}]$, the so-called *visible region*, is strictly smaller than the interval $[-\frac{\pi}{\Delta}, \frac{\pi}{\Delta}]$. If the model of propagation is accurate all the signal component planewaves will correspond to wavenumbers $k$ in the visible region and the background noise will also appear as a superposition of such propagating planewaves. In practice, there can be components in the noise that appear to come from wavenumbers $k$ outside of the visible region; this means these components of the noise are not due to distant sources propagating as planewaves, but, perhaps, to sources that are in the *near field*, or localized around individual sensors, or coming from the electronics within the sensors.

Using the formula $\lambda\omega = 2\pi c$ we can calculate the Nyquist spacing for any particular case of planewave array processing. For electromagnetic waves the propagation speed is the speed of light, which we shall take here to be $c = 3 \times 10^8$ meters per second. The wavelength $\lambda$ for gamma rays is around one Angstrom, which is $10^{-10}$ meters; for x-rays it is about one millimicron, or $10^{-9}$ meters; the visible spectrum has wavelengths that are a little less than one micron, that is, $10^{-6}$ meters. Shortwave radio has wavelength around one millimeter; broadcast radio has a $\lambda$ running from about 10 meters to 1000 meters, while the so-called long radio waves can have wavelengths several thousand meters long. At the one extreme it is impractical (if not physically impossible) to place individual sensors at the Nyquist spacing of fractions of microns, while at the other end, managing to place the sensors far enough apart is the challenge.

The wavelengths used in primitive early radar at the start of World War II were several meters long. Since resolution is proportional to aperture, which, in turn, is the length of the array, in units of wavelength, antennae for such radar needed to be quite large. As Körner notes in [128], the general feeling at the time was that the side with the shortest wavelength would win the war. The cavity magnetron, invented during the war by British scientists, made possible 10 cm wavelength radar, which could then easily be mounted on planes.

In ocean acoustics it is usually assumed that the speed of propagation of sound is around 1500 meters per second, although deviations from this *ambient sound speed* are significant, and since they are caused by such things as temperature differences in the ocean, can be used to estimate these differences. At around the frequency $\omega = 50$ Hz we find sound generated by man-made machinery, such as motors in vessels, with higher frequency harmonics sometimes present also; at other frequencies the main sources of acoustic energy may be wind-driven waves or whales. The wavelength for 50 Hz is $\lambda = 30$ meters; sonar will typically operate both above and below this wavelength. It is sometimes the case that the array of sensors is fixed in place, so what may be Nyquist spacing for 50 Hz will be oversampling for 20 Hz.

It is often the case that we are primarily interested in the values $|f(\mathbf{k})|$, not the complex values $f(\mathbf{k})$. Since the Fourier transform of the function $|f(\mathbf{k})|^2$ is the autocorrelation function obtained by convolving the function $F$ with $\overline{F}$, we can mimic the approach used earlier for power spectrum estimation to find $|f(\mathbf{k})|$. We can now employ the nonlinear methods such as Burg's MEM and Capon's maximum likelihood method.

In array processing, as in other forms of signal and image processing, we want to remove the noise and enhance the information-bearing component, the signal. To do this we need some idea of the statistical behavior of the noise, we need a physically accurate description of what the signals probably look like and we need a way to use this information. Much of our discussion up to now has been about the many ways in which such prior information can be incorporated in linear and nonlinear procedures. We have not said much about the important issue of the sensitivity of these methods to mismatch; that is, What happens when our physical model is wrong or the statistics of the noise is not what we thought it was? We did note earlier how Burg's MEM resolves closely spaced sinusoids when the background is white noise, but when the noise is correlated, MEM can degrade rapidly.

Even when the physical model and noise statistics are reasonably accurate, slight errors in the hardware can cause rapid degradation of the processor. Sometimes acoustic signal processing is performed with sensors that are designed to be expendable and are therefore less expensive and more prone to errors than more permanent equipment. Knowing what a sensor has received is important, but so is knowing when it received it. Slight phase errors caused by the hardware can go unnoticed when the data is processed in one manner, but can ruin the performance of another method.

The information we seek is often stored redundantly in the data and hardware errors may harm only some of these storage locations, making robust processing still possible. As we saw in our discussion of eigenvector methods, information about the frequencies of the complex exponential

components of the signal are stored in the roots of the polynomials obtained from some of the eigenvectors. In [52] it was demonstrated that, in the presence of correlated noise background, phase errors distort the roots of some of these polynomials more than others; robust estimation of the frequencies is still possible if the stable roots are interrogated.

We have focused here exclusively on planewave propagation, which results when the source is far enough way from the sensors and the speed of propagation is constant. In many important applications these conditions are violated, different versions of the wave equation are needed, which have different solutions. For example, sonar signal processing in environments such as shallow channels, in which some of the sound reaches the sensors only after interacting with the ocean floor or the surface, requires more complicated parameterized models for solutions of the appropriate wave equation. Lack of information about the depth and nature of the bottom can also cause errors in the signal processing. In some cases it is possible to use acoustic energy from known sources to determine the needed information.

Array signal processing can be done in *passive* or *active* mode. In passive mode the energy is either reflected off of or originates at the object of interest: the moon reflects sunlight, while ships generate their own noise. In the active mode the object of interest does not generate or reflect enough energy by itself, so the energy is generated by the party doing the sensing: active sonor is sometimes used to locate quiet vessels, while radar is used to locate planes in the sky or to map the surface of the earth. In the February 2003 issue of Harper's is an article on scientific apocalypse, dealing with the search for near-earth asteroids. These objects are initially detected by passive optical observation, as small dots of reflected sunlight; once detected, they are then imaged by active radar to determine their size, shape, rotation and such.

# Chapter 72

# Matched Field Processing

Previously we considered the array processing problem in the context of planewave propagation. When the environment is more complicated the wave equation must be modified to reflect the physics of the situation and the signal processing modified to incorporate that physics. A good example of such modification is provided by acoustic signal processing in shallow water, the topic of this chapter.

In the shallow water situation the acoustic energy from the source interacts with the surface and with the bottom of the channel, prior to being received by the sensors. The nature of this interaction is described by the wave equation in cylindrical coordinates. The deviation from the ambient pressure is the function $p(t, \mathbf{s}) = p(t, r, z, \theta)$, where $\mathbf{s} = (r, z, \theta)$ is the spatial vector variable, $r$ is the range, $z$ the depth and $\theta$ the bearing angle in the horizontal. We assume a single frequency, $\omega$, so that

$$p(t, \mathbf{s}) = e^{i\omega t} g(r, z, \theta).$$

We shall assume cylindrical symmetry to remove the $\theta$ dependence; in many applications the bearing is essentially known or limited by the environment or can be determined by other means. The sensors are usually positioned in a vertical array in the channel, with the top of the array taken to be the origin of the coordinate system and positive $z$ taken to mean positive depth below the surface. We shall also assume that there is a single source of acoustic energy located at range $r_s$ and depth $z_s$.

To simplify a bit we assume here that the sound speed $c = c(z)$ does not change with range, but only with depth, and that the channel has constant depth and density. Then the Helmholtz equation for the function $g(r, z)$ is

$$\nabla^2 g(r, z) + [\omega / c(z)]^2 g(r, z) = 0.$$

The Laplacian is

$$\nabla^2 g(r,z) = g_{rr}(r,z) + \frac{1}{r} g_r(r,z) + g_{zz}(r,z).$$

We separate the variables once again, writing

$$g(r,z) = f(r)u(z).$$

Then the range function $f(r)$ must satisfy the differential equation

$$f''(r) + \frac{1}{r} f'(r) = -\alpha f(r)$$

and the depth function $u(z)$ satisfies the differential equation

$$u''(z) + k(z)^2 u(z) = \alpha u(z),$$

where $\alpha$ is a separation constant and

$$k(z)^2 = [\omega/c(z)]^2.$$

Taking $\lambda^2 = \alpha$ the range equation becomes

$$f''(r) + \frac{1}{r} f'(r) + \lambda^2 f(r) = 0,$$

which is Bessel's equation, with Hankel function solutions. The depth equation becomes

$$u''(z) + (k(z)^2 - \lambda^2)u(z) = 0,$$

which is of Sturm-Liouville type. The boundary conditions pertaining to the surface and the channel bottom will determine the values of $\lambda$ for which a solution exists.

To illustrate the way in which the boundary conditions become involved, we consider two examples.

## The homogeneous layer model:

We assume now that the channel consists of a single homogeneous layer of water of constant density, constant depth $d$ and constant sound speed $c$. We impose the following boundary conditions:

**a.** Pressure-release surface: $u(0) = 0$;

**b.** Rigid bottom: $u'(d) = 0$.

With $\gamma^2 = (k^2 - \lambda^2)$ we get $\cos(\gamma d) = 0$, so the permissible values of $\lambda$ are

$$\lambda_m = (k^2 - [(2m-1)\pi/2d]^2)^{1/2},\ m = 1, 2, ....$$

The normalized solutions of the depth equation are now

$$u_m(z) = \sqrt{2/d}\sin(\gamma_m z),$$

where

$$\gamma_m = \sqrt{k^2 - \lambda_m^2} = (2m-1)\pi/2d,\ m = 1, 2, ....$$

For each $m$ the corresponding function of the range satisfies the differential equation

$$f''(r) + \frac{1}{r}f'(r) + \lambda_m^2 f(r),$$

which has solution $H_0^{(1)}(\lambda_m r)$, where $H_0^{(1)}$ is the zeroth order Hankel function solution of Bessel's equation. The asymptotic form for this function is

$$\pi i H_0^{(1)}(\lambda_m r) = \sqrt{2\pi/\lambda_m r}\exp(-i(\lambda_m r + \frac{\pi}{4})).$$

It is this asymptotic form that is used in practice. Note that when $\lambda_m$ is complex with a negative imaginary part there will be a decaying exponential in this solution, so this term will be omitted in the signal processing.

Having found the range and depth functions we write $g(r, z)$ as a superposition of these elementary products, called the *modes*:

$$g(r, z) = \sum\nolimits_{m=1}^{M} A_m H_0^{(1)}(\lambda_m r) u_m(z),$$

where $M$ is the number of propagating modes free of decaying exponentials. The $A_m$ can be found from the original Helmholtz equation; they are

$$A_m = (i/4)u_m(z_s),$$

where $z_s$ is the depth of the source of the acoustic energy. Notice that the depth of the source also determines the strength of each mode in this superposition; this is described by saying that the source has *excited* certain modes and not others.

The eigenvalues $\lambda_m$ of the depth equation will be complex when

$$k = \frac{\omega}{c} < \frac{(2m-1)\pi}{2d},$$

If $\omega$ is below the *cut-off frequency* $\frac{\pi c}{2d}$ then all the $\lambda_m$ are complex and there are no propagating modes ($M = 0$). The number of propagating modes is

$$M = \frac{1}{2} + \frac{\omega d}{\pi c},$$

which is $\frac{1}{2}$ plus the depth of the channel in units of half-wavelengths.

This model for shallow water propagation is helpful in revealing a number of the important aspects of modal propagation, but is of limited practical utility. A more useful and realistic model is the *Pekeris waveguide*.

## The Pekeris waveguide:

Now we assume that the water column has constant depth $d$, sound speed $c$ and density $b$. Beneath the water is an infinite half-space with sound speed $c' > c$ and density $b'$. Figure 72.1 illustrates the situation.

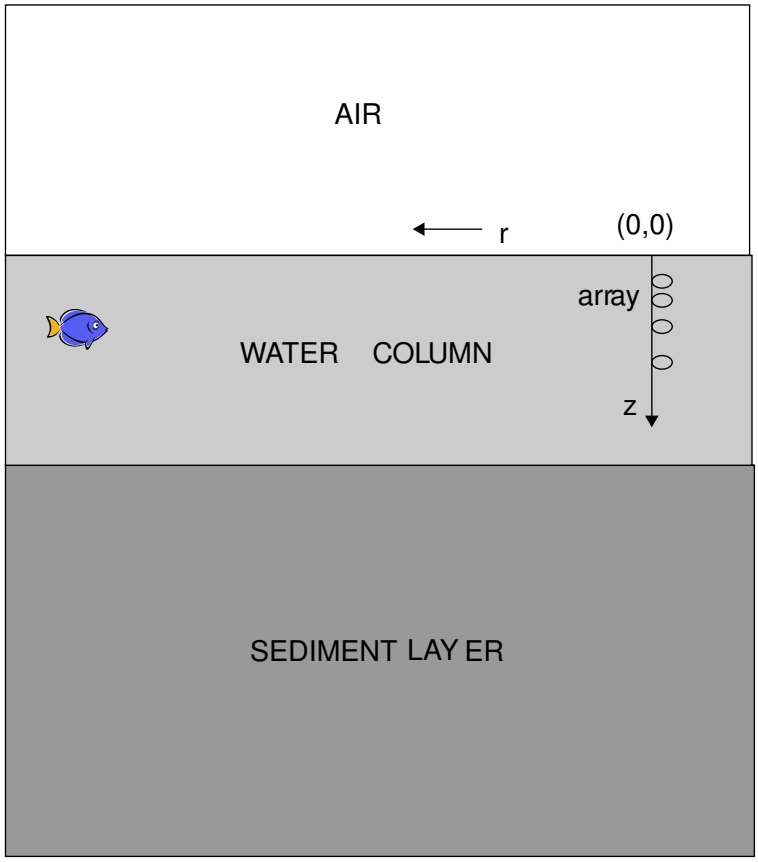



Figure 72.1: The Pekeris model.

Using the new depth variable $v = \frac{\omega z}{c}$ the depth equation becomes

$$u''(v) + \lambda^2 u(v) = 0, \text{ for } 0 \leq v \leq \frac{\omega d}{c},$$

and

$$u''(v) + ((\frac{c}{c'})^2 - 1 + \lambda^2)u(v) = 0, \text{ for } \frac{\omega d}{c} < v.$$

To have a solution $\lambda$ must satisfy the equation

$$\tan(\lambda\omega d/c) = -(\lambda b/b')/\sqrt{1-(\frac{c}{c'})^2-\lambda^2},$$

with

$$1-(\frac{c}{c'})^2-\lambda^2 \geq 0.$$

The *trapped modes* are those whose corresponding $\lambda$ satisfies

$$1 \geq 1-\lambda^2 \geq (\frac{c}{c'})^2.$$

The eigenfunctions are

$$u_m(v) = \sin(\lambda_m v), \text{ for } 0 \leq v \leq \frac{\omega d}{c}$$

and

$$u_m(v) = \exp\left(-v\sqrt{1-(\frac{c}{c'})^2-\lambda^2}\right), \text{ for } \frac{\omega d}{c} < v.$$

Although the Pekeris model has its uses, it still may not be realistic enough in some cases and more complicated propagation models will be needed.

## The general normal mode model:

Regardless of the model by which the modal functions are determined, the general *normal mode expansion* for the range-independent case is

$$g(r,z) = \sum\nolimits_{m=1}^{M} u_m(z)s_m(r,z_s),$$

where $M$ is the number of propagating modes and $s_m(r,z_s)$ is the *modal amplitude* containing all the information about the source of the sound.

## Matched field processing:

In planewave array processing we write the acoustic field as a superposition of planewave fields and try to find the corresponding amplitudes. This can be done using a matched filter, although high resolution methods can also be used. In the matched filter approach, we fix a wavevector and then match the data with the vector that describes what we would have received at the sensors had there been but a single planewave present corresponding to that fixed wavevector; we then repeat for other fixed wavevectors. In more complicated acoustic environments, such as normal mode propagation in shallow water, we write the acoustic field as a superposition of fields due to sources of acoustic energy at individual points in range and depth and

then seek the corresponding amplitudes. Once again, this can be done using a matched filter.

In matched field processing we fix a particular range and depth and compute what we would have received at the sensors had the acoustic field been generated solely by a single source at that location. We then match the data with this computed vector. We repeat this process for many different choices of range and depth, obtaining a function of $r$ and $z$ showing the likely locations of actual sources. As in the planewave case, high resolution nonlinear methods can also be used.

As in the planewave case, the performance of our processing methods can be degraded by incorrect description of the environment, as well as by phase errors and the like introduced by the hardware [28]. Once again, it is necessary to seek out those locations within the data where the information we seek is less disturbed by such errors [41], [49].

Good sources for more information concerning matched field processing are the book by Tolstoy [176] and the papers [5], [24], [90], [112], [113], [165], [166], [167], [175] and [184].

# Chapter 73

# Transmission Tomography

In this chapter we show how the two dimensional Fourier transform arises in transmission tomographic image processing. See the texts [147] and [148] for more detailed discussion of these matters.

As an x-ray beam passes through the body it encounters various types of matter, soft tissue, bone, ligaments, air, each weakening the beam to a greater or lesser extent. If the intensity of the beam upon entry is $I_{in}$ and $I_{out}$ is its lesser intensity after passing through the body, then

$$I_{out} = I_{in} e^{-\int_L f},$$

where $f = f(x, y) \geq 0$ is the *attenuation function* describing the two-dimensional distribution of matter within the slice of the body being scanned and $\int_L f$ is the integral of the function $f$ over the line $L$ along which the x-ray beam has passed. To see why this is the case imagine the line $L$ parametrized by the variable $s$ and consider the intensity function $I(s)$ as a function of $s$. For small $\Delta s > 0$ the drop in intensity from the start to the end of the interval $[s, s+\Delta s]$ is approximately proportional to the intensity $I(s)$, to the attenuation $f(s)$ and to $\Delta s$, the length of the interval; that is,

$$I(s) - I(s + \Delta s) \approx f(s) I(s) \Delta s.$$

Dividing by $\Delta s$ and letting $\Delta s$ approach zero, we get

$$\frac{dI}{ds} = -f(s) I(s).$$

The solution of this differential equation is

$$I(s) = I(0) \exp(-\int_{u=0}^{u=s} f(u) du).$$

From knowledge of $I_{in}$ and $I_{out}$ we can determine $\int_L f$. As we shall see, if we know $\int_L f$ for every line in the $x, y$-plane we can reconstruct the attenuation function $f$. In actual *computer-assisted tomography* (CAT) scans we know line integrals only approximately and only for finitely many lines. Figure 73.1 illustrates the situation. In practice the function $f$ is replaced by a grid of pixels, as shown in Figure 73.2.

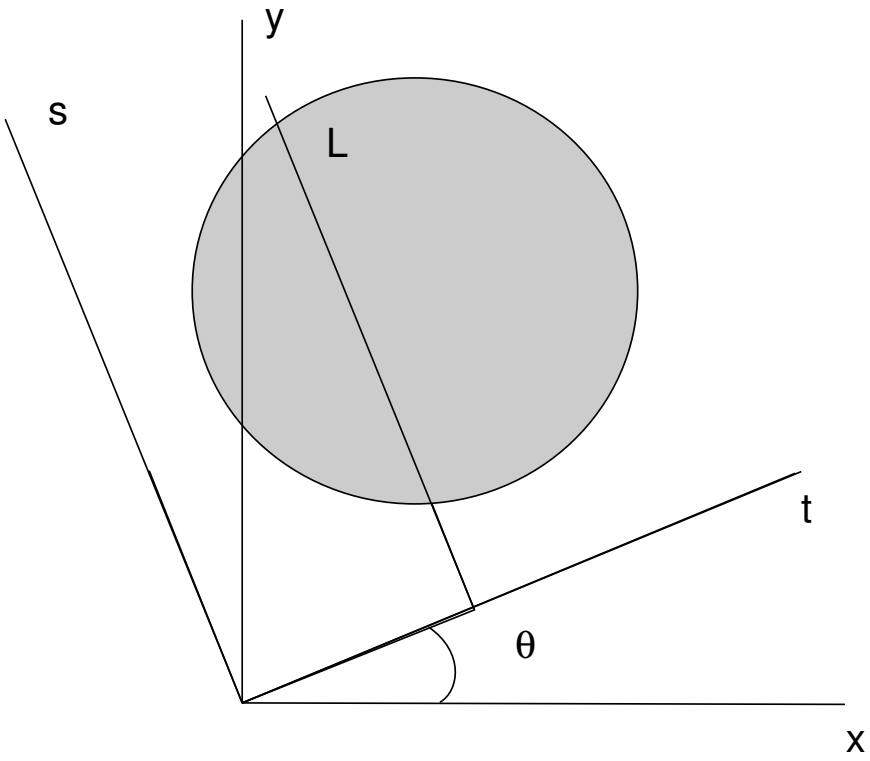



Figure 73.1: The Radon transform of $f$ at $(t, \theta)$ is the line integral of $f$ along line $L$.

Let $\theta$ be a fixed angle in the interval $[0, \pi)$ and consider the rotation of

the $x, y$ coordinate axes to produce the $t, s$ axis system, where

$$t = x\cos\theta + y\sin\theta,$$

and

$$s = -x\sin\theta + y\cos\theta.$$

We can then write the attenuation function $f$ as a function of the variables $t$ and $s$. For each fixed value of $t$ we compute the integral $\int f(x,y)ds$, obtaining the integral of $f(x,y) = f(t\cos\theta - s\sin\theta, t\sin\theta + s\cos\theta)$ along the single line $L$ corresponding to the fixed values of $\theta$ and $t$. We repeat this process for every value of $t$ and then change the angle $\theta$ and repeat again. In this way we obtain the integrals of $f$ over every line $L$ in the plane. We denote by $r_f(\theta, t)$ the integral

$$r_f(\theta, t) = \int_L f(x,y)ds.$$

The function $r_f(\theta, t)$ is called the *Radon transform* of $f$.

For fixed $\theta$ the function $r_f(\theta, t)$ is a function of the single real variable $t$; let $R_f(\theta, \omega)$ be its Fourier transform. Then

$$R_f(\theta,\omega) = \int (\int f(x,y)ds)e^{i\omega t}dt,$$

which we can write as

$$R_f(\theta,\omega) = \int\int f(x,y)e^{i\omega(x\cos\theta + y\sin\theta)}dxdy = F(\omega\cos\theta, \omega\sin\theta),$$

where $F(\omega\cos\theta, \omega\sin\theta)$ is the two-dimensional Fourier transform of the function $f(x,y)$, evaluated at the point $(\omega\cos\theta, \omega\sin\theta)$; this relationship is called the *central slice theorem*. For fixed $\theta$ as we change the value of $\omega$ we obtain the values of the function $F$ along the points of the line making the angle $\theta$ with the horizontal axis. As $\theta$ varies in $[0, \pi)$ we get all the values of the function $F$. Once we have $F$ we can obtain $f$ using the formula for the two-dimensional inverse Fourier transform. We conclude that we are able to determine $f$ from its line integrals.

The inversion formula tells us that the function $f(x,y)$ can be obtained as

$$f(x,y) = \frac{1}{4\pi^2}\int\int F(u,v)e^{-i(xu+yv)}dudv.$$

Expressing the double integral in polar coordinates $(\omega, \theta)$, with $\omega \geq 0$, $u = \omega\cos\theta$ and $v = \omega\sin\theta$, we get

$$f(x,y) = \frac{1}{4\pi^2}\int_0^{2\pi}\int_0^{\infty} F(u,v)e^{-i(xu+yv)}\omega d\omega d\theta,$$

or

$$f(x,y) = \frac{1}{4\pi^2}\int_0^{\pi}\int_{-\infty}^{\infty} F(u,v)e^{-i(xu+yv)}|\omega|d\omega d\theta.$$

Now write

$$F(u,v) = F(\omega\cos\theta, \omega\sin\theta) = R_f(\theta,\omega),$$

where $R_f(\theta,\omega)$ is the FT with respect to $t$ of $r_f(\theta,t)$ so that

$$\int_{-\infty}^{\infty} F(u,v)e^{-i(xu+yv)}|\omega|d\omega = \int_{-\infty}^{\infty} R_f(\theta,\omega)|\omega|e^{-i\omega t}d\omega.$$

The function $h_f(\theta,t)$ defined for $t = x\cos\theta + y\sin\theta$ by

$$h_f(\theta, x\cos\theta + y\sin\theta) = \int_{-\infty}^{\infty} R_f(\theta,\omega)|\omega|e^{-i\omega t}d\omega$$

is the result of a linear filtering of $r_f(\theta,t)$ using a *ramp filter* with transfer function $G(\omega) = |\omega|$. Then

$$f(x,y) = \frac{1}{4\pi^2}\int_0^{\pi} h_f(\theta, x\cos\theta + y\sin\theta)d\theta$$

gives $f(x,y)$ as the result of a *backprojection operator*; for every fixed value of $(\theta,t)$ add $h_f(\theta,t)$ to the current value at the point $(x,y)$ for all $(x,y)$ lying on the straight line determined by $\theta$ and $t$ by $t = x\cos\theta + y\sin\theta$. The final value at a fixed point $(x,y)$ is then the sum of all the values $h_f(\theta,t)$ for those $(\theta,t)$ for which $(x,y)$ is on the line $t = x\cos\theta + y\sin\theta$. It is therefore said that $f(x,y)$ can be obtained by *filtered backprojection* (FBP) of the line integral data.

Knowing that $f(x,y)$ is related to the complete set of line integrals by filtered backprojection suggests that when only finitely many line integrals are available a similar ramp filtering and backprojection can be used to estimate $f(x,y)$; in the clinic this is the most widely used method for the reconstruction of tomographic images.

There is a second way to recover $f(x,y)$ using backprojection and filtering, this time in the reverse order; that is, we backproject the Radon transform and then ramp filter the resulting function of two variables. We begin again with the relation

$$f(x,y) = \frac{1}{4\pi^2}\int_0^{2\pi}\int_0^{\infty} F(u,v)e^{-i(xu+yv)}\omega d\omega d\theta,$$

which we write as

$$f(x,y) = \frac{1}{4\pi^2}\int_0^{2\pi}\int_0^{\infty} \frac{F(u,v)}{\sqrt{u^2+v^2}}\sqrt{u^2+v^2}e^{-i(xu+yv)}\omega d\omega d\theta$$

$$= \frac{1}{4\pi^2} \int_0^{2\pi} \int_0^{\infty} G(u,v)\sqrt{u^2+v^2} e^{-i(xu+yv)} \omega d\omega d\theta, \tag{73.1}$$

using

$$G(u,v) = \frac{F(u,v)}{\sqrt{u^2+v^2}}$$

for $(u,v) \neq (0,0)$. Equation (73.1) expresses $f(x,y)$ as the result of ramp filtering $g(x,y)$, the inverse Fourier transform of $G(u,v)$. We show now that $g(x,y)$ is the backprojection of the function $r_f(\omega,t)$; that is, we show that

$$g(x,y) = \int_0^{\pi} r_f(\theta, x\cos\theta + y\sin\theta) d\theta.$$

From the central slice theorem we know that $g(x,y)$ can be written as

$$g(x,y) = \int_0^{\pi} h_g(\theta, x\cos\theta + y\sin\theta) d\theta,$$

where

$$h_g(\theta, x\cos\theta + y\sin\theta) = \int_{-\infty}^{\infty} R_g(\theta,\omega)|\omega| e^{-i\omega(x\cos\theta + y\sin\theta)} d\omega.$$

Since

$$R_g(\theta,\omega) = G(\omega\cos\theta, \omega\sin\theta)$$

we have

$$\begin{aligned} g(x,y) &= \int_0^{\pi} \int_{-\infty}^{\infty} G(\omega\cos\theta, \omega\sin\theta)|\omega| e^{-i\omega(x\cos\theta + y\sin\theta)} d\omega d\theta \\ &= \int_0^{\pi} \int_{-\infty}^{\infty} F(\omega\cos\theta, \omega\sin\theta) e^{-i\omega(x\cos\theta + y\sin\theta)} d\omega d\theta \\ &= \int_0^{\pi} \int_{-\infty}^{\infty} R_f(\theta,\omega) e^{-i\omega(x\cos\theta + y\sin\theta)} d\omega d\theta \\ &= \int_0^{\pi} r_f(\theta, x\cos\theta + y\sin\theta) d\theta. \end{aligned}$$

This is what we wanted.

We have found that the recovery of $f(x,y)$ from its line integrals can be accomplished using filtering and backprojection in two different ways: one way is to filter the function $r_f(\theta,t)$, viewed as a function of $t$, with a ramp filter, then backproject; the other way is to backproject $r_f(\theta,t)$ first and then filter the resulting function of two variables with a ramp filter in two dimensions. Both of these filtered backprojection methods have their analogs in the processing of actual finite data.

As we noted above, in actual CAT scans only finitely many $\theta$ are used and for each $\theta$ only finitely many $t$ are employed. Therefore at each step along the way we are dealing only with approximations of what the theory would provide. In addition to that, the data we have are not exactly line integrals of $f$ but more precisely integrals of $f$ along narrow strips.

Although the one and two dimensional Fourier transforms do play roles in CAT scan imaging there are better reconstruction methods based on iterative algorithms such as ART and the EMML.

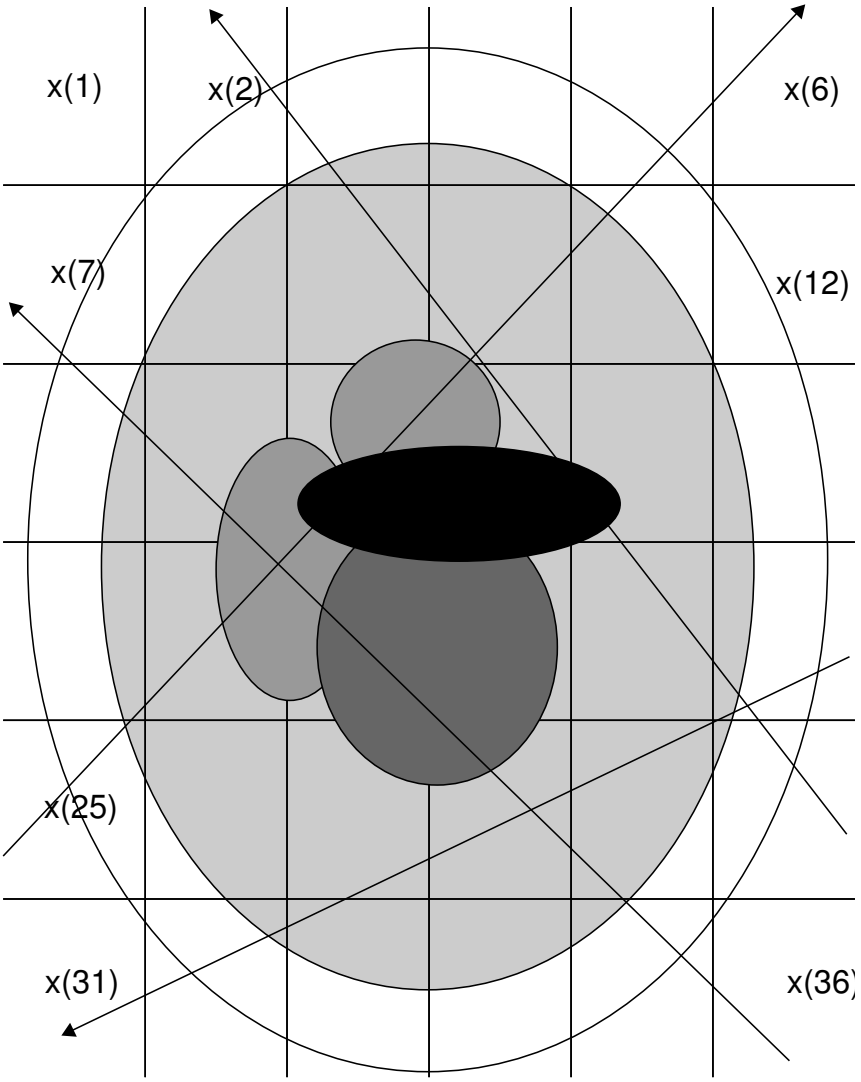



Figure 73.2: The Radon transform for a discretized object.

# Chapter 74

# Scattering

X-ray transmission tomography is based on the reasonable assumption that the rays travel in a straight line through the object, more or less. In other forms of remote sensing this assumption is not reasonable. We consider here the example of the *scattering* of an electromagnetic incident planewave by a dielectric (for more detail see [17], p. 695).

We know from our earlier discussion of the wave equation that a time-harmonic solution $u(t, x, y, z) = e^{i\omega t} g(x, y, z)$ of the wave equation

$$u_{tt} = c^2 \nabla^2 u$$

will have a spatial component $g(x, y, z)$ that satisfies the Helmholtz equation

$$\nabla^2 g(x, y, z) + \frac{\omega^2}{c^2} g(x, y, z) = 0.$$

In that earlier discussion it was assumed that the speed of propagation $c^2$ was constant. In the scalar theory of electromagnetic propagation we find that each Cartesian component function $g(x, y, z)$ of a time-harmonic wave will satisfy the Helmholtz equation, provided that the refractive index $n(x, y, z, \omega)$ is independent of the spatial variables. Otherwise, we must write

$$\nabla^2 g(x, y, z) + \frac{\omega^2}{c^2} n^2(x, y, z, \omega) g(x, y, z) = 0. \tag{74.1}$$

Usually the refractive index is one outside of a localized region $D$ and what we are interested in is the object within that region that is causing the refractive index there not to be one; that is, we want the *scattering potential* function

$$V(x, y, z) = n^2(x, y, z) - 1.$$

For simplicity we no longer show the dependence on $\omega$. We write the spatial variables in vector form as $(x, y, z) = \mathbf{r}$ and let $k = \frac{\omega}{c}$.

Rewriting equation (74.1) as

$$\nabla^2 g(\mathbf{r}) + \frac{\omega^2}{c^2} g(\mathbf{r}) = -V(\mathbf{r}) g(\mathbf{r}) \tag{74.2}$$

we can then view the problem as a non-homogeneous Helmholtz partial differential equation.

The solution of equation (74.2) is the sum of two functions, $g = g_0 + g_s$, where $g_0(\mathbf{r})$ is the incident field that would be present at $\mathbf{r}$ even if the refractive index were constant, and $g_s(\mathbf{r})$ is the *scattered field* due to the deviations in the refractive index. Assuming that the observation location $\mathbf{r}$ is far enough from the object region $D$, the scattered field can be written as

$$g_s(\mathbf{r}) = k^2 \frac{e^{ikr}}{4\pi r} \int_D V(\mathbf{r}') g(\mathbf{r}') e^{-ik\hat{\mathbf{r}}\cdot\mathbf{r}'} d\mathbf{r}', \tag{74.3}$$

where $\hat{\mathbf{r}} = \frac{\mathbf{r}}{||\mathbf{r}||}$ and $r = ||\mathbf{r} - \mathbf{r}'||$. From equation (74.3) we see that we cannot find the scattered field without knowing the entire field. Obviously, some sort of approximation is needed.

The *first Born approximation* is to replace $g(\mathbf{r}')$ in the integrand in equation (74.3) with the incident field $g_0(\mathbf{r}')$. In most cases the incident field is a planewave field of the form

$$g_0(\mathbf{r}) = e^{ik\hat{\mathbf{r}}_0\cdot\mathbf{r}},$$

where $\mathbf{r}_0$ is the direction vector normal to the incident planewave field. With this simplification equation (74.3) becomes

$$g_s(\mathbf{r}) = k^2 \frac{e^{ikr}}{4\pi r} \int_D V(\mathbf{r}') e^{ik\hat{\mathbf{r}}_0\cdot\mathbf{r}'} e^{-ik\hat{\mathbf{r}}\cdot\mathbf{r}'} d\mathbf{r}'. \tag{74.4}$$

The function of $\mathbf{r}$ given by the integral in equation (74.4) is the Fourier transform of the function $V(\mathbf{r})$, evaluated at the point $k(\hat{\mathbf{r}}_0 - \hat{\mathbf{r}})$. As the observation location $r$ changes, we obtain this Fourier transform at points of the sphere of radius $k$ centered at $k\hat{\mathbf{r}}_0$; this is the *Ewald sphere*. By changing the direction of the incident field as well we eventually obtain all the values of the Fourier transform of $V(\mathbf{r})$ in a sphere centered at the origin and having radius $2k$.

# Chapter 75

# A Simple Model for Remote Sensing

Although remote sensing problems differ from one another in many respects, they often share a fundamental aspect that can best be illustrated by a simple model involving dice and bowls of colored marbles.

Suppose that we have a pair of not necessarily fair dice. Each roll of the pair of dice produces a whole number between 2 and 12, but we do not know the probability of each outcome. In fact, this is precisely what we want to estimate. Unfortunately, we do not have direct access to the dice and cannot roll them many times and observe the outcomes. Instead, we have only indirect access.

In addition to the dice there are eleven bowls, numbered 2, 3, ..., 12 to correspond to the possible outcomes of a roll of the pair of dice. Each bowl contains a large number of marbles of various colors: red, blue, green, and so on. For each bowl we know exactly what the proportions of the various colors are; for example, we may know that bowl no. 2 has fifty percent green, twenty percent white and thirty percent red. We then proceed as follows.

The dice are rolled by someone else; we do not observe the outcome. That other person looks at the outcome, goes to the bowl having that number, removes one marble at random, says its color to me and replaces the marble. The only information I get is the color of the marble drawn. This process is repeated many time, so that I end up with a long list of colors. My job is to estimate the probability $p(j)$ that the dice comes up $j$, for each $j = 2, 3, ..., 12$.

Clearly, if two of the bowls, say no. 2 and no. 3, have identical proportions of marbles, I cannot estimate $p(2)$ and $p(3)$ separately. On the other hand, suppose bowl no. 2 has only black marbles, while none of the other

bowls has any black marbles. Then every time I hear that the color was black I know immediately that the bowl was no. 2, so that the dice showed 2. Generally, the more distinct the bowl contents are from each other, the easier the problem becomes.

What we need is an estimation procedure to take us from the list of colors to the probabilities $p(j)$. How can we do this?

This may seem like an artificial problem, but it is basically what is involved in a number of real-world applications, including satellite imaging and medical tomography. The point is this: when we do remote sensing we obtain information about lots of things that are 'out there', but that information is all mixed up. It is sometimes described as the 'cocktail party problem', in which many people are talking at once and we want to hear each of them separately. When the information comes to us in the form of waves, as in optics or acoustics, we often end up with (part of) the Fourier transform of what we really want. Other times we have a mixture probability, such as a Poisson or binomial mixture. But the basic problem is the same: separate out the individual pieces of information.

**Exercise 1:** Simulate the dice-rolling problem described above and use the EMML and SMART algorithms to find the $p(j)$.

# Chapter 76

# Poisson Mixtures

A problem that arises in both the physical sciences and the social sciences is the *mixture problem*. In this chapter we consider a particular case, the *Poisson mixture problem*.

In [89] the authors examine a data set consisting of all the death notices of women aged eighty years or older that appeared in the *Times of London* on each day of three consecutive years. A simple Poisson model for such data would assume that there is a mean $\lambda > 0$ such that the probability $p(n)$ that there would be $n$ deaths on a particular day would be given by the Poisson distribution

$$p(n) = \lambda^n e^{-\lambda}/n!.$$

A more sophisticated model is a Poisson mixture that assumes that there are up to $J$ subgroups of the women, each having their own somewhat different mean values, $\lambda_j$. Then the probability $p(n)$ is given by the Poisson mixture formula

$$p(n) = \sum\nolimits_{j=1}^{J} c_j \lambda_j^n e^{-\lambda_j}/n!,$$

where $c_j \geq 0$ is the proportion of the women belonging to the $j$-th group. The objective is to analyze the data and determine from it accurate estimates of $J$, the means $\lambda_j$ and the proportions $c_j$. For the death notice data the authors show convincingly that $J = 2$ and that the deaths rates are roughly $\lambda_1 = 1.1$ and $\lambda_2 = 2.6$.

In [160] Qian uses the same model of the Poisson mixture to track the changing number of fluorescent molecules from photon count data.

We can extend the finite Poisson mixture model to a continuous mixture, defining the probabilities $p(n)$, for $n = 0, 1, ...$ to be

$$p(n) = \int_0^\infty C(\omega) e^{-\omega} \omega^n / n! \, d\omega,$$

for some nonnegative probability density function $C(\omega)$ having $\int C(\omega)d\omega = 1$. Such a probability model is called a *compound Poisson* distribution with *compounding function* $C(\omega)$. The problem then is to use the data to estimate the function $C(\omega)$, for $0 \leq \omega < \infty$. The sequence $\{p(n)\}$ is sometimes called the *Poisson transform* of the function $C(\omega)$. The finite Poisson mixture then corresponds to a $C(\omega)$ that is a finite sum of delta functions.

The approach commonly used is to derive estimates of the $p(n)$ for as many values of $n$ as the data permits and view these estimates as noisy values of the Poisson transformation of $C(\omega)$. This problem is analogous to the estimation of the Fourier transform $F(\omega)$ from noisy samples of the function $f(x)$ and it is no surprise that some of the same techniques can be employed. In [51] and [50] we used the PDFT and high resolution eigenvector methods to solve the finite Poisson mixture problem.

# Chapter 77

# Hyperspectral Imaging

Hyperspectral image processing provides an excellent example of the need for estimating Fourier transform values from limited data. In this chapter we describe one novel approach, due to Mooney *et al*[144]; the presentation here follows [21], [149]and [110].

In this hyperspectral imaging problem the electromagnetic energy reflected or emitted by a point, such as light reflected from a location on the earth's surface, is passed through a prism to separate the components as to their wavelengths. Due to the dispersion of the different frequency components caused by the prism, these components are recorded in the image plane not at a single spatial location, but at distinct points along a line. Since the received energy comes from a region of points, not a single point, what is received in the image plane is a superposition of different wavelength components associated with different points within the object. The first task is to reorganize the data so that each location in the image plane is associated with all the components of a single point of the object being imaged; this is a Fourier transform estimation problem, which we can solve using band-limited extrapolation.

The points of the image plane are in one-to-one correspondence with points of the object. These spatial locations in the image plane and in the object are discretized into finite two-dimensional grids. Once we have reorganized the data we have, for each grid point in the image plane, a function of wavelength, describing the intensity of each component of the energy from the corresponding grid point on the object. Practical considerations limit the fineness of the grid in the image plane; the resulting discretization of the object is into pixels. In some applications, such as satellite imaging, a single pixel may cover an area several meters on a side. Achieving sub-pixel resolution is one goal of hyperspectral imaging; capturing other subtleties of the scene is another.

Within a single pixel of the object there may well be a variety of object

types, each reflecting or emitting energy differently. The data we now have corresponding to a single pixel is therefore a mixture of the energy associated with each of the sub-objects within the pixel. With prior knowledge of the possible types and their reflective or emissive properties, we can separate the mixture to determine which object types are present within the pixel and to what extent. This mixture problem can be solved using the RBI-EMML method.

Hyperspectral imaging gives rise to several of the issues we discuss in this book. From an abstract perspective the problem is the following: $F$ and $f$ are a Fourier transform pair, as are $G$ and $g$; $F$ and $G$ have finite support; we measure $G$ and want $F$; $g$ determines some, but not all, of the values of $f$. We will have, of course, only finitely many measurements of $G$ from which to estimate values of $g$. Having estimated finitely many values of $g$ we have the corresponding estimates of $f$. We apply band-limited extrapolation of these finitely many values of $f$ to estimate $F$. In fact, once we have estimated values of $F$ we may not be finished; each value of $F$ is a mixture whose individual components may be what we really want. For this unmixing step we use the RBI-EMML algorithm.

The region of the object that we wish to image is described by the two-dimensional spatial coordinate $\mathbf{x} = (x_1, x_2)$. For simplicity, we take these coordinates to be continuous, leaving until the end the issue of discretization. We shall also denote by $\mathbf{x}$ the point in the image plane corresponding to the point $\mathbf{x}$ on the object; the units of distance between two such points in one plane and their corresponding points in the other plane may, of course, be quite different. For each $\mathbf{x}$ we let $F(\mathbf{x}, \lambda)$ denote the intensity of the component at wavelength $\lambda$ of the electromagnetic energy that is reflected from or emitted by location $\mathbf{x}$. We shall assume that $F(\mathbf{x}, \lambda) = 0$ for $(\mathbf{x}, \lambda)$ outside some bounded portion of three-dimensional space.

Consider, for a moment, the case in which the energy sensed by the imaging system comes from a single point $\mathbf{x}$. If the dispersion axis of the prism is oriented according to the unit vector $\mathbf{p}_\theta$, for some $\theta \in [0, 2\pi)$, then the component at wavelength $\lambda$ of the energy from $\mathbf{x}$ on the object is recorded not at $\mathbf{x}$ in the image plane but at the point $\mathbf{x} + \mu(\lambda - \lambda_0)\mathbf{p}_\theta$. Here $\mu > 0$ is a constant and $\lambda_0$ is the wavelength for which the component from point $\mathbf{x}$ of the object is recorded at $\mathbf{x}$ in the image plane.

Now imagine energy coming to the imaging system for all the points within the imaged region of the object. Let $G(\mathbf{x}, \theta)$ be the intensity of the energy received at location $\mathbf{x}$ in the image plane when the prism orientation is $\theta$. It follows from above that

$$G(\mathbf{x}, \theta) = \int_{-\infty}^{+\infty} F(\mathbf{x} - \mu(\lambda - \lambda_0)\mathbf{p}_\theta, \lambda) d\lambda. \tag{77.1}$$

The limits of integration are not really infinite due to the finiteness of the aperture and the focal plane of the imaging system. Our data will consist

of finitely many values of $G(\mathbf{x}, \theta)$, as $\mathbf{x}$ varies over the grid points of the image plane and $\theta$ varies over some finite discretized set of angles.

We begin the image processing by taking the two-dimensional inverse Fourier transform of $G(\mathbf{x}, \theta)$ with respect to the spatial variable $\mathbf{x}$ to get

$$g(\mathbf{y}, \theta) = \frac{1}{(2\pi)^2} \int G(\mathbf{x}, \theta) \exp(-i\mathbf{x} \cdot \mathbf{y}) d\mathbf{x}. \tag{77.2}$$

Inserting the expression for $G$ in equation (77.1) into equation (77.2) we obtain

$$g(\mathbf{y}, \theta) = \exp(i\mu\lambda_0 \mathbf{p}_\theta \cdot \mathbf{y}) \int \exp(-i\mu\lambda \mathbf{p}_\theta \cdot \mathbf{y}) f(\mathbf{y}, \lambda) d\lambda, \tag{77.3}$$

where $f(\mathbf{y}, \lambda)$ is the two-dimensional inverse Fourier transform of $F(\mathbf{x}, \lambda)$ with respect to the spatial variable $\mathbf{x}$. Therefore

$$g(\mathbf{y}, \theta) = \exp(i\mu\lambda_0 \mathbf{p}_\theta \cdot \mathbf{y}) \mathcal{F}(\mathbf{y}, \gamma_\theta), \tag{77.4}$$

where $\mathcal{F}(\mathbf{y}, \gamma)$ denotes the three-dimensional inverse Fourier transform of $F(\mathbf{x}, \lambda)$ and $\gamma_\theta = \mu \mathbf{p}_\theta \cdot \mathbf{y}$. We see then that each value of $g(\mathbf{y}, \theta)$ that we estimate from our measurements provides us with a single estimated value of $\mathcal{F}$.

We use the measured values of $G(\mathbf{x}, \theta)$ to estimate values of $g(\mathbf{y}, \theta)$ guided by the discussion in our earlier chapter on discretization. Having obtained finitely many estimated values of $\mathcal{F}$ we use the support of the function $F(\mathbf{x}, \lambda)$ in three-dimensional space to perform a band-limited extrapolation estimate of the function $F$.

Alternatively, for each fixed $\mathbf{y}$ for which we have values of $g(\mathbf{y}, \theta)$ we use the PDFT or MDFT to solve equation (77.3), obtaining an estimate of $f(\mathbf{y}, \lambda)$ as a function of the continuous variable $\lambda$. Then, for each fixed $\lambda$, we again use the PDFT or MDFT to estimate $F(\mathbf{x}, \lambda)$ from the values of $f(\mathbf{y}, \lambda)$ previously obtained.

Once we have the estimated function $F(\mathbf{x}, \lambda)$ on a finite grid in three-dimensional space we can use the RBI-EMML method, as in [142], to solve the mixture problem and identify the individual object types contained within the single pixel denoted $\mathbf{x}$. For each fixed $\mathbf{x}$ corresponding to a pixel denote by $\mathbf{b} = (b_1, ..., b_I)^T$ the column vector with entries $b_i = F(\mathbf{x}, \lambda_i)$, where $\lambda_i$, $i = 1, ..., I$ constitute a discretization of the wavelength space of those $\lambda$ for which $F(\mathbf{x}, \lambda) > 0$. We assume that this energy intensity distribution vector $\mathbf{b}$ is a superposition of those vectors corresponding to a number of different object types; that is, we assume that

$$\mathbf{b} = \sum_{j=1}^{J} a_j \mathbf{q}_j, \tag{77.5}$$

for some $a_j \geq 0$ and intensity distribution vectors $\mathbf{q}_j$, $j = 1, ..., J$. Each column vector $\mathbf{q}_j$ is a model for what $\mathbf{b}$ would be if there had been only one object type filling the entire pixel. These $\mathbf{q}_j$ are assumed to be known *a priori*. Our objective is to find the $a_j$.

With $Q$ the $I$ by $J$ matrix whose $j$th column is $\mathbf{q}_j$ and $\mathbf{a}$ the column vector with entries $a_j$ we write equation (77.5) as $\mathbf{b} = Q\mathbf{a}$. Since the entries of $Q$ are nonnegative, the entries of $\mathbf{b}$ are positive and we seek a nonnegative solution $a$ we can use any of the entropy-based iterative algorithms discussed earlier. Because of its simplicity of form and speed of convergence our preference is the RBI-EMML algorithm. The recent master's thesis of E. Meidunas [142] discusses just such an application.

# Chapter 78

# Solutions to Selected Exercises

### Complex Numbers

**Exercise 1:** Derive the formula for dividing one complex number in rectangular form by another (non-zero) one.

**Solution:** For any complex numbers $z = (a, b)$ its reciprocal $z^{-1} = (c, d)$ must satisfy the equation $zz^{-1} = (1, 0) = 1$. Therefore $ac - bd = 1$ and $ad + bc = 0$. Multiplying the first equation by $a$ and the second by $b$ and adding, we get $(a^2 + b^2)c = a$, so $c = a/(a^2 + b^2)$. Inserting this in place of $c$ in the second equation gives $d = -b/(a^2 + b^2)$. To divide any complex number $w$ by $z$ we multiply $w$ by $z^{-1}$.

**Exercise 2:** Show that for any two complex numbers $z$ and $w$ we have

$$|zw| \geq \frac{1}{2}(z\overline{w} + \overline{z}w).$$

Hint: Write $|zw|$ as $|z\overline{w}|$.

**Solution:** Using the polar form for $z$ and $w$ it is easy to see that $|zw| = |z\overline{w}|$. With $v = z\overline{w}$ the problem is now to show that $|v| \geq \frac{1}{2}(v + \overline{v})$, or $|v| \geq Re(v)$, which is obvious.

### Complex Exponentials

**Exercise 2:** The *Dirichlet kernel* of size $M$ is defined as

$$D_M(x) = \sum\nolimits_{m=-M}^{M} e^{imx}.$$

Obtain the closed-form expression

$$D_M(x) = \frac{\sin((M+\frac{1}{2})x)}{\sin(\frac{x}{2})};$$

note that $D_M(x)$ is real-valued.
**Hint:** Reduce the problem to that of Exercise 1 by factoring appropriately.

**Solution:** Factor out the term $e^{-i(M+1)x}$ to get

$$D_M(x) = e^{-i(M+1)x} \sum_{m=1}^{2M+1} e^{imx}.$$

Now use the solution to the previous exercise.

**Exercise 3:** Use the formula for $E_M(x)$ to obtain the closed-form expressions

$$\sum_{m=N}^{M} \cos mx = \cos(\frac{M+N}{2}x)\frac{\sin(\frac{M-N+1}{2}x)}{\sin\frac{x}{2}}$$

and

$$\sum_{m=N}^{M} \sin mx = \sin(\frac{M+N}{2}x)\frac{\sin(\frac{M-N+1}{2}x)}{\sin\frac{x}{2}}.$$

**Hint**: Recall that $\cos mx$ and $\sin mx$ are the real and imaginary parts of $e^{imx}$.

**Solution:** Begin with

$$S(x) = \sum_{m=N}^{M} e^{imx}$$

and factor out $e^{i(N-1)x}$ to get

$$S(x) = e^{i(N-1)x} \sum_{m=1}^{M-N+1} e^{imx}.$$

Now apply the formula for $E_M(x)$. Finally, use the fact that the two sums we seek are the real and imaginary parts of $S(x)$.

### Hidden Periodicities

**Exercise 1:** Determine the formulas giving the horizontal and vertical coordinates of the position of a particular rider at an arbitrary time $t$ in the time interval $[0, T]$.

**Solution:** Since the choice of the origin of our coordinate system is arbitrary, we take the origin $(0, 0)$ to be the point on the ground directly under the center of the wheel. The center of the wheel is then located at the point

$(0, R + H)$. Let the rider be at the point $(0 + R\cos\theta, R + H + \sin\theta)$ at time $t = 0$. Since the wheel turns with angular frequency $\omega$ the horizontal position of the rider at any subsequent time will be

$$x(t) = 0 + R\cos(\theta + t\omega)$$

and the vertical position will be

$$y(t) = R + H + R\sin(\theta + t\omega).$$

Note that we can represent the rider's position as a complex number

$$0 + (R + H)i + Re^{i(\theta+t\omega)}.$$

**Exercise 2:** Now find the formulas giving the horizontal and vertical coordinates of the position of a particular rider at an arbitrary time $t$ in the time interval $[0, T]$.

**Solution:** The position of the center of the smaller wheel is the same as that of the rider in the previous exercise; that is,

$$x(t) = 0 + R_1\cos(\theta_1 + t\omega_1)$$

and

$$y(t) = R_1 + H + R_1\sin(\theta_1 + t\omega_1).$$

The rider's position deviates from that of the center of the smaller wheel in the same way that the rider's position in the previous exercise deviated from the center of the single large wheel. Therefore, the horizontal position of the rider now is

$$x(t) = 0 + R_1\cos(\theta_1 + t\omega_1) + R_2\cos(\theta_2 + t\omega_2)$$

and the vertical position is

$$y(t) = R_1 + H + R_1\sin(\theta_1 + t\omega_1) + R_2\sin(\theta_2 + t\omega_2).$$

Again, we can represent the position as a complex number:

$$0 + (R + H)i + R_1e^{i(\theta_1+t\omega_1)} + R_2e^{i(\theta_2+t\omega_2)}.$$

**Exercise 3:** Repeat the previous exercise, but for the case of $J$ nested wheels.

**Solution:** Reasoning as above, and using the complex representation, we find the position to be

$$0 + (R + H)i + \sum_{j=1}^{J} R_je^{i(\theta_j+t\omega_j)}.$$

**Convolution and the Vector DFT**

**Exercise 1:** Let $\mathbf{F} = vDFT_{\mathbf{f}}$ and $\mathbf{D} = vDFT_{\mathbf{d}}$. Define a third vector $\mathbf{E}$ having for its $k$-th entry $E_k = F_k D_k$, for $k = 0, ..., N-1$. Show that $\mathbf{E}$ is the vDFT of the vector $\mathbf{f} * \mathbf{d}$.

**Solution:** For notational convenience we define $d_{k-N} = d_k$, for $k = 0, 1, ..., N$. Then we can write

$$(\mathbf{f} * \mathbf{d})_n = \sum\nolimits_{m=0}^{N-1} f_m d_{n-m}.$$

Using this extended notation we find that the sum

$$\sum\nolimits_{n=0}^{N-1} d_{n-m} e^{i(n-m)2\pi k/N}$$

does not depend on $m$ and is equal to

$$\sum\nolimits_{j=0}^{N-1} d_j e^{2\pi jki/N},$$

which is $D_k$. The vDFT of the vector $\mathbf{f} * \mathbf{d}$ has for its $k$-th entry the quantity

$$\sum\nolimits_{n=0}^{N-1} (\mathbf{f} * \mathbf{d})_n e^{2\pi ink/N},$$

which we write as the double sum

$$\sum\nolimits_{n=0}^{N-1} \sum\nolimits_{m=0}^{N-1} f_m d_{n-m} e^{2\pi ink/N}.$$

Now we simply reverse the order of summation, write

$$e^{2\pi ink/N} = e^{2\pi imk/N} e^{2\pi i(n-m)k/N}$$

and use the fact already shown that the sum on $n$ is independent of $m$. We then have that the $k$-th entry is

$$\sum\nolimits_{m=0}^{N-1} f_m e^{2\pi imk/N} \sum\nolimits_{j=0}^{N-1} d_j e^{2\pi ijk/N} = F_k D_k.$$

**Exercise 2:** Let $G$ be the $N$ by $N$ matrix whose entries are $G_{jk} = e^{i(j-1)(k-1)2\pi/N}$. The matrix $G$ is sometimes called the *DFT matrix*. Show that the inverse of $G$ is $G^{-1} = \frac{1}{N} G^\dagger$, where $G^\dagger$ is the conjugate transpose of the matrix $G$. Then $\mathbf{f} * \mathbf{d} = G^{-1}\mathbf{E} = \frac{1}{N} G^\dagger \mathbf{E}$.

**Solution:** Compute the entry of the matrix $G^\dagger G$ in the $m$-th row, $n$-th column. Use the definition of matrix multiplication to express this entry

as a sum of the same type as in the definition of $E_M(x)$. Consider what happens when $m = n$ and when $m \neq n$.

**Cauchy's Inequality**

**Exercise 1:** Use Cauchy's inequality to show that

$$||\mathbf{u} + \mathbf{v}|| \leq ||\mathbf{u}|| + ||\mathbf{v}||;$$

this is called the *triangle inequality.*

**Solution:** We have

$$||\mathbf{u} + \mathbf{v}||^2 = (\mathbf{u} + \mathbf{v}) \cdot (\mathbf{u} + \mathbf{v}) = \mathbf{u} \cdot \mathbf{u} + \mathbf{u} \cdot \mathbf{v} + \mathbf{v} \cdot \mathbf{u} + \mathbf{v} \cdot \mathbf{v}$$

$$= ||\mathbf{u}||^2 + ||\mathbf{v}||^2 + \mathbf{u} \cdot \mathbf{v} + \overline{\mathbf{u} \cdot \mathbf{v}} = ||\mathbf{u}||^2 + ||\mathbf{v}||^2 + 2Re(\mathbf{u} \cdot \mathbf{v}).$$

Also we have

$$(||\mathbf{u}|| + ||\mathbf{v}||)^2 = ||\mathbf{u}||^2 + ||\mathbf{v}||^2 + 2||\mathbf{u}||\,||\mathbf{v}||.$$

Now use Cauchy's inequality to conclude that

$$Re(\mathbf{u} \cdot \mathbf{v}) \leq |Re(\mathbf{u} \cdot \mathbf{v})| \leq |\mathbf{u} \cdot \mathbf{v}| \leq ||\mathbf{u}||\,||\mathbf{v}||.$$

**Orthogonal Vectors**

**Exercise 1:** Use the Gram-Schmidt approach to find a third vector in $R^3$ orthogonal to both $(1, 1, 1)$ and $(1, 0, -1)$.

**Solution:** Let the third vector be $\mathbf{v} = (a, b, c)$. Begin by selecting a vector that cannot be written as $\alpha(1, 1, 1) + \beta(1, 0, -1)$. How can we be sure we have such a vector? Notice that such a vector must have the form $(\alpha + \beta, \alpha, \alpha - \beta)$, so the middle entry is the average of the other two. Now take any vector that does not have this property; let's take $(1, 2, 2)$. We know that we can write $(1, 2, 2)$ as

$$(1, 2, 2) = \alpha(1, 1, 1) + \beta(1, 0, -1) + \gamma(a, b, c),$$

for some choices of $\alpha$, $\beta$ and $\gamma$. Let's find $\alpha$ and $\beta$. Take the dot product of both sides of the last equation with the vector $(1, 1, 1)$ to get

$$5 = (1, 1, 1) \cdot (1, 2, 2) = \alpha(1, 1, 1) \cdot (1, 1, 1) = 3\alpha.$$

So $\alpha = 5/3$. Now take the inner product of both sides with $(1, 0, -1)$ to get

$$-1 = (1, 0, -1) \cdot (1, 2, 2) = \beta(1, 0, -1) \cdot (1, 0, -1) = 2\beta.$$

Therefore, $\beta = -1/2$. So we now have

$$(1,2,2) - \frac{5}{3}(1,1,1) + \frac{1}{2}(1,0,-1) = (-\frac{1}{6}, \frac{1}{3}, -\frac{1}{6}) = \frac{-1}{6}(1,-2,1).$$

We can then take $\gamma = \frac{-1}{6}$ and $\mathbf{v} = (a,b,c) = (1,-2,1)$.

**Discrete Linear Filters**

**Exercise 1:** Show that $F(\omega) = G(\omega)H(\omega)$ for all $\omega$.

**Solution:** Using the definition of $F(\omega)$ and $f_n$ we write

$$F(\omega) = \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} g_m h_{n-m} e^{i\omega m} e^{i\omega(n-m)}$$

$$= \sum_{m=-\infty}^{\infty} g_m [\sum_{n=-\infty}^{\infty} h_{n-m} e^{i\omega(n-m)}] e^{i\omega m}.$$

Since the inner sum

$$\sum_{n=-\infty}^{\infty} h_{n-m} e^{i\omega(n-m)} = \sum_{k=-\infty}^{\infty} h_k e^{i\omega k}$$

does not really depend on the index $m$ it can be taken outside the sum over that index.

**Exercise 2:** The *three-point moving average* filter is defined as follows: given the input sequence $\{h_n, n = -\infty, ..., \infty\}$ the output sequence is $\{f_n, n = -\infty, ..., \infty\}$, with

$$f_n = (h_{n-1} + h_n + h_{n+1})/3.$$

Let $g_k = 1/3$, if $k = 0, 1, -1$ and $g_k = 0$, otherwise. Then we have

$$f_n = \sum_{k=-\infty}^{\infty} g_k h_{n-k},$$

so that $f$ is the *discrete convolution* of $h$ and $g$. Let

$$F(\omega) = \sum_{n=-\infty}^{\infty} f_n e^{in\omega},$$

for $\omega$ in the interval $[-\pi, \pi]$, be the Fourier series for the sequence $f$; similarly define $G$ and $H$. To recover $h$ from $f$ we might proceed as follows: calculate $F$, then divide $F$ by $G$ to get $H$, then compute $h$ from $H$; does this always work? If we let $h$ be the sequence $\{..., 1, 1, 1, ...\}$ then $f = h$; if we take $h$ to be the sequence $\{..., 3, 0, 0, 3, 0, 0, ...\}$ then we again get $f = \{..., 1, 1, 1, ...\}$. Therefore, we cannot expect to recover $h$ from $f$ in general. We know that $G(\omega) = \frac{1}{3}(1 + 2\cos(\omega))$; what does this have to do with the problem of recovering $h$ from $f$?

**Solution:** If the input sequence is $h = \{..., 2, -1, -1, 2, -1, -1, ...\}$ then the output sequence is $f = \{..., 0, 0, 0, 0, 0, ...\}$. Since

$$G(\omega) = \frac{1}{3}(1 + 2\cos(\omega)),$$

the zeros of $G(\omega)$ are at $\omega = \frac{2\pi}{3}$ and $\omega = -\frac{2\pi}{3}$. Consider the sequence defined by

$$h_n = e^{in\frac{2\pi}{3}} + e^{-in\frac{2\pi}{3}};$$

this is the sequence $\{..., 2, -1, -1, 2, -1, -1, ...\}$. This sequence consists of two complex exponential components, with associated frequencies at precisely the roots of $G(\omega)$. The three-point moving average has the output of all zeros because the function $G(\omega)$ has *nulled out* the only two sinusoidal components in $h$.

**Exercise 3:** Let $f$ be the autocorrelation sequence for $g$. Show that $f_{-n} = \overline{f}_n$ and $f_0 \geq |f_n|$ for all $n$.

**Solution:** The first part follows immediately from the definition of the autocorrelation. The second part is a consequence of the Cauchy-Schwarz inequality for infinite sequences.

**Inner Products**

**Exercise 1:** Find polynomial functions $f(x)$, $g(x)$ and $h(x)$ that are orthogonal on the interval $[0, 1]$ and have the property that every polynomial of degree two or less can be written as a linear combination of these three functions.

**Solution:** Let's find $f(x) = a$, $g(x) = bx + c$ and $h(x) = dx^2 + ex + k$ that do the job. Clearly, we can start by taking $f(x) = 1$. Then

$$0 = \int_0^1 1g(x)dx = b\int_0^1 xdx + c = \frac{b}{2} + c$$

says that $b = -2c$. Let $c = 1$ so that $b = -2$ and $g(x) = -2x + 1$. Then

$$0 = \int_0^1 1h(x)dx = \frac{d}{3} + \frac{e}{2} + k$$

and

$$0 = \int_0^1 g(x)h(x)dx = \int_0^1 (-2x + 1)(dx^2 + ex + k)dx.$$

Therefore we have

$$0 = \frac{-2}{4}d + \frac{-2}{3}e + \frac{-2}{2}k + \frac{d}{3} + \frac{e}{2} + k.$$

We can let $d = 6$, from which it follows that $e = -6$ and $k = 1$. So the three polynomials are $f(x) = 1$, $g(x) = -2x + 1$ and $h(x) = 6x^2 - 6x + 1$. To show that any quadratic polynomial can be written as a sum of these three, take an arbitrary quadratic, $ax^2 + bx + c$ and write

$$ax^2 + bx + c = \alpha f(x) + \beta g(x) + \gamma h(x).$$

Then show that you can solve for the $\alpha$, $\beta$ and $\gamma$ in terms of the $a$, $b$ and $c$.

**Exercise 2:** Show that the functions $e^{inx}$, $n$ an integer, are orthogonal on the interval $[-\pi, \pi]$. Let $f(x)$ have the Fourier expansion

$$f(x) = \sum_{n=-\infty}^{\infty} a_n e^{inx},\ |x| \leq \pi.$$

Use orthogonality to find the coefficients $a_n$.

**Solution:** Compute the integral

$$\int_{-\pi}^{\pi} e^{inx} e^{-imx} dx$$

and show that it is zero for $m \neq n$. To find the coefficients multiply both sides by $e^{-imx}$ and integrate; on the left we get $\int_{-\pi}^{\pi} f(x) e^{-imx} dx$ and on the right we get $2\pi a_m$.

**Fourier Transforms and Fourier Series**

**Exercise 1:** Use the orthogonality of the functions $e^{im\omega}$ on $[-\pi, \pi]$ to establish *Parseval's equation*:

$$\langle f, g \rangle = \sum_{m=-\infty}^{\infty} f_m \overline{g_m} = \int_{-\pi}^{\pi} F(\omega) \overline{G(\omega)} d\omega / 2\pi,$$

from which it follows that

$$\langle f, f \rangle = \int_{-\infty}^{\infty} |F(\omega)|^2 d\omega / 2\pi.$$

**Solution:** Since we have

$$F(\omega) = \sum_{m=-\infty}^{\infty} f_m e^{im\omega},\ |\omega| \leq \pi,$$

with a similar expression for $G(\omega)$, we have

$$\langle F, G \rangle = \int_{-\pi}^{\pi} F(\omega) \overline{G(\omega)} d\omega / 2\pi$$

$$= \int_{-\pi}^{\pi} \sum_{m=-\infty}^{\infty} f_m e^{im\omega} \sum_{n=-\infty}^{\infty} \overline{g_n} e^{-in\omega} d\omega/2\pi$$
$$= \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} f_m \overline{g_n} \int_{-\pi}^{\pi} e^{i(n-m)\omega} d\omega/2\pi,$$

which equals

$$\sum_{m=-\infty}^{\infty} f_m \overline{g_m} = \langle f, g \rangle$$

because the integral is zero unless $m = n$.

**Exercise 3:** Let $f(x)$ be defined for all real $x$ and let $F(\omega)$ be its FT. Let

$$g(x) = \sum_{k=-\infty}^{\infty} f(x + 2\pi k),$$

assuming the sum exists. Show that $g$ is a $2\pi$ -periodic function. Compute its Fourier series and use it to derive the *Poisson summation formula*:

$$\sum_{k=-\infty}^{\infty} f(2\pi k) = \frac{1}{2\pi} \sum_{n=-\infty}^{\infty} F(n).$$

**Solution:** Clearly $g(x + 2\pi) = g(x)$ for all $x$, so $g(x)$ is $2\pi$-periodic. The Fourier series for $g(x)$ is

$$g(x) = \sum_{n=-\infty}^{\infty} a_n e^{inx},$$

where

$$a_n = \int_{-\pi}^{\pi} g(x) e^{-inx} dx/2\pi$$
$$= \int_{-\pi}^{\pi} \sum_{k=-\infty}^{\infty} f(x + 2\pi k) e^{-inx} dx/2\pi$$
$$= \sum_{k=-\infty}^{\infty} \int_{-\pi}^{\pi} f(x + 2\pi k) e^{-inx} dx/2\pi$$
$$= \sum_{k=-\infty}^{\infty} e^{i2\pi nk} \int_{-\pi}^{\pi} f(t) e^{-int} dt/2\pi$$
$$= \sum_{k=-\infty}^{\infty} \int_{-\pi}^{\pi} f(t) e^{-in(t-2\pi k)} dt/2\pi$$
$$= \sum_{k=-\infty}^{\infty} \int_{-\pi+2\pi k}^{\pi+2\pi k} f(t) e^{-int} dt/2\pi$$

$$= \int_{-\infty}^{\infty} f(t)e^{-int}dt/2\pi = \frac{1}{2\pi}F(-n).$$

Therefore

$$g(x) = \frac{1}{2\pi}\sum\nolimits_{n=-\infty}^{\infty} F(-n)e^{inx}.$$

Now let $x = 0$ to get

$$g(0) = \sum_{k=-\infty}^{\infty} f(2\pi k) = \frac{1}{2\pi}\sum\nolimits_{n=-\infty}^{\infty} F(-n).$$

**More on the Fourier Transform**

**Exercise 1:** Let $F(\omega)$ be the FT of the function $f(x)$. Use the definitions of the FT and IFT to establish the following basic properties of the Fourier transform operation:

**Differentiation:** The FT of the $n$-th derivative, $f^{(n)}(x)$ is $(-i\omega)^n F(\omega)$. The IFT of $F^{(n)}(\omega)$ is $(ix)^n f(x)$.

**Solution:** Begin with the inverse FT equation

$$f(x) = \int F(\omega)e^{-ix\omega}d\omega/2\pi$$

and differentiate with respect to $x$ inside the integral sign $n$ times.

**Convolution in $x$:** Let $f, F$, $g, G$ and $h, H$ be FT pairs, with

$$h(x) = \int f(y)g(x-y)dy,$$

so that $h(x) = (f * g)(x)$ is the convolution of $f(x)$ and $g(x)$. Then $H(\omega) = F(\omega)G(\omega)$.

**Solution:** From the definitions of $F(\omega)$ and $G(\omega)$ we have

$$F(\omega)G(\omega) = \int f(y)e^{iy\omega}dy \int g(t)e^{it\omega}dt$$

$$= \int\int f(y)g(t)e^{i(y+t)\omega}dy\,dt.$$

Changing variables by setting $x = y + t$, so $t = x - y$ and $dt = dx$ we get

$$= \int\int f(y)g(x-y)e^{ix\omega}dydx$$

$$= \int [\int f(y)g(x-y)dy]\, e^{ix\omega}dx = \int h(x)e^{ix\omega}dx = H(\omega).$$

**Exercise 2:** Show that the Fourier transform of $f(x) = e^{-\alpha^2 x^2}$ is $F(\omega) = \frac{\sqrt{\pi}}{\alpha}e^{-(\frac{\omega}{2\alpha})^2}$.

**Solution:** From the FT formula

$$F(\omega) = \int f(x)e^{ix\omega}dx = \int e^{-\alpha^2 x^2}e^{ix\omega}dx$$

we have

$$F'(\omega) = \int ixe^{-\alpha^2 x^2}e^{ix\omega}dx.$$

Integrating by parts gives

$$F'(\omega) = -\frac{\omega}{2\alpha^2}F(\omega),$$

so that

$$F(\omega) = c\exp(-\frac{\omega^2}{4\alpha^2}).$$

To find $c$ we set $\omega = 0$. Then

$$c = F(0) = \int e^{-\alpha^2 x^2}dx = \frac{\sqrt{\pi}}{\alpha}.$$

This last integral occurs frequently in texts on probability theory, in the discussion of normal random variables and is obtained by using a trick involving polar coordinates.

**Exercise 3:** Calculate the FT of the function $f(x) = u(x)e^{-ax}$, where $a$ is a positive constant.

**Solution:** We have

$$F(\omega) = \int_0^\infty e^{-ax}e^{ix\omega}dx = \int_0^\infty e^{(i\omega-a)x}dx$$

$$= \frac{1}{i\omega - a}[\lim_{X\to+\infty}(e^{(i\omega-a)X}) - e^{(i\omega-a)(0)}] = \frac{1}{a-i\omega}.$$

**Exercise 4:** Calculate the FT of $f(x) = \chi_X(x)$.

**Solution:** We now have

$$F(\omega) = \int_{-X}^{X} e^{ix\omega}dx = \int_{-X}^{X}\cos(x\omega)dx$$

$$= \frac{2}{\omega}\sin(X\omega).$$

**Exercise 5:** Show that the IFT of the function $F(\omega) = 2i/\omega$ is $f(x) = \text{sgn}(x)$. Hints: write the formula for the inverse Fourier transform of $F(\omega)$ as

$$f(x) = \frac{1}{2\pi}\int_{-\infty}^{+\infty}\frac{2i}{\omega}\cos\omega x d\omega - \frac{i}{2\pi}\int_{-\infty}^{+\infty}\frac{2i}{\omega}\sin\omega x d\omega$$

which reduces to

$$f(x) = \frac{1}{\pi}\int_{-\infty}^{+\infty}\frac{1}{\omega}\sin\omega x d\omega,$$

since the integrand of the first integral is odd. For $x > 0$ consider the Fourier transform of the function $\chi_x(t)$. For $x < 0$ perform the change of variables $u = -x$.

**Solution:** See the hints.

**Exercise 6:** Use the fact that $\text{sgn}(x) = 2u(x) - 1$ and the previous exercise to show that $f(x) = u(x)$ has the FT $F(\omega) = i/\omega + \pi\delta(\omega)$.

**Solution:** From the previous exercise we know that the FT of $f(x) = \text{sgn}(x)$ is $F(\omega) = \frac{2i}{\omega}$. We also know that the FT of the function $f(x) = 1$ is $F(\omega) = 2\pi\delta(\omega)$. Writing

$$u(x) = \frac{1}{2}(\text{sgn}(x) + 1)$$

we find that the FT of $u(x)$ is $\frac{i}{\omega} + \pi\delta(\omega)$.

**Exercise 7:** Let $F(\omega) = R(\omega) + iX(\omega)$, where $R$ and $X$ are real-valued functions, and similarly, let $f(x) = f_1(x) + if_2(x)$, where $f_1$ and $f_2$ are real-valued. Find relationships between the pairs $R$,$X$ and $f_1$,$f_2$.

**Solution:** From $F(\omega) = R(\omega) + iX(\omega)$ and

$$F(\omega) = \int f(x)e^{ix\omega}dx = \int (f_1(x) + if_2(x))e^{ix\omega}dx$$

we get

$$R(\omega) = \int f_1(x)\cos(x\omega) - f_2(x)\sin(x\omega)dx$$

and

$$X(\omega) = \int f_1(x)\sin(x\omega) + f_2(x)\cos(x\omega)dx.$$

**Exercise 8:** Let $f, F$ be a FT pair. Let $g(x) = \int_{-\infty}^{x} f(y)dy$. Show that the FT of $g(x)$ is $G(\omega) = \pi F(0)\delta(\omega) + \frac{F(\omega)}{i\omega}$.

**Solution:** Since $g(x)$ is the convolution of $f(x)$ and the Heaviside function $u(x)$ it follows that

$$G(\omega) = F(\omega)(\frac{i}{\omega} + \pi\delta(\omega))$$

$$= i\frac{F(\omega)}{\omega} + \pi F(0)\delta(\omega).$$

**Exercise 9:** Let $f(x), F(\omega)$ and $g(x), G(\omega)$ be Fourier transform pairs. Establish the Parseval-Plancherel equation

$$\langle f, g\rangle = \int f(x)\overline{g(x)}dx = \frac{1}{2\pi}\int F(\omega)\overline{G(\omega)}d\omega.$$

**Solution:** Begin by inserting

$$f(x) = \int F(\omega)e^{-ix\omega}d\omega/2\pi$$

and

$$g(x) = \int G(\alpha)e^{-ix\alpha}d\alpha/2\pi$$

into

$$\int f(x)\overline{g(x)}dx$$

and interchanging the order of integration to get

$$\int f(x)\overline{g(x)}dx = (\frac{1}{2\pi})^2\int\int F(\omega)\overline{G(\alpha)}[\int e^{ix(\omega-\alpha)}dx]d\omega d\alpha.$$

The innermost integral is

$$\int e^{ix(\omega-\alpha)}dx = \delta(\omega-\alpha)$$

so we get

$$\int f(x)\overline{g(x)}dx = (\frac{1}{2\pi})^2\int F(\omega)[\int \overline{G(\alpha)}\delta(\omega-\alpha)d\alpha/2\pi]d\omega/2\pi$$

$$= \int F(\omega)\overline{G(\omega)}d\omega/2\pi.$$

**Exercise 10:** Show that, if $f$ is causal, then $R$ and $X$ are related; specifically, show that $X$ is the *Hilbert transform* of $R$, that is,

$$X(\omega) = 2\int_{-\infty}^{\infty} \frac{R(\alpha)}{\omega - \alpha} d\alpha.$$

**Solution:** Since $f(x) = 0$ for $x < 0$ we have $f(x)\text{sgn}(x) = f(x)$. Taking the FT of both sides and applying the convolution theorem, we get

$$F(\omega) = 2i \int F(\alpha) \frac{1}{\omega - \alpha} d\alpha / 2\pi.$$

Now compute the real and imaginary parts of both sides.

**Exercise 11:** Compute $\mathcal{F}(z)$ for $f(x) = u(x)$, the Heaviside function. Compare $\mathcal{F}(-i\omega)$ with the FT of $u$.

**Solution:** Let $z = a + bi$, where $a > 0$. For $f(x) = u(x)$ the integral becomes

$$\mathcal{F}(z) = \int_0^{\infty} e^{-zx} dx = \frac{-1}{z}[0 - 1] = \frac{1}{z}.$$

Inserting $z = -i\omega$ we get

$$\frac{i}{\omega} = \mathcal{F}(-i\omega) = \int u(x) e^{ix\omega} dx.$$

The integral is the Fourier transform of the Heaviside function $u(x)$, which is not quite equal to $\frac{1}{\omega}$. The point here is that we erroneously evaluated the Laplace transform integral at a point $z$ whose real part is not positive.

**The Uncertainty Principle**

**Exercise 1:** Show that, if the inequality is an equation for some $f$, then $f'(x) = kxf(x)$, so that $f(x) = e^{-\alpha^2 x^2}$ for some $\alpha > 0$.

**Solution:** We get equality in the Cauchy-Schwarz inequality if and only if

$$f'(x) = cxf(x),$$

for some constant. Solving this differential equation by separation of variables we obtain the solution

$$f(x) = K \exp(\frac{c}{2} x^2).$$

Since we want $\int f(x)dx$ to be finite, we must select $c < 0$.

**Wavelets**

**Exercise 1:** Let $u(x) = 1$ for $0 \leq x < \frac{1}{2}$, $u(x) = -1$ for $\frac{1}{2} \leq x < 1$ and zero otherwise. Show that the functions $u_{jk}(x) = u(2^j x - k)$ are mutually orthogonal on the interval $[0, 1]$, where $j = 0, 1, ...$ and $k = 0, 1, ..., 2^j - 1$.

**Solution:** Consider $u_{jk}$ and $u_{mn}$, where $m \geq j$. If $m = j$ and $k \neq n$ then the supports are disjoint and the functions are orthogonal. If $m > j$ and the supports are disjoint, then, again, the functions are orthogonal. So suppose that $m > j$ and the supports are not disjoint. Then the support of $u_{mn}$ is a subset of the support of $u_{jk}$. On that subset $u_{jk}(x)$ is constant, while $u_{mn}(x)$ is that constant for half of the $x$ and is the negative of that constant for the other half; therefore the inner product is zero.

**The FT in Higher Dimensions**

**Exercise 1:** Show that if $f$ is radial then its FT $F$ is also radial. Find the FT of the radial function $f(x, y) = \frac{1}{\sqrt{x^2+y^2}}$.

**Solution:** Inserting $f(r, \theta) = g(r)$ in the equation for $F(\rho, \omega)$ we obtain

$$F(\rho, \omega) = \int_0^\infty \int_{-\pi}^{\pi} g(r) e^{ir\rho\cos(\theta-\omega)} r dr d\theta$$

or

$$F(\rho, \omega) = \int_0^\infty r g(r) [\int_{-\pi}^{\pi} e^{ir\rho\cos(\theta-\omega)} d\theta] dr.$$

Although it does not appear to be, the inner integral is independent of $\omega$; if we replace the variable $\theta - \omega$ with $\theta$ we have $\cos\theta$ is the exponent, $d(\theta - \omega) = d\theta$ remains unchanged, and the limits of integration become $-\pi + \omega$ to $\pi + \omega$. But since the integrand is $2\pi$-periodic, this integral is the same as the one from $-\pi$ to $\pi$.

To find the FT of the radial function $f(x, y) = \frac{1}{\sqrt{x^2+y^2}}$, we write it in polar coordinates as $f(r, \theta) = g(r) = 1/r$. Then

$$H(\rho) = 2\pi \int_0^\infty J_0(r\rho) dr = \frac{2\pi}{\rho} \int_0^\infty J_0(r\rho)\rho dr = \frac{2\pi}{\rho},$$

since $\int J_0(x)dx = 1$; the basic facts about the Bessel function $J_0(x)$ can be found in most texts on differential equations. So, for the two-dimensional case, the radial function $f(r, \theta) = g(r) = \frac{1}{r}$ is, except for a scaling, its own Fourier transform, as is the case for the standard Gaussian function in one dimension.

**Discretization**

**Exercise 1:** In the top half of Figure 78.1 the FT graph shows values of $0.5\sin(\frac{\pi}{2}n)/(\frac{\pi}{2}n)$ for $0 \leq n \leq 63$. The FFT graph shows estimates given by fft values obtained from 128 equispaced sampled of $\chi_{[\frac{\pi}{2},\frac{3\pi}{2}]}(\omega)$ on $[0, 2\pi]$. The bottom half displays the values for $n = 64$ to $n = 127$.

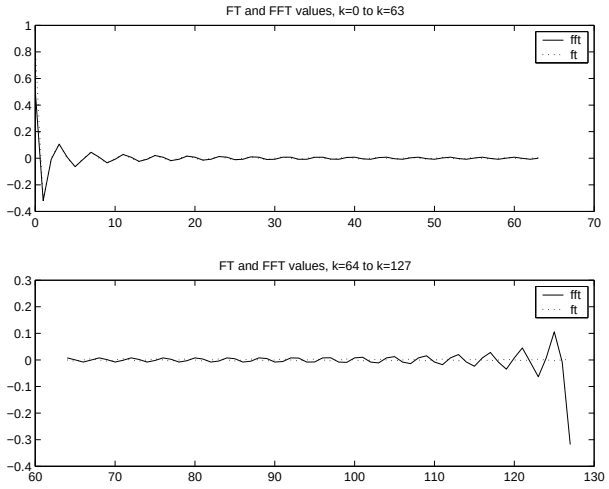



Figure 78.1: FT and FFT comparison

**Fourier Transform Estimation**

**Exercise 1:** Use the orthogonality principle to show that the DFT minimizes the distance

$$\int_{-\pi}^{\pi} |F(\omega) - \sum\nolimits_{m=1}^{M} a_m e^{im\omega}|^2 d\omega.$$

**Solution:** The orthogonality principle asserts that, for the optimal choice of the $a_n$, we have

$$\int_{-\pi}^{\pi} (F(\omega) - \sum\nolimits_{m=1}^{M} a_m e^{im\omega}) e^{-in\omega} d\omega = 0,$$

for $n = 1, ..., M$. It follows, much as in the previous exercise, that $a_n = f(n)$.

**Exercise 2:** Suppose that $0 < \Omega$ and $F(\omega) = 0$ for $|\omega| > \Omega$. Let $f(x)$ be the inverse Fourier transform of $F(\omega)$ and suppose that the data is

$f(x_m)$, $m = 1, ..., M$. Use the orthogonality principle to find the coefficients $a_m$ that minimize the distance

$$\int_{-\Omega}^{\Omega} |F(\omega) - \sum\nolimits_{m=1}^{M} a_m e^{ix_m\omega}|^2 d\omega.$$

Show that the resulting estimate of $F(\omega)$ is consistent with the data.

**Solution:** The orthogonality principle tells us that, for the optimal choice of the $a_m$, we have

$$\int_{-\Omega}^{\Omega} (F(\omega - \sum\nolimits_{m=1}^{M} a_m e^{ix_m\omega}) e^{-ix_n\omega} d\omega = 0,$$

for $n = 1, 2, ..., M$. This says that, for these $n$,

$$f(x_n) = \sum\nolimits_{m=1}^{M} a_m \int_{-\Omega}^{\Omega} e^{i(x_m - x_n)\omega} d\omega/2\pi$$

or

$$f(x_n) = \sum\nolimits_{m=1}^{M} a_m \frac{\sin \Omega(x_m - x_n)}{\pi(x_m - x_n)}.$$

The inverse Fourier transform of the function

$$F_\Omega(\omega) = \chi_\Omega(\omega) \sum\nolimits_{m=1}^{M} a_m e^{ix_m\omega}$$

is

$$f_\Omega(x) = \sum\nolimits_{m=1}^{M} a_m \frac{\sin \Omega(x_m - x)}{\pi(x_m - x)};$$

setting $x = x_n$ we see that $f_\Omega(x_n) = f(x_n)$, for $n = 1, ..., M$, so the optimal estimate is data consistent.

### More on Bandlimited Extrapolation

**Exercise 1:** The purpose of this exercise is to show that, for an Hermitian nonnegative-definite $M$ by $M$ matrix $Q$, a norm-one eigenvector $\mathbf{u}^1$ of $Q$ associated with its largest eigenvalue, $\lambda_1$, maximizes the quadratic form $\mathbf{a}^\dagger Q\mathbf{a}$ over all vectors $\mathbf{a}$ with norm one. Let $Q = ULU^\dagger$ be the eigenvector decomposition of $Q$, where the columns of $U$ are mutually orthogonal eigenvectors $\mathbf{u}^n$ with norms equal to one, so that $U^\dagger U = I$, and $L = diag\{\lambda_1, ..., \lambda_M\}$ is the diagonal matrix with the eigenvalues of $Q$ as its entries along the main diagonal. Assume that $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_M$. Then maximize

$$\mathbf{a}^\dagger Q\mathbf{a} = \sum_{n=1}^{M} \lambda_n |\mathbf{a}^\dagger \mathbf{u}^n|^2,$$

subject to the constraint

$$\mathbf{a}^\dagger \mathbf{a} = \mathbf{a}^\dagger U^\dagger U \mathbf{a} = \sum_{n=1}^{M} |\mathbf{a}^\dagger \mathbf{u}^n|^2 = 1.$$

**Solution:** Since we have

$$\sum\nolimits_{n=1}^{M} |\mathbf{a}^\dagger \mathbf{u}^n|^2 = 1$$

the sum

$$\sum_{n=1}^{M} \lambda_n \, |\mathbf{a}^\dagger \mathbf{u}^n|^2$$

is a convex combination of the nonnegative numbers $\lambda_n$. Such a convex combination must be no greater than the greatest $\lambda_n$, which is $\lambda_1$. But it can equal $\lambda_1$ if we select the unit vector $\mathbf{a}$ to be $\mathbf{a} = \mathbf{u}^1$. So the greatest value $\mathbf{a}^\dagger Q \mathbf{a}$ can attain is $\lambda_1$.

**Exercise 2:** Show that for the sinc matrix $Q_\Omega$ the quadratic form $\mathbf{a}^\dagger Q \mathbf{a}$ in the previous exercise becomes

$$\mathbf{a}^\dagger Q_\Omega \mathbf{a} = \frac{1}{2\pi} \int_{-\Omega}^{\Omega} |\sum\nolimits_{n=1}^{M} a_n e^{in\omega}|^2 d\omega.$$

Show that the norm of the vector $\mathbf{a}$ is the integral

$$\frac{1}{2\pi} \int_{-\pi}^{\pi} |\sum\nolimits_{n=1}^{M} a_n e^{in\omega}|^2 d\omega.$$

**Solution:** Write

$$|\sum\nolimits_{n=1}^{M} a_n e^{in\omega}|^2 = \sum\nolimits_{n=1}^{M} \sum\nolimits_{m=1}^{M} a_n \overline{a_m} e^{i(n-m)\omega}.$$

**Exercise 3:** For $M = 30$ compute the eigenvalues of the matrix $Q_\Omega$ for various choices of $\Omega$, such as $\Omega = \frac{\pi}{k}$, for $k = 2, 3, ..., 10$. For each $k$ arrange the set of eigenvalues in decreasing order and note the proportion of them that are not near zero. The set of eigenvalues of a matrix is sometimes called its *eigenspectrum* and the nonnegative function $\chi_\Omega(\omega)$ is a power spectrum; here is one time in which different notions of a *spectrum* are related.

**Solution:** We find that the eigenvalues separate, more or less, into two groups: those near one and those near zero. The number of eigenvalues in the first group is roughly $30\Omega/\pi$.

**Exercise 5:** Show that the non-iterative Gerchberg-Papoulis bandlimited extrapolation method leads to the estimate of $F(\omega)$ given by

$$F_\Omega(\omega) = \chi_\Omega(\omega) \sum_{m=1}^{M} \frac{1}{\lambda_m} (\mathbf{u}^m)^\dagger \mathbf{d}\, U^m(\omega),$$

where $\mathbf{d}$ is the data vector.

**Solution:** Expand $Q^{-1}f$ using the eigenvector/eigenvalue expression for $Q^{-1}$.

**Exercise 6:** Show that the DFT estimate of $F(\omega)$, restricted to the interval $[-\Omega, \Omega]$, is

$$F_{DFT}(\omega) = \chi_\Omega(\omega) \sum_{m=1}^{M} (\mathbf{u}^m)^\dagger \mathbf{d}\, U^m(\omega).$$

**Solution:** Use the fact that the identity matrix can be written as $I = UU^\dagger$.

**The PDFT**

**Exercise 1:** Show that the $c_m$ must satisfy the equations

$$f(x_n) = \sum_{m=1}^{M} c_m p(x_n - x_m),\ n = 1, ..., M,$$

where $p(x)$ is the inverse Fourier transform of $P(\omega)$.

**Solution:** The inverse FT of the function $F_{PDFT}(\omega)$ is

$$f_{PDFT}(x) = \sum_{m=1}^{M} c_m p(x - x_m).$$

In order for $f_{PDFT}(x)$ to be data consistent we must have

$$f_{PDFT}(x_n) = \sum_{m=1}^{M} c_m p(x_n - x_m)$$

for $n = 1, ..., M$.

**Exercise 2:** Show that the estimate $F_{PDFT}(\omega)$ minimizes the distance

$$\int |F(\omega) - P(\omega) \sum_{m=1}^{M} a_m \exp(i x_m \omega)|^2 P(\omega)^{-1} d\omega$$

over all choices of the coefficients $a_m$.

**Solution:** According to the orthogonality principle the optimal choice $a_m = c_m$ must satisfy

$$0 = \int (F(\omega) - P(\omega) \sum_{m=1}^{M} c_m \exp(i x_m \omega)) P(\omega) e^{-i x_n \omega} P(\omega)^{-1} d\omega,$$

for $n = 1, ..., M$. Therefore

$$0 = \int (F(\omega) - P(\omega) \sum_{m=1}^{M} c_m \exp(ix_m\omega))e^{-ix_n\omega} d\omega,$$

which tells us that

$$f(x_n) = \sum_{m=1}^{M} c_m p(x_n - x_m)$$

for $n = 1, ..., M$.

**A Little Matrix Theory**

**Exercise 1:** Show that if $\mathbf{z} = (z_1, ..., z_N)^T$ is a column vector with complex entries and $H = H^\dagger$ is an $N$ by $N$ Hermitian matrix with complex entries then the quadratic form $\mathbf{z}^\dagger H\mathbf{z}$ is a real number. Show that the quadratic form $\mathbf{z}^\dagger H\mathbf{z}$ can be calculated using only real numbers. Let $\mathbf{z} = \mathbf{x} + i\mathbf{y}$, with $\mathbf{x}$ and $\mathbf{y}$ real vectors and let $H = A + iB$, where $A$ and $B$ are real matrices. Then show that $A^T = A$, $B^T = -B$, $\mathbf{x}^T B\mathbf{x} = 0$ and finally,

$$\mathbf{z}^\dagger H\mathbf{z} = [\,\mathbf{x}^T \quad \mathbf{y}^T\,] \begin{bmatrix} A & -B \\ B & A \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix}.$$

Use the fact that $\mathbf{z}^\dagger H\mathbf{z}$ is real for every vector $\mathbf{z}$ to conclude that the eigenvalues of $H$ are real.

**Solution:** The quadratic form $\mathbf{z}^\dagger H\mathbf{z}$ is a complex number and also the product of three matrices. Its conjugate transpose is simply its complex conjugate, since it is only 1 by 1; but

$$(\mathbf{z}^\dagger H\mathbf{z})^\dagger = \mathbf{z}^\dagger H^\dagger (\mathbf{z}^\dagger)^\dagger = \mathbf{z}^\dagger H\mathbf{z}$$

since $H$ is Hermitian. The complex conjugate of $\mathbf{z}^\dagger H\mathbf{z}$ is itself, so it must be real. We have

$$A + iB = H = H^\dagger = A^T - iB^T,$$

so that $A = A^T$ and $B^T = -B$.

Writing $\mathbf{z}^\dagger Q\mathbf{z}$ in terms of $A$, $B$, $\mathbf{x}$ and $\mathbf{y}$ we get

$$\mathbf{z}^\dagger Q\mathbf{z} = (\mathbf{x}^T - i\mathbf{y}^T)(A + iB)(\mathbf{x} + i\mathbf{y}) = (\mathbf{x}^T - i\mathbf{y}^T)(A\mathbf{x} - B\mathbf{y} + i(B\mathbf{x} + A\mathbf{y})$$

$$= \mathbf{x}^T A\mathbf{x} - \mathbf{x}^T B\mathbf{y} + \mathbf{y}^T B\mathbf{x} + \mathbf{y}^T A\mathbf{y} + i(\mathbf{x}^T B\mathbf{x} + \mathbf{x}^T A\mathbf{y} - \mathbf{y}^T A\mathbf{x} + \mathbf{y}^T B\mathbf{y})$$

$$= \mathbf{x}^T A\mathbf{x} + \mathbf{y}^T A\mathbf{y} - \mathbf{x}^T B\mathbf{y} + \mathbf{y}^T B\mathbf{x}$$

since

$$\mathbf{x}^T B\mathbf{x} = (\mathbf{x}^T B\mathbf{x})^T = \mathbf{x}^T B^T\mathbf{x} = -\mathbf{x}^T B\mathbf{x}$$

implies that $\mathbf{x}^T B\mathbf{x} = 0$ and, similarly, $\mathbf{y}^T B\mathbf{y} = 0$.

Let $\lambda$ be an eigenvalue of $H$ associated with eigenvector $\mathbf{u}$. Then

$$\mathbf{u}^\dagger H\mathbf{u} = \mathbf{u}^\dagger(\lambda\mathbf{u}) = \lambda\mathbf{u}^\dagger\mathbf{u} = \lambda.$$

Since $\mathbf{u}^\dagger H\mathbf{u}$ is real, so is $\lambda$.

**Exercise 2:** Let $A$ be an $M$ by $N$ matrix with complex entries. View $A$ as a linear function with domain $C^N$, the space of all $N$-dimensional complex column vectors, and range contained within $C^M$, via the expression $A(\mathbf{x}) = A\mathbf{x}$. Suppose that $M > N$. The range of $A$, denoted $R(A)$, cannot be all of $C^M$. Show that every vector $\mathbf{z}$ in $C^M$ can be written uniquely in the form $\mathbf{z} = A\mathbf{x} + \mathbf{w}$, where $A^\dagger\mathbf{w} = \mathbf{0}$. Show that $\|\mathbf{z}\|^2 = \|A\mathbf{x}\|^2 + \|\mathbf{w}\|^2$, where $\|\mathbf{z}\|^2$ denotes the square of the norm of $\mathbf{z}$. Hint: If $\mathbf{z} = A\mathbf{x} + \mathbf{w}$ then consider $A^\dagger\mathbf{z}$. Assume $A^\dagger A$ is invertible.

**Solution:** We assume that $A^\dagger A$ is invertible. If $\mathbf{z} = A\mathbf{x} + \mathbf{v}$ with $A^\dagger\mathbf{v} = 0$ then $A^\dagger\mathbf{z} = A^\dagger A\mathbf{x}$, so that $\mathbf{x} = (A^\dagger A)^{-1}A^\dagger\mathbf{z}$. Then

$$\mathbf{v} = \mathbf{z} - A(A^\dagger A)^{-1}A^\dagger\mathbf{z}$$

and we see easily that $A^\dagger\mathbf{v} = 0$. Then we have

$$||\mathbf{z}||^2 = ||A\mathbf{x} + \mathbf{v}||^2 = \mathbf{x}^\dagger A^\dagger A\mathbf{x} + \mathbf{x}^\dagger A^\dagger\mathbf{v} + \mathbf{v}^\dagger A\mathbf{x} + \mathbf{v}^\dagger\mathbf{v} = ||A\mathbf{x}||^2 + ||\mathbf{v}||^2.$$

**Exercise 5:** Show that the vector $\mathbf{x} = (x_1, ..., x_N)^T$ minimizes the mean squared error

$$\|A\mathbf{x} - \mathbf{b}\|^2 = \sum_{m=1}^{N}(A\mathbf{x}_m - b_m)^2,$$

if and only if $\mathbf{x}$ satisfies the system of linear equations $A^T(A\mathbf{x} - \mathbf{b}) = \mathbf{0}$, where $A\mathbf{x}_m = (A\mathbf{x})_m = \sum_{n=1}^{N} A_{mn}x_n$. Hint: calculate the partial derivative of $\|A\mathbf{x} - \mathbf{b}\|^2$ with respect to each $x_n$.

**Solution:** The partial derivative of $\|A\mathbf{x} - \mathbf{b}\|^2$ with respect to $x_n$ is

$$2\sum_{m=1}^{M} A_{mn}(A\mathbf{x}_m - b_m).$$

Setting each of these partial derivatives equal to zero gives

$$A^T(A\mathbf{x} - \mathbf{b}) = 0.$$

**Exercise 8:** Show that $F_\epsilon$ always has a unique minimizer $\hat{\mathbf{x}}_\epsilon$ given by

$$\hat{\mathbf{x}}_\epsilon = ((1-\epsilon)A^TA + \epsilon I)^{-1}((1-\epsilon)A^T\mathbf{b} + \epsilon\mathbf{p});$$

this is a regularized solution of $A\mathbf{x} = \mathbf{b}$. Here $\mathbf{p}$ is a prior estimate of the desired solution. Note that the inverse above always exists.

**Solution:** Set to zero the partial derivatives with respect to each of the variables $x_n$. Show that the second derivative matrix is $A^TA+\epsilon I$, which is positive-definite; therefore the partial derivatives are zero at a minimum.

**Exercise 9:** Show that, in **Case 1**, taking limits as $\epsilon \to 0$ on both sides of the expression for $\hat{\mathbf{x}}_\epsilon$ gives $\hat{\mathbf{x}}_\epsilon \to (A^TA)^{-1}A^T\mathbf{b}$, the least squares solution of $A\mathbf{x} = \mathbf{b}$.

**Solution:** In this case we can simply set $\epsilon = 0$, since the inverse $(A^TA)^{-1}$ exists.

**Exercise 10:** Show that

$$((1-\epsilon)A^TA + \epsilon I)^{-1}(\epsilon\mathbf{r}) = \mathbf{r}, \forall\epsilon.$$

**Solution:** As in the hint, let

$$\mathbf{t}_\epsilon = ((1-\epsilon)A^TA + \epsilon I)^{-1}(\epsilon\mathbf{r}).$$

Then multiplying by $A$ gives

$$A\mathbf{t}_\epsilon = A((1-\epsilon)A^TA + \epsilon I)^{-1}(\epsilon\mathbf{r}).$$

Now it follows from $A\mathbf{r} = \mathbf{0}$ and

$$((1-\epsilon)AA^T + \epsilon I)^{-1}A = A((1-\epsilon)A^TA + \epsilon I)^{-1}$$

that $A\mathbf{t}_\epsilon = \mathbf{0}$. Now multiply both sides of the equation

$$\mathbf{t}_\epsilon = ((1-\epsilon)A^TA + \epsilon I)^{-1}(\epsilon\mathbf{r})$$

by $(1-\epsilon)A^TA + \epsilon I$ to get $\epsilon\mathbf{t}_\epsilon = \epsilon\mathbf{r}$. Now we take the limit of $\hat{\mathbf{x}}_\epsilon$, as $\epsilon \to 0$, by setting $\epsilon = 0$, to get $\hat{\mathbf{x}}_\epsilon \to A^T(AA^T)^{-1}\mathbf{b} + \mathbf{r} = \hat{\mathbf{x}}$.

Now we show that $\hat{\mathbf{x}}$ is the solution of $A\mathbf{x} = \mathbf{b}$ closest to $\mathbf{p}$. By the orthogonality theorem it must then be the case that $\langle \mathbf{p} - \hat{\mathbf{x}}, \mathbf{x} - \hat{\mathbf{x}}\rangle = \mathbf{0}$ for every $\mathbf{x}$ with $A\mathbf{x} = \mathbf{b}$. Since $\mathbf{p} - \hat{\mathbf{x}} = A^T\mathbf{q} - A^T(AA^T)^{-1}\mathbf{b}$ we have

$$\langle \mathbf{p} - \hat{\mathbf{x}}, \mathbf{x} - \hat{\mathbf{x}}\rangle = \langle \mathbf{q} - (AA^T)^{-1}\mathbf{b}, A\mathbf{x} - A\hat{\mathbf{x}}\rangle = 0.$$

**Matrix and Vector Calculus**

**Exercise 1:** Let $\mathbf{y}$ be a fixed real column vector and $z = f(\mathbf{x}) = \mathbf{y}^T\mathbf{x}$. Show that

$$\frac{\partial z}{\partial \mathbf{x}} = \mathbf{y}.$$

**Solution:** We write

$$z = \mathbf{y}^T\mathbf{x} = \sum_{n=1}^{N} x_n y_n$$

so that

$$\frac{\partial z}{\partial x_n} = y_n$$

for each $n$.

**Exercise 2:** Let $Q$ be a real symmetric nonnegative definite matrix and let $z = f(\mathbf{x}) = \mathbf{x}^T Q\mathbf{x}$. Show that the gradient of this quadratic form is

$$\frac{\partial z}{\partial \mathbf{x}} = 2Q\mathbf{x}.$$

**Solution:** Following the hint, we write $Q$ as a linear combination of dyads involving the eigenvectors; that is

$$Q = \sum_{m=1}^{N} \lambda_m \mathbf{u}^m (\mathbf{u}^m)^\dagger.$$

Then

$$z = \mathbf{x}^T Q\mathbf{x} = \sum_{m=1}^{N} \lambda_m (\mathbf{x}^T\mathbf{u}^m)^2$$

so that

$$z = \sum_{m=1}^{N} \lambda_m (\sum_{n=1}^{N} x_n u_n^m)^2.$$

Therefore, the partial derivative of $z$ with respect to $x_n$ is

$$\frac{\partial z}{\partial x_n} = 2\sum_{m=1}^{N} \lambda_n (x_n u_m^n) u_m^n,$$

which can then be written as

$$\frac{\partial z}{\partial \mathbf{x}} = 2Q\mathbf{x}.$$

**Exercise 3:** Let $z = ||A\mathbf{x} - \mathbf{b}||^2$. Show that

$$\frac{\partial z}{\partial \mathbf{x}} = 2A^T A\mathbf{x} - 2A^T\mathbf{b}.$$

**Solution:** Using $z = (A\mathbf{x} - \mathbf{b})^T(A\mathbf{x} - \mathbf{b})$ we get

$$z = \mathbf{x}^T A^T A\mathbf{x} - \mathbf{b}^T A\mathbf{x} - \mathbf{x}^T A^T \mathbf{b} + \mathbf{b}^T \mathbf{b}.$$

Then it follows from the two previous exercises that

$$\frac{\partial z}{\partial \mathbf{x}} = 2A^T A\mathbf{x} - 2A^T \mathbf{b}.$$

**Exercise 4:** Suppose $(u, v) = (u(x, y), v(x, y))$ is a change of variables from the Cartesian $(x, y)$ coordinate system to some other $(u, v)$ coordinate system. Let $\mathbf{x} = (x, y)^T$ and $\mathbf{z} = (u(\mathbf{x}), v(\mathbf{x}))^T$.

**a:** Calculate the Jacobian for the rectangular coordinate system obtained by rotating the $(x, y)$ system through an angle of $\theta$.

**Solution:** The equations for this change of coordinates are

$$u = x\cos\theta + y\sin\theta,$$

and

$$v = -x\sin\theta + y\cos\theta.$$

Then $u_x = \cos\theta$, $u_y = \sin\theta$, $v_x = -\sin\theta$ and $v_y = \cos\theta$. The Jacobian is therefore one.

**b:** Calculate the Jacobian for the transformation from the $(x, y)$ system to polar coordinates.

**Solution:** We have $r = \sqrt{x^2 + y^2}$ and $\tan\theta = \frac{y}{x}$. Writing $r^2 = x^2 + y^2$, we get $2rr_x = 2x$ and $2rr_y = 2y$, so that $r_x = x/r$ and $r_y = y/r$. Also

$$(\sec\theta)^2\theta_x = -y/x^2$$

and

$$(\sec\theta)^2\theta_y = 1/x.$$

Since $\sec\theta = r/x$ we get

$$\theta_x = \frac{x^2}{r^2}\frac{-y}{x^2} = \frac{-y}{r^2}$$

and

$$\theta_y = \frac{x^2}{r^2}\frac{1}{x} = \frac{x}{r^2}.$$

The Jacobian is therefore $\frac{1}{r}$.

**Exercise 6:** Show that the derivative of $z = \text{trace}\,(DAC)$ with respect to $A$ is

$$\frac{\partial z}{\partial A} = D^T C^T.$$

**Solution:** Just write out the general term of $DAC$.

**Exercise 7:** Let $z = \text{trace}\,(A^T CA)$. Show that the derivative of $z$ with respect to the matrix $A$ is

$$\frac{\partial z}{\partial A} = CA + C^T A.$$

Therefore, if $C = Q$ is symmetric, then the derivative is $2QA$.

**Solution:** Again, just write out the general term of $A^T CA$.

**The Singular Value Decomposition**

**Exercise 1:** Show that the nonzero eigenvalues of $A$ and $B$ are the same.

**Solution:** Let $\lambda$ be a nonzero eigenvalue of $A$, with $A\mathbf{u} = \lambda\mathbf{u}$ for some nonzero vector $\mathbf{u}$. Then $CA\mathbf{u} = \lambda C\mathbf{u}$ or $(CC^\dagger)C\mathbf{u} = BC\mathbf{u} = \lambda C\mathbf{u}$; with $C\mathbf{u} = \mathbf{v}$ we have $B\mathbf{v} = \lambda\mathbf{v}$. Since $B$ is invertible $\mathbf{v}$ is not the zero vector. So $\lambda$ is an eigenvalue of $B$.

Conversely, let $\lambda \neq 0$ be an eigenvalue of $B$, with $B\mathbf{v} = \lambda\mathbf{v}$ for some nonzero $\mathbf{v}$. Then $B\mathbf{v} = CC^\dagger\mathbf{v} = \lambda\mathbf{v}$ and so $C^\dagger B\mathbf{v} = (C^\dagger C)C^\dagger\mathbf{v} = AC^\dagger\mathbf{v} = \lambda C^\dagger\mathbf{v}$. We need to show that $\mathbf{w} = C^\dagger\mathbf{v}$ is not the zero vector. If $\mathbf{0} = \mathbf{w} = C^\dagger\mathbf{v}$ then $\mathbf{0} = C\mathbf{w} = CC^\dagger\mathbf{v} = B\mathbf{v}$. But $B$ is invertible and $\mathbf{v}$ is nonzero; this is a contradiction, so we conclude that $\mathbf{w} \neq \mathbf{0}$.

**Exercise 2:** Show that $UMV^\dagger$ equals $C$.

**Solution:** The first $N$ columns of the matrix $UM$ form the matrix

$$ULL^{-1/2} = BUL^{-1/2}$$

and the remaining columns are zero. Consider the product $V(UM)^\dagger$. The first $N$ columns of $V$ form the matrix $C^\dagger UL^{-1/2}$ so

$$V(UM)^\dagger = C^\dagger UL^{-1}U^\dagger B = C^\dagger B^{-1}B = C^\dagger$$

and so $UMV^\dagger = C$.

**Exercise 3:** If $N > K$ the system $C\mathbf{x} = \mathbf{d}$ probably has no exact solution. Show that $C^* = (C^\dagger C)^{-1}C^\dagger$ so that the vector $\mathbf{x} = C^*\mathbf{d}$ is the least squares approximate solution.

**Solution:** Show that $(C^\dagger C)C^* = C^\dagger = VM^TU^\dagger$.

**Exercise 4:** If $N < K$ the system $C\mathbf{x} = \mathbf{d}$ probably has infinitely many solutions. Show that the pseudo-inverse is now $C^* = C^\dagger(CC^\dagger)^{-1}$, so that the vector $\mathbf{x} = C^*\mathbf{d}$ is the exact solution of $C\mathbf{x} = \mathbf{d}$ closest to the origin; that is, it is the minimum norm solution.

**Solution:** Show that $C^*(CC^\dagger) = C^\dagger$.

**Discrete Random Processes**

**Exercise 1:** Show that the autocorrelation matrix $R$ is nonnegative definite. Under what conditions can $R$ fail to be positive-definite?

**Solution:** Let

$$A(\omega) = \sum\nolimits_{n=1}^{N+1} a_n e^{in\omega}.$$

Then we have

$$\int |A(\omega)|^2 R(\omega) d\omega = \mathbf{a}^\dagger R\mathbf{a} \geq 0.$$

If the quadratic form $\mathbf{a}^\dagger R\mathbf{a} = 0$ for some vector $\mathbf{a}$ then the integral must also be zero, which says that the power spectrum is nonzero only when the polynomial is zero; that is, the power spectrum $R(\omega)$ is a sum of not more than $N$ delta functions.

**Best Linear Unbiased Estimation**

**Exercise 1:** Show that

$$E(|\hat{\mathbf{x}} - \mathbf{x}|^2) = \text{trace} K^\dagger Q K.$$

**Solution:** Write the left side as

$$E(\text{trace}\,((\hat{\mathbf{x}} - \mathbf{x})(\hat{\mathbf{x}} - \mathbf{x})^\dagger)).$$

Also use the fact that the trace and expected value operations commute. Then

$$E(|\hat{\mathbf{x}} - \mathbf{x}|^2) = \text{trace}(E(K^\dagger \mathbf{z}\mathbf{z}^\dagger K - \mathbf{x}\mathbf{z}^\dagger K - K^\dagger \mathbf{z}\mathbf{x}^\dagger + \mathbf{x}\mathbf{x}^\dagger)) = E(K^\dagger \mathbf{z}\mathbf{z}^\dagger K) - \mathbf{x}\mathbf{x}^\dagger.$$

Notice that

$$\mathbf{z}\mathbf{z}^\dagger = H\mathbf{x}\mathbf{x}^\dagger H^\dagger + H\mathbf{x}\mathbf{v}^\dagger + \mathbf{v}\mathbf{x}^\dagger H^\dagger + \mathbf{v}\mathbf{v}^\dagger.$$

Therefore

$$E(K^\dagger \mathbf{z}\mathbf{z}^\dagger K) = K^\dagger H\mathbf{x}\mathbf{x}^\dagger H^\dagger K + K^\dagger Q K.$$

It follows that

$$E(|\hat{\mathbf{x}} - \mathbf{x}|^2) = \text{trace} K^\dagger Q K.$$

**The Vector Wiener Filter**

**Exercise 1:** Apply the vector Wiener filter to the simplest problem discussed earlier. Here let $K = 1$ and $NN^\dagger = Q$.

**Solution:** Let $\mathbf{1} = (1, 1, ..., 1)^T$, so that the signal vector is $\mathbf{s} = c\mathbf{1}$ for some constant $c$ and the data vector is $\mathbf{z} = c\mathbf{1} + \mathbf{v}$. Then $SS^\dagger = \mathbf{1}\mathbf{1}^T$. We have

$$(Q + \mathbf{1}\mathbf{1}^\dagger)^{-1} = Q^{-1} - (1 + \mathbf{1}^\dagger Q^{-1}\mathbf{1})^{-1} Q^{-1}\mathbf{1}\mathbf{1}^\dagger Q^{-1},$$

so we get

$$\hat{\mathbf{s}} = \frac{\mathbf{1}^\dagger Q^{-1}\mathbf{z}}{1 + \mathbf{1}^\dagger Q^{-1}\mathbf{1}}\mathbf{1},$$

and the estimate of the constant $c$ is

$$\hat{c} = \frac{\mathbf{1}^\dagger Q^{-1}\mathbf{z}}{1 + \mathbf{1}^\dagger Q^{-1}\mathbf{1}}.$$

When the noise power is very low the denominator is dominated by the second term and we get the BLUE estimate.

**Eigenvector Methods**

**Exercise 2:** Show that $\lambda_m = \sigma^2$ for $m = J + 1, ..., M$, while $\lambda_m > \sigma^2$ for $m = 1, ..., J$.

**Solution:** From Exercise 1 we conclude that, for any vector $\mathbf{u}$ the quadratic form $\mathbf{u}^\dagger R\mathbf{u}$ is

$$\mathbf{u}^\dagger R\mathbf{u} = \sum_{j=1}^{J} |A_j|^2 |\mathbf{u}^\dagger \mathbf{e}_j|^2 + \sigma^2 |\mathbf{u}^\dagger \mathbf{u}|^2.$$

The norm-one eigenvectors of $R$ associated with the $J$ largest eigenvalues will lie in the linear span of the vectors $\mathbf{e}_j$, $j = 1, ..., J$, while the remaining $M - J$ eigenvectors will be orthogonal to the $\mathbf{e}_j$. For these remaining eigenvectors the quadratic form will have the value $\lambda_m = \sigma^2$, since the eigenvectors have norm equal to one. For the eigenvectors associated with the $J$ largest eigenvalues, the quadratic form will be greater than $\sigma^2$, since it will also involve a positive term coming from the sum.

Since $M > J$ the $M - J$ orthogonal eigenvectors $\mathbf{u}_m$ corresponding to $\lambda_m$ for $m = J+1, ..., M$ will be orthogonal to each of the $\mathbf{e}_j$. Then consider the quadratic forms $\mathbf{u}_m^\dagger R\mathbf{u}_m$.

**Signal Detection and Estimation**

**Exercise 1:** Use Cauchy's inequality to show that, for any fixed vector $\mathbf{a}$, the choice $\mathbf{b} = \beta\mathbf{a}$ maximizes the quantity $|\mathbf{b}^\dagger\mathbf{a}|^2/\mathbf{b}^\dagger\mathbf{b}$, for any constant $\beta$.

**Solution:** According to Cauchy's inequality the quantity $\frac{|\mathbf{b}^\dagger\mathbf{a}|^2}{\mathbf{b}^\dagger\mathbf{b}}$ does not exceed $\mathbf{a}^\dagger\mathbf{a}$. The choice of $\mathbf{b} = \beta\mathbf{a}$ makes the ratio equal to $\mathbf{a}^\dagger\mathbf{a}$, so maximizes the ratio.

**Exercise 2:** Use the definition of the correlation matrix $Q$ to show that $Q$ is Hermitian and that, for any vector $\mathbf{y}$, $\mathbf{y}^\dagger Q\mathbf{y} \geq 0$. Therefore $Q$ is a nonnegative definite matrix and, using its eigenvector decomposition, can be written as $Q = CC^\dagger$, for some invertible square matrix $C$.

**Solution:** The entry of $Q$ in the $m$-th row and $n$-th column is $Q_{mn} = E(z_m\overline{z_n})$, so $Q_{nm} = \overline{Q_{mn}}$. For any vector $\mathbf{y}$ the quadratic form $\mathbf{y}^\dagger Q\mathbf{y} = E(|\mathbf{y}^\dagger\mathbf{z}|^2)$ and the expected value of a nonnegative random variable is nonnegative. Therefore $Q$ is Hermitian and nonnegative-definite, so its eigenvalues are nonnegative. The eigenvector/eigenvalue decomposition is $Q = ULU^\dagger$, where $L$ is the diagonal matrix with the eigenvalues on the main diagonal. Since these eigenvalues are nonnegative, they have nonnegative square roots. Make these the diagonal elements of the matrix $L^{1/2}$ and write $C = UL^{1/2}U^\dagger$. Then we have $C = C^\dagger$ and $CC^\dagger = C^\dagger C = Q$.

**Exercise 3:** Consider now the problem of maximizing $|\mathbf{b}^\dagger\mathbf{s}|^2/\mathbf{b}^\dagger Q\mathbf{b}$. Using the two previous exercises, show that the solution is $\mathbf{b} = \beta Q^{-1}\mathbf{s}$, for some arbitrary constant $\beta$.

**Solution:** Write $\mathbf{b}^\dagger Q\mathbf{b} = \mathbf{b}^\dagger C^\dagger C\mathbf{b} = \mathbf{d}^\dagger\mathbf{d}$, for $\mathbf{d} = C\mathbf{b}$. We assume that $Q$ is invertible, so $C$ is also. Write

$$\mathbf{b}^\dagger\mathbf{s} = \mathbf{b}^\dagger C^\dagger(C^\dagger)^{-1}\mathbf{s} = \mathbf{d}^\dagger\mathbf{e},$$

for $\mathbf{e} = (C^\dagger)^{-1}\mathbf{s}$. So the problem now is to maximize the ratio $\frac{|\mathbf{d}^\dagger\mathbf{e}|^2}{\mathbf{d}^\dagger\mathbf{d}}$. By the first exercise we know that this ratio is maximized when we select $\mathbf{d} = \beta\mathbf{e}$ for some constant $\beta$. This means that $C\mathbf{b} = \beta(C^\dagger)^{-1}\mathbf{s}$ or $\mathbf{b} = \beta Q^{-1}\mathbf{s}$. Here the $\beta$ is a free choice; we select it so that $\mathbf{b}^\dagger\mathbf{s} = 1$.

**A Little Probability Theory**

**Exercise 1:** Show that the sequence $\{p_k\}_{k=0}^\infty$ sums to one.

**Solution:** The Taylor series expansion of the function $e^x$ is

$$e^x = \sum_{k=0}^\infty \frac{x^k}{k!},$$

so

$$\sum_{k=0}^{\infty} p_k = e^{-\lambda} \sum_{k=0}^{\infty} \frac{\lambda^k}{k!} = 1.$$

**Exercise 2:** Show that the expected value $E(X)$ is $\lambda$, where the expected value in this case is

$$E(X) = \sum_{k=0}^{\infty} k p_k.$$

**Solution:** Note that

$$\sum_{k=0}^{\infty} k p_k = e^{-\lambda} \sum_{k=1}^{\infty} \frac{\lambda^k}{(k-1)!}$$

$$= e^{-\lambda} \lambda \sum_{k=0}^{\infty} \frac{\lambda^k}{k!} = \lambda.$$

**Exercise 3:** Show that the variance of $X$ is also $\lambda$, where the variance of $X$ in this case is

$$\mathrm{var}(X) = \sum_{k=0}^{\infty} (k-\lambda)^2 p_k.$$

**Solution:** Use

$$(k-\lambda)^2 = k^2 - 2k\lambda + \lambda^2 = k(k-1) + k - 2k\lambda + \lambda^2.$$

**Exercise 4:** Prove these two assertions.

**Solution:** The expected value of $\overline{X}$ is

$$E(\overline{X}) = \frac{1}{N} \sum_{n=1}^{N} E(X_n) = \frac{1}{N} \sum_{n=1}^{N} \mu = \mu.$$

The variance of $\overline{X}$ is

$$E((\overline{X} - \mu)^2) = E(\overline{X}^2 - 2\mu\overline{X} + \mu^2)$$

$$= E(\overline{X}^2) - \mu^2.$$

Then

$$E(\overline{X}^2) = \frac{1}{N^2} E(\sum_{n=1}^{N} X_n \sum_{m=1}^{N} X_m).$$

Now use the fact that $E(X_n X_m) = E(X_n)E(X_m) = \mu^2$ if $m \neq n$ while $E(X_n X_n) = \sigma^2 + \mu^2$.

**More on the ART**

**Exercise 1:** Establish the following facts concerning the ART.

Fact 1:

$$||\mathbf{x}^k||^2 - ||\mathbf{x}^{k+1}||^2 = (A(\mathbf{x}^k)_{m(k)})^2 - (b_{m(k)})^2.$$

**Solution:** Write $||\mathbf{x}^{k+1}||^2 = ||\mathbf{x}^k + (\mathbf{x}^{k+1} - \mathbf{x}^k)||^2$ and expand using the complex dot product.

Fact 2:

$$||\mathbf{x}^{rM}||^2 - ||\mathbf{x}^{(r+1)M}||^2 = ||\mathbf{v}^r||^2 - ||\mathbf{b}||^2.$$

**Solution:** The solution is similar to that of the previous exercise.

Fact 3:

$$||\mathbf{x}^k - \mathbf{x}^{k+1}||^2 = ((A\mathbf{x}^k)_{m(k)} - b_m)^2.$$

**Solution:** Easy.

Fact 4: There exists $B > 0$ such that, for all $r = 0, 1, ...$, if $||\mathbf{v}^r|| \leq ||\mathbf{b}||$ then $||\mathbf{x}^{rM}|| \geq ||\mathbf{x}^{(r+1)M}|| - B$.

**Solution:** This is an application of the triangle inequality.

Fact 5: Let $\mathbf{x}^0$ and $\mathbf{y}^0$ be arbitrary and $\{\mathbf{x}^k\}$ and $\{\mathbf{y}^k\}$ the sequences generated by applying the ART algorithm. Then

$$||\mathbf{x}^0 - \mathbf{y}^0||^2 - ||\mathbf{x}^M - \mathbf{y}^M||^2 = \sum_{m=1}^{M} ((A\mathbf{x}^{m-1})_m - (A\mathbf{y}^{m-1})_m)^2.$$

**Solution:** Calculate $||\mathbf{x}^m - \mathbf{y}^m||^2 - ||\mathbf{x}^{m+1} - \mathbf{y}^{m+1}||^2$ for each $m = 0, 1, ..., M-1$ and then add.

**Exercise 3:** Show that if we select $B$ so that $C$ is invertible and $B^T A = 0$ then the exact solution of $C\mathbf{z} = \mathbf{b}$ is the concatenation of the least squares solutions of $A\mathbf{x} = \mathbf{b}$ and $B\mathbf{y} = \mathbf{b}$.

**Solution:** Calculate the solution of $C\mathbf{z} = \mathbf{b}$ as the least squares solution of $C\mathbf{z} = \mathbf{b}$.

**The MART and related methods**

**Exercise 1:** Show that

$$KL(\mathbf{x},\mathbf{z}) = KL(x_+,z_+) + KL(\mathbf{x},\frac{x_+}{z_+}\mathbf{z})$$

for any nonnegative vectors $\mathbf{x}$ and $\mathbf{z}$, with $x_+$ and $z_+ > 0$ denoting the sums of the entries of vectors $\mathbf{x}$ and $\mathbf{z}$, respectively.

**Solution:** Begin with $KL(\mathbf{x},\frac{x_+}{z_+}\mathbf{z})$ and write it out as

$$KL(\mathbf{x},\frac{x_+}{z_+}\mathbf{z}) = \sum_{n=1}^{N} x_n \log(x_n/\frac{x_+}{z_+}z_n) + \frac{x_+}{z_+}\sum_{n=1}^{N} z_n - \sum_{n=1}^{N} x_n$$

$$= \sum_{n=1}^{N}(x_n \log\frac{x_n}{z_n} + z_n - x_n) - \sum_{n=1}^{N}(x_n \log\frac{x_+}{z_+} + (\frac{x_+}{z_+} - 1)z_n)$$

$$= KL(\mathbf{x},\mathbf{z}) - x_+ \log\frac{x_+}{z_+} + x_+ - z_+ = KL(\mathbf{x},\mathbf{z}) - KL(x_+,z_+).$$

**The Wave Equation**

**Exercise 1:** Show that the radial function $u(r,t) = \frac{1}{r}h(r-ct)$ satisfies the wave equation for any twice differentiable function $h$.

**Solution:** The partial derivatives are as follows:

$$u_t = -c\frac{1}{r}h'(r-ct),$$

$$u_{tt} = c^2\frac{1}{r}h''(r-ct),$$

$$u_r = -\frac{1}{r^2}h(r-ct) + \frac{1}{r}h'(r-ct),$$

and

$$u_{rr} = 2\frac{1}{r^3}h(r-ct) - \frac{2}{r^2}h'(r-ct) + \frac{1}{r}h''(r-ct).$$

The result follows immediately from these facts.

**Exercise 2:** Let $\mathbf{s} = (x,y,z)$ and $u(\mathbf{s},t) = u(x,y,z,t) = e^{i\omega t}e^{i\mathbf{k}\cdot\mathbf{s}}$. Show that $u$ satisfies the wave equation $u_{tt} = c^2\nabla^2 u$ for any real vector $\mathbf{k}$, so long as $||\mathbf{k}||^2 = \omega^2/c^2$.

**Solution:** Easy.

# Bibliography


[1] Agmon, S. (1954) The relaxation method for linear inequalities, *Canadian Journal of Mathematics*, **6**, pp. 382–392.

[2] Anderson, T. (1972) Efficient estimation of regression coefficients in time series, *Proc. of Sixth Berkeley Symposium on Mathematical Statistics and Probability*, **1**, pp. 471–482.

[3] Anderson, A. and Kak, A. (1984) Simultaneous algebraic reconstruction technique (SART): a superior implementation of the ART algorithm, *Ultrasonic Imaging*, **6**, pp. 81–94.

[4] Ash, R., and Gardner, M. (1975) *Topics in Stochastic Processes*, Academic Press.

[5] Baggeroer, A., Kuperman, W., and Schmidt, H. (1988) Matched field processing: source localization in correlated noise as optimum parameter estimation, *Journal of the Acoustical Society of America*, **83**, pp. 571–587.

[6] Baillon, J., and Haddad, G. (1977) Quelques proprietes des operateurs angle-bornes et n-cycliquement monotones, *Israel J. of Mathematics*, **26**, pp. 137-150.

[7] H. Barrett, T. White and L. Parra (1997) List-mode likelihood, *J. Opt. Soc. Am. A*, **14**, pp. 2914–2923.

[8] Bauschke, H. (2001) Projection algorithms: results and open problems, in *Inherently Parallel Algorithms in Feasibility and Optimization and their Applications*, Butnariu, D., Censor, Y. and Reich, S., editors, Elsevier Publ., pp. 11–22.

[9] Bauschke, H., and Borwein, J. (1996) On projection algorithms for solving convex feasibility problems, *SIAM Review*, **38 (3)**, pp. 367–426.

[10] Bauschke, H., Borwein, J., and Lewis, A. (1997) The method of cyclic projections for closed convex sets in Hilbert space, *Contemporary Mathematics: Recent Developments in Optimization Theory and Nonlinear Analysis*, **204**, American Mathematical Society, pp. 1–38.

[11] Bertero, M. (1992) Sampling theory, resolution limits and inversion methods, in [13], pp. 71–94.

[12] Bertero, M., and Boccacci, P. (1998) *Introduction to Inverse Problems in Imaging*, Institute of Physics Publishing, Bristol, UK.

[13] Bertero, M., and Pike, E.R. (eds.) (1992) *Inverse Problems in Scattering and Imaging*, Malvern Physics Series, Adam Hilger, IOP Publishing, London.

[14] Bertsekas, D.P. (1997) A new class of incremental gradient methods for least squares problems, *SIAM J. Optim.*, **7**, pp. 913-926.

[15] Blackman, R., and Tukey, J. (1959) *The Measurement of Power Spectra*, Dover.

[16] Boggess, A., and Narcowich, F. (2001) *A First Course in Wavelets, with Fourier Analysis*, Prentice-Hall, NJ.

[17] Born, M., and Wolf, E. (1999) *Principles of Optics: 7-th edition*, Cambridge University Press.

[18] Bochner, S., and Chandrasekharan, K. (1949) *Fourier Transforms*, Annals of Mathematical Studies, No. 19, Princeton University Press.

[19] Borwein, J., and Lewis, A. (2000) *Convex Analysis and Nonlinear Optimization*, Canadian Mathematical Society Books in Mathematics, Springer, New York.

[20] Bregman, L.M. (1967) The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming, *USSR Computational Mathematics and Mathematical Physics*, **7**: 200–217.

[21] Brodzik, A., and Mooney, J. (1999) Convex projections algorithm for restoration of limited-angle chromotomographic images, *Journal of the Optical Society of America, A*, **16 (2)**, pp. 246–257.

[22] Browne, J. and A. DePierro, A. (1996) A row-action alternative to the EM algorithm for maximizing likelihoods in emission tomography, *IEEE Trans. Med. Imag.*, **15**, 687-699.

[23] Bruyant, P., Sau, J., and Mallet, J-J. (1999) Noise removal using factor analysis of dynamic structures: application to cardiac gated studies, *Journal of Nuclear Medicine*, **40 (10)**, 1676–1682.

[24] Bucker, H. (1976) Use of calculated sound fields and matched field detection to locate sound sources in shallow water, *Journal of the Acoustical Society of America*, **59**, pp. 368–373.

[25] Burg, J. (1967) Maximum entropy spectral analysis, *paper presented at the 37th Annual SEG meeting, Oklahoma City, OK*.

[26] Burg, J. (1972) The relationship between maximum entropy spectra and maximum likelihood spectra, *Geophysics*, **37**, pp. 375–376.

[27] Burg, J. (1975) *Maximum Entropy Spectral Analysis*, Ph.D. dissertation, Stanford University.

[28] Byrne, C. (1992) Effects of modal phase errors on eigenvector and nonlinear methods for source localization in matched field processing, *Journal of the Acoustical Society of America*, **92(4)**, pp. 2159–2164.

[29] Byrne, C. (1993) Iterative image reconstruction algorithms based on cross-entropy minimization, *IEEE Transactions on Image Processing*, **IP-2**, pp. 96–103.

[30] Byrne, C. (1995) Erratum and addendum to "Iterative image reconstruction algorithms based on cross-entropy minimization", *IEEE Transactions on Image Processing*, **IP-4**, pp. 225–226.

[31] Byrne, C. (1996) Iterative reconstruction algorithms based on cross-entropy minimization, in: *Image Models (and their Speech Model Cousins)*, (S.E. Levinson and L. Shepp, Editors), the IMA Volumes in Mathematics and its Applications, Volume 80, Springer-Verlag, New York, pp. 1–11.

[32] Byrne, C. (1996) Block-iterative methods for image reconstruction from projections, *IEEE Transactions on Image Processing*, **IP-5**, pp. 792–794.

[33] Byrne, C. (1997) Convergent block-iterative algorithms for image reconstruction from inconsistent data, *IEEE Transactions on Image Processing*, **IP-6**, pp. 1296–1304.

[34] Byrne, C. (1998) Accelerating the EMML algorithm and related iterative algorithms by rescaled block-iterative (RBI) methods, *IEEE Transactions on Image Processing*, **IP-7**, pp. 100-109.

[35] Byrne, C. (1999) Iterative projection onto convex sets using multiple Bregman distances, *Inverse Problems*, **15**, pp. 1295-1313.

[36] Byrne, C. (2000) Block-iterative interior point optimization methods for image reconstruction from limited data, *Inverse Problems*, **16**, pp. 1405–1419.

[37] Byrne, C. (2001) Bregman-Legendre multidistance projection algorithms for convex feasibility and optimization, in *Inherently Parallel Algorithms in Feasibility and Optimization and their Applications*, Butnariu, D., Censor, Y. and Reich, S., editors, Elsevier Publ., pp. 87–100.

[38] Byrne, C. (2001) Likelihood maximization for list-mode emission tomographic image reconstruction, *IEEE Transactions on Medical Imaging*, **20(10)**, pp. 1084–1092.

[39] Byrne, C. (2002) Iterative oblique projection onto convex sets and the split feasibility problem, *Inverse Problems*, **18**, pp. 441-453.

[40] Byrne, C. (2004) A unified treatment of some iterative algorithms in signal processing and image reconstruction, *Inverse Problems*, **20**, pp. 103–120.

[41] Byrne, C., Brent, R., Feuillade, C., and DelBalzo, D (1990) A stable data-adaptive method for matched-field array processing in acoustic waveguides, *Journal of the Acoustical Society of America*, **87(6)**, pp. 2493–2502.

[42] Byrne, C. and Censor, Y. (2001) Proximity function minimization using multiple Bregman projections, with applications to split feasibility and Kullback-Leibler distance minimization, *Annals of Operations Research*, **105**, pp. 77–98.

[43] Byrne, C. and Fiddy, M. (1987) Estimation of continuous object distributions from Fourier magnitude measurements, *JOSA A*, **4**, pp. 412–417.

[44] Byrne, C., and Fiddy, M. (1988) Images as power spectra; reconstruction as Wiener filter approximation, *Inverse Problems*, **4**, pp. 399–409.

[45] Byrne, C. and Fitzgerald, R. (1979) A unifying model for spectrum estimation, *Proceedings of the RADC Workshop on Spectrum Estimation- October 1979*, Griffiss AFB, Rome, NY.

[46] Byrne, C. and Fitzgerald, R. (1982) Reconstruction from partial information, with applications to tomography, *SIAM J. Applied Math.*, **42(4)**, pp. 933–940.

[47] Byrne, C., Fitzgerald, R., Fiddy, M., Hall, T. and Darling, A. (1983) Image restoration and resolution enhancement, *J. Opt. Soc. Amer.*, **73**, pp. 1481–1487.

[48] Byrne, C. and Fitzgerald, R. (1984) Spectral estimators that extend the maximum entropy and maximum likelihood methods, *SIAM J. Applied Math.*, **44(2)**, pp. 425–442.

[49] Byrne, C., Frichter, G., and Feuillade, C. (1990) Sector-focused stability methods for robust source localization in matched-field processing, *Journal of the Acoustical Society of America*, **88(6)**, pp. 2843–2851.

[50] Byrne, C., Haughton, D., and Jiang, T. (1993) High-resolution inversion of the discrete Poisson and binomial transformations, *Inverse Problems*, **9**, pp. 39–56.

[51] Byrne, C., Levine, B.M., and Dainty, J.C. (1984) Stable estimation of the probability density function of intensity from photon frequency counts, *JOSA Communications*, **1(11)**, pp. 1132–1135.

[52] Byrne, C., and Steele, A. (1985) Stable nonlinear methods for sensor array processing, *IEEE Transactions on Oceanic Engineering*, **OE-10(3)**, pp. 255–259.

[53] Byrne, C., and Wells, D. (1983) Limit of continuous and discrete finite-band Gerchberg iterative spectrum extrapolation, *Optics Letters*, **8 (10)**, pp. 526–527.

[54] Byrne, C., and Wells, D. (1985) Optimality of certain iterative and non-iterative data extrapolation procedures, *Journal of Mathematical Analysis and Applications*, **111 (1)**, pp. 26–34.

[55] Candy, J. (1988) *Signal Processing: The Modern Approach*, McGraw-Hill.

[56] Capon, J. (1969) High-resolution frequency-wavenumber spectrum analysis, *Proc. of the IEEE*, **57**, pp. 1408–1418.

[57] Cederquist, J., Fienup, J., Wackerman, C., Robinson, S., and Kryskowski, D. (1989) Wave-front phase estimation from Fourier intensity measurements, *Journal of the Optical Society of America A*, **6(7)**, pp. 1020–1026.

[58] Censor, Y. (1981) Row-action methods for huge and sparse systems and their applications, *SIAM Review*, **23**: 444–464.

[59] Censor, Y. and Elfving, T. (1994) A multiprojection algorithm using Bregman projections in a product space, *Numerical Algorithms*, **8**: 221–239.

[60] Censor, Y., Eggermont, P.P.B., and Gordon, D. (1983) Strong underrelaxation in Kaczmarz's method for inconsistent systems, *Numerische Mathematik*, **41**, pp. 83-92.

[61] Censor, Y., Iusem, A.N. and Zenios, S.A. (1998) An interior point method with Bregman functions for the variational inequality problem with paramonotone operators, *Mathematical Programming*, **81**: 373–400.

[62] Censor, Y. and Segman, J. (1987) On block-iterative maximization, *J. of Information and Optimization Sciences*, **8**, pp. 275-291.

[63] Censor, Y. and Zenios, S.A. (1997) *Parallel Optimization: Theory, Algorithms and Applications*, Oxford University Press, New York.

[64] Chang, J.-H., Anderson, J.M.M., and Votaw, J.R. (2004) Regularized image reconstruction algorithms for positron emission tomography, *IEEE Transactions on Medical IMaging*, **23(9)**, pp. 1165–1175.

[65] Childers, D. (ed.)(1978) *Modern Spectral Analysis*, IEEE Press, New York.

[66] Christensen, O. (2003) *An Introduction to Frames and Riesz Bases*, Birkhäuser, Boston.

[67] Chui, C. (1992) *An Introduction to Wavelets*, Academic Press, Boston.

[68] Chui, C., and Chen, G. (1991) *Kalman Filtering*, second edition, Springer-Verlag, Berlin.

[69] Cimmino, G. (1938) Calcolo approssimato per soluzioni die sistemi di equazioni lineari, *La Ricerca Scientifica XVI, Series II, Anno IX*, **1**, pp. 326–333.

[70] Combettes, P. (1993) The foundations of set theoretic estimation, *Proceedings of the IEEE*, **81 (2)**, pp. 182–208.

[71] Combettes, P. (1996) The convex feasibility problem in image recovery, *Advances in Imaging and Electron Physics*, **95**, pp. 155–270.

[72] Combettes, P. (2000) Fejér monotonicity in convex optimization, in *Encyclopedia of Optimization*, C.A. Floudas and P. M. Pardalos, Eds., Kluwer Publ., Boston, MA .

[73] Combettes, P., and Trussell, J. (1990) Method of successive projections for finding a common point of sets in a metric space, *Journal of Optimization Theory and Applications*, **67 (3)**, pp. 487–507.

[74] Cooley, J., and Tukey, J. (1965) An algorithm for the machine calculation of complex Fourier series, *Math. Comp.*, **19**, pp. 297–301.

[75] Cox, H. (1973) Resolving power and sensitivity to mismatch of optimum array processors, *Journal of the Acoustical Society of America*, **54**, pp. 771–785.

[76] Csiszár, I., and Tusnády, G. (1984) Information geometry and alternating minimization procedures, *Statistics and Decisions*, Supp. 1, pp. 205–237.

[77] Csiszár, I. (1989)A geometric interpretation of Darroch and Ratcliff's generalized iterative scaling, *The Annals of Statistics*, **17 (3)**, pp. 1409–1413.

[78] Csiszár, I. (1991) Why least squares and maximum entropy? An axiomatic approach to inference for linear inverse problems, *The Annals of Statistics*, **19 (4)**, pp. 2032–2066.

[79] Dainty, C., and Fiddy, M. (1984) The essential role of prior knowleldge in phase retrieval, *Optica Acta*, **31**, pp. 325–330.

[80] Darroch, J., and Ratcliff, D. (1972) Generalized iterative scaling for log-linear models, *Annals of Mathematical Statistics*, **43**, pp. 1470–1480.

[81] De Bruijn, N. (1967) Uncertainty principles in Fourier analysis, in *Inequalties*, O. Shisha, (ed.), Academic Press, pp. 57–71.

[82] Dempster, A.P., Laird, N.M. and Rubin, D.B. (1977) Maximum likelihood from incomplete data via the EM algorithm, *Journal of the Royal Statistical Society, Series B*, **37**: 1–38.

[83] De Pierro, A. (1995) A modified expectation maximization algorithm for penalized likelihood estimation in emission tomography, *IEEE Transactions on Medical Imaging*, **14**, pp. 132–137.

[84] De Pierro, A., and Iusem, A. (1990) On the asymptotic behaviour of some alternate smoothing series expansion iterative methods, *Linear Algebra and its Applications*, **130**, pp. 3–24.

[85] Dhanantwari, A., Stergiopoulos, S., and Iakovidis, I. (2001) Correcting organ motion artifacts in x-ray CT medical imaging systems by adaptive processing. I. Theory, *Med. Phys.*, **28(8)**, pp. 1562–1576.

[86] Dolidze, Z.O. (1982) Solution of variational inequalities associated with a class of monotone maps, *Ekonomika i Matem. Metody*, **18 (5)**, pp. 925–927 (in Russian).

[87] Dugundji, J. (1970) *Topology*, Allyn and Bacon, Inc., Boston.

[88] Eggermont, P.P.B., Herman, G.T., and Lent, A. (1981) Iterative algorithms for large partitioned linear systems, with applications to image reconstruction, *Linear Algebra and its Applications*, **40**, pp. 37–67.

[89] Everitt, B., and Hand, D. (1981) *Finite Mixture Distributions*, Chapman and Hall, London.

[90] Feuillade, C., DelBalzo, D., and Rowe, M. (1989) Environmental mismatch in shallow-water matched-field processing: geoacoustic parameter variability, *Journal of the Acoustical Society of America*, **85**, pp. 2354–2364.

[91] Feynman, R., Leighton, R., and Sands, M. (1963) *The Feynman Lectures on Physics, Vol. 1*, Addison-Wesley.

[92] Fiddy, M. (1983) The phase retrieval problem, in *Inverse Optics*, SPIE Proceedings 413 (A.J. Devaney, ed.), pp. 176–181.

[93] Fienup, J. (1979) Space object imaging through the turbulent atmosphere, *Optical Engineering*, **18**, pp. 529–534.

[94] Fienup, J. (1987) Reconstruction of a complex-valued object from the modulus of its Fourier transform using a support constraint, *Journal of the Optical Society of America A*, **4(1)**, pp. 118–123.

[95] Frieden, B. R. (1982) *Probability, Statistical Optics and Data Testing*, Springer.

[96] Gabor, D. (1946) Theory of communication, *Journal of the IEE (London)*, **93**, pp. 429–457.

[97] Gasquet, C., and Witomski, F. (1998) *Fourier Analysis and Applications*, Springer.

[98] Gelb, A. (1974) (ed.) *Applied Optimal Estimation*, written by the technical staff of The Analytic Sciences Corporation, MIT Press.

[99] Geman, S., and Geman, D. (1984) Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **PAMI-6**, pp. 721–741.

[100] Gerchberg, R. W. (1974) Super-restoration through error energy reduction, *Optica Acta*, **21**, pp. 709–720.

[101] Golshtein, E., and Tretyakov, N. (1996) *Modified Lagrangians and Monotone Maps in Optimization*, John Wiley, NY.

[102] Gordon, R., Bender, R., and Herman, G.T. (1970) Algebraic reconstruction techniques (ART) for three-dimensional electron microscopy and x-ray photography, *J. Theoret. Biol.*, **29**, pp. 471-481.

[103] Green, P. (1990) Bayesian reconstructions from emission tomography data using a modified EM algorithm, *IEEE Transactions on Medical Imaging*, **9**, pp. 84–93.

[104] Groetsch, C. (1999) *Inverse Problems: Activities for Undergraduates*, The Mathematical Association of America.

[105] Gubin, L.G., Polyak, B.T. and Raik, E.V. (1967) The method of projections for finding the common point of convex sets, *USSR Computational Mathematics and Mathematical Physics*, **7**: 1–24.

[106] Haykin, S. (1985) *Array Signal Processing*, Prentice-Hall.

[107] Hebert, T., and Leahy, R. (1989) A generalized EM algorithm for 3-D Bayesian reconstruction from Poisson data using Gibbs priors, *IEEE Transactions on Medical Imaging*, **8**, pp. 194–202.

[108] Herman, G.T. (1999) *private communication.*

[109] Herman, G. T. and Meyer, L. (1993) Algebraic reconstruction techniques can be made computationally efficient, *IEEE Transactions on Medical Imaging*, **12**, pp. 600-609.

[110] Higbee, S. (2004) *private communication.*

[111] Hildreth, C. (1957) A quadratic programming procedure, *Naval Research Logistics Quarterly*, **4**, pp. 79–85. Erratum, ibid., p. 361.

[112] Hinich, M. (1973) Maximum likelihood signal processing for a vertical array, *Journal of the Acoustical Society of America*, **54**, pp. 499–503.

[113] Hinich, M. (1979) Maximum likelihood estimation of the position of a radiating source in a waveguide, *Journal of the Acoustical Society of America*, **66**, pp. 480–483.

[114] Hoffman, K. (1962) *Banach Spaces of Analytic Functions*, Prentice-Hall.

[115] Hogg, R., and Craig, A. (1978) *Introduction to Mathematical Statistics*, MacMillan.

[116] Holte, S., Schmidlin, P., Linden, A., Rosenqvist, G. and Eriksson, L. (1990) Iterative image reconstruction for positron emission tomography: a study of convergence and quantitation problems, *IEEE Transactions on Nuclear Science*, **37**, pp. 629–635.

[117] Hubbard, B. (1998) *The World According to Wavelets*, A.K. Peters, Publ., Natick, MA.

[118] Hudson, H. M., and Larkin, R. S. (1994) Accelerated image reconstruction using ordered subsets of projection data, *IEEE Transactions on Medical Imaging*, **13**, pp. 601-609.

[119] R. Huesman, G. Klein, W. Moses, J. Qi, B. Ruetter and P. Virador (2000) *IEEE Transactions on Medical Imaging*, **19 (5)**, pp. 532–537.

[120] Hutton, B., Kyme, A., Lau, Y., Skerrett, D., and Fulton, R. (2002) A hybrid 3-D reconstruction/registration algorithm for correction of head motion in emission tomography, *IEEE Transactions on Nuclear Science*, **49 (1)**, pp. 188–194.

[121] Johnson, R. (1960) *Advanced Euclidean Geometry*, Dover.

[122] Kaczmarz, S. (1937) Angenäherte Auflösung von Systemen linearer Gleichungen, *Bulletin de l'Academie Polonaise des Sciences et Lettres*, **A35**, 355-357.

[123] Kaiser, G. (1994) *A Friendly Guide to Wavelets*, Birkhäuser, Boston.

[124] Kalman, R. (1960) A new approach to linear filtering and prediction problems, *Trans. ASME, J. Basic Eng.*, **82**, pp. 35–45.

[125] Katznelson, Y. (1983) *An Introduction to Harmonic Analysis*, Wiley.

[126] Kheifets, A. (2004) *private communication.*

[127] Körner, T. (1988) *Fourier Analysis*, Cambridge University Press.

[128] Körner, T. (1996) *The Pleasures of Counting*, Cambridge University Press.

[129] Kullback, S. and Leibler, R. (1951) On information and sufficiency, *Annals of Mathematical Statistics*, **22**: 79–86.

[130] Landweber, L. (1951) An iterative formula for Fredholm integral equations of the first kind, *Amer. J. of Math.*, **73**, pp. 615-624.

[131] Lane, R. (1987) Recovery of complex images from Fourier magnitude, *Optics Communications*, **63(1)**, pp. 6–10.

[132] Lange, K. and Carson, R. (1984) EM reconstruction algorithms for emission and transmission tomography, *Journal of Computer Assisted Tomography*, **8**: 306–316.

[133] Lange, K., Bahn, M. and Little, R. (1987) A theoretical study of some maximum likelihood algorithms for emission and transmission tomography, *IEEE Trans. Med. Imag.*, **MI-6(2)**, 106-114.

[134] Leahy, R., Hebert, T., and Lee, R. (1989) Applications of Markov random field models in medical imaging, *Proceedings of the Conference on Information Processing in Medical Imaging*, Lawrence-Berkeley Laboratory.

[135] Leahy, R., and Byrne, C. (2000) Guest editorial: Recent development in iterative image reconstruction for PET and SPECT, *IEEE Trans. Med. Imag.*, **19**, pp. 257-260.

[136] Lent, A. (1998) *private communication.*

[137] Levitan, E., and Herman, G. (1987) A maximum *a posteriori* probability expectation maximization algorithm for image reconstruction in emission tomography, *IEEE Transactions on Medical Imaging*, **6**, pp. 185–192.

[138] Liao, C.-W., Fiddy, M., and Byrne, C. (1997) Imaging from the zero locations of far-field intensity data, *Journal of the Optical Society of America -A*, **14 (12)**, pp. 3155–3161.

[139] Magness, T., and McQuire, J. (1962) Comparison of least squares and minimum variance estimates of regression parameters, *Annals of Mathematical Statistics*, **33**, pp. 462–470.

[140] Mann, W. (1953) Mean value methods in iteration, *Proc. Amer. Math. Soc.,* **4**, pp. 506–510.

[141] McLachlan, G.J. and Krishnan, T. (1997) *The EM Algorithm and Extensions*, John Wiley and Sons, New York.

[142] Meidunas, E. (2001) *Re-scaled Block Iterative Expectation Maximization Maximum Likelihood (RBI-EMML) Abundance Estimation and Sub-pixel Material Identification in Hyperspectral Imagery*, MS thesis, Department of Electrical Engineering, University of Massachusetts Lowell, Lowell MA.

[143] Meyer, Y. (1993) *Wavelets: Algorithms and Applications*, SIAM, Philadelphia, PA.

[144] Mooney, J., Vickers, V., An, M., and Brodzik, A. (1997) High-throughput hyperspectral infrared camera, *Journal of the Optical Society of America, A*, **14 (11)**, pp. 2951–2961.

[145] Motzkin, T., and Schoenberg, I. (1954) The relaxation method for linear inequalities, *Canadian Journal of Mathematics*, **6**, pp. 393–404.

[146] Narayanan, M., Byrne, C. and King, M. (2001) An interior point iterative maximum-likelihood reconstruction algorithm incorporating upper and lower bounds with application to SPECT transmission imaging, *IEEE Transactions on Medical Imaging*, **TMI-20 (4)**, pp. 342–353.

[147] Natterer, F. (1986) *Mathematics of Computed Tomography*, Wiley and Sons, NY.

[148] Natterer, F., and Wübbeling, F. (2001) *Mathematical Methods in Image Reconstruction*, SIAM.

[149] Nelson, R. (2001) Derivation of the Missing Cone, *unpublished notes*.

[150] Oppenheim, A., and Schafer, R. (1975) *Digital Signal Processing*, Prentice-Hall.

[151] Papoulis, A. (1975) A new algorithm in spectral analysis and band-limited extrapolation, *IEEE Transactions on Circuits and Systems*, **22**, pp. 735–742.

[152] Papoulis, A. (1977) *Signal Analysis*, McGraw-Hill.

[153] L. Parra and H. Barrett (1998) List-mode likelihood: EM algorithm and image quality estimation demonstrated on 2-D PET, *IEEE Transactions on Medical Imaging*, **17**, pp. 228–235.

[154] Paulraj, A., Roy, R., and Kailath, T. (1986) A subspace rotation approach to signal parameter estimation, *Proceedings of the IEEE*, pp. 1044–1045.

[155] Peressini, A., Sullivan, F., and Uhl, J. (1988) *The Mathematics of Nonlinear Programming*, Springer.

[156] Pisarenko, V. (1973) The retrieval of harmonics from a covariance function, *Geoph. J. R. Astrom. Soc.*, **30**.

[157] Poggio, T., and Smale, S. (2003) The mathematics of learning: dealing with data, *Notices of the American Mathematical Society*, **50 (5)**, pp. 537–544.

[158] Priestley, M. B. (1981) *Spectral Analysis and Time Series*, Academic Press.

[159] Prony, G.R.B. (1795) Essai expérimental et analytique sur les lois de la dilatabilité de fluides élastiques et sur celles de la force expansion de la vapeur de l'alcool, à différentes températures, *Journal de l'Ecole Polytechnique* (Paris), **1(2)**, pp. 24–76.

[160] Qian, H. (1990) Inverse Poisson transformation and shot noise filtering, *Rev. Sci. Instrum.*, **61**, pp. 2088–2091.

[161] Rockafellar, R. (1970) *Convex Analysis*, Princeton University Press.

[162] Schmidlin, P. (1972) Iterative separation of sections in tomographic scintigrams, *Nucl. Med.*, **15(1)**, Schatten Verlag, Stuttgart.

[163] Schmidt, R. (1981) *A Signal Subspace Approach to Multiple Emitter Location and Spectral Estimation*, PhD thesis, Stanford University, CA.

[164] Schuster, A. (1898) On the investigation of hidden periodicities with application to a supposed 26 day period of meteorological phenomena, *Terrestrial Magnetism*, **3**, pp. 13–41.

[165] Shang, E. (1985) Source depth estimation in waveguides, *Journal of the Acoustical Society of America*, **77**, pp. 1413–1418.

[166] Shang, E. (1985) Passive harmonic source ranging in waveguides by using mode filter, *Journal of the Acoustical Society of America*, **78**, pp. 172–175.

[167] Shang, E., Wang, H., and Huang, Z. (1988) Waveguide characterization and source localization in shallow water waveguides using Prony's method, *Journal of the Acoustical Society of America*, **83**, pp. 103–106.

[168] Smith, C. Ray, and Grandy, W.T., eds. (1985) *Maximum-Entropy and Bayesian Methods in Inverse Problems*, Reidel.

[169] Smith, C. Ray, and Erickson, G., eds. (1987) *Maximum-Entropy and Bayesian Spectral Analysis and Estimation Problems*, Reidel.

[170] Stark, H. and Yang, Y. (1998) *Vector Space Projections: A Numerical Approach to Signal and Image Processing, Neural Nets and Optics*, John Wiley and Sons, New York.

[171] Strang, G. (1980) *Linear Algebra and its Applications*, Academic Press, New York.

[172] Strang, G., and Nguyen, T. (1997) *Wavelets and Filter Banks*, Wellesley-Cambridge Press.

[173] Tanabe, K. (1971) Projection method for solving a singular system of linear equations and its applications, *Numer. Math.*, **17**, 203-214.

[174] Therrien, C. (1992) *Discrete Random Signals and Statistical Signal Processing*, Prentice-Hall.

[175] Tindle, C., Guthrie, K., Bold, G., Johns, M., Jones, D., Dixon, K., and Birdsall, T. (1978) Measurements of the frequency dependence of normal modes, *Journal of the Acoustical Society of America*, **64**, pp. 1178–1185.

[176] Tolstoy, A. (1993) *Matched Field Processing for Underwater Acoustics*, World Scientific.

[177] Twomey, S. (1996) *Introduction to the Mathematics of Inversion in Remote Sensing and Indirect Measurement*, Dover.

[178] Van Trees, H. (1968) *Detection, Estimation and Modulation Theory*, Wiley, New York.

[179] Vardi, Y., Shepp, L.A. and Kaufman, L. (1985) A statistical model for positron emission tomography, *Journal of the American Statistical Association*, **80**: 8–20.

[180] Walnut, D. (2002) *An Introduction to Wavelets*, Birkhäuser, Boston.

[181] Widrow, B., and Stearns, S. (1985) *Adaptive Signal Processing*, Prentice-Hall.

[182] Wiener, N. (1949) *Time Series*, MIT Press.

[183] Wright, W., Pridham, R., and Kay, S. (1981) Digital signal processing for sonar, *Proc. IEEE*, **69**, pp. 1451–1506.

[184] Yang, T.C. (1987) A method of range and depth estimation by modal decomposition, *Journal of the Acoustical Society of America*, **82**, pp. 1736–1745.

[185] Youla, D. (1978) Generalized image restoration by the method of alternating projections, *IEEE Transactions on Circuits and Systems*, **CAS-25 (9)**, pp. 694–702.

[186] Youla, D.C. (1987) Mathematical theory of image restoration by the method of convex projections, in: Stark, H. (Editor) (1987) *Image Recovery: Theory and Applications*, Academic Press, Orlando, FL, USA, pp. 29–78.

[187] Young, R. (1980) *An Introduction to Nonharmonic Fourier Analysis*, Academic Press.

[188] Zeidler, E. (1990) *Nonlinear Functional Analysis and its Applications: II/B- Nonlinear Monotone Operators*, Springer.

# Index