

Mathematics seems to endow one with something like a new sense.

Charles Darwin

2

Basic Mathematical Concepts

Chapter Overview

THIS CHAPTER PRESENTS the basic mathematics underlying shape analysis and classification. It starts by presenting some elementary concepts, including propositional logic, functions and complex numbers, and follows by covering important topics in linear algebra, such as vector spaces, linear transformations and metric spaces. Since several properties of shapes exhibit a differential nature, a review of the main related concepts from differential geometry and multivariate calculus is subsequently presented. Next, the key operations known as convolution and correlation are introduced and exemplified, which is followed by a review of probability and statistics, including probability distributions, autocorrelation and the Karhunen-Loève transform. The chapter concludes by presenting the main issues in Fourier analysis, from the Fourier series to discrete convolution performed in the frequency domain.

2.1 Basic Concepts

This chapter presents some basic mathematical concepts so as to provide the key to the full understanding and application of image and shape analysis, not only in the context of the present book but also of the most modern approaches covered elsewhere in the related literature. Every effort has been made to develop this chapter in an introductory fashion that should be accessible even to those with only elementary mathematical background. For those who are already familiar with the covered subjects, this chapter might still be read as a review. Although the topics are presented in a logical, progressive and integrated sequence throughout the

The *Laplacian* of a bivariate function $z = g(x, y)$ is the scalar field given by

$$\nabla^2 g(x, y) = \frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2}.$$

For example, the Laplacian of the function $g(x, y) = x^2 + 2y^2$ is $\nabla^2 g(x, y) = 2x + 4y$, which is immediately verified to be a plane.

Finally, the *divergent* of a vector field $\vec{g}(x, y) = [g_x(x, y), g_y(x, y)]$ is the scalar field defined as

$$\vec{\nabla} \cdot \vec{g}(x, y) = \frac{\partial g_x(x, y)}{\partial x} + \frac{\partial g_y(x, y)}{\partial y}.$$

For instance, the divergent of the vector function $\vec{h}(x, y) = 2x\hat{i} + 4y\hat{j}$ is the constant function $\vec{\nabla} \cdot \vec{h}(x, y) = 6$.

To probe further: *Multivariate Calculus*

Excellent classical textbooks on multivariate calculus include [Apostol, 1969; Edwards & Penney, 1998; Leithold, 1990; Williamson & Trotter, 1996]. The comprehensive textbook [Kreyszig, 1993] also covers several related aspects, while [Schey, 1997] provides a nice introduction to the main differential operators.

2.5 Convolution and Correlation

The convolution and correlation are both operations involving two functions, let us say $g(t)$ and $h(t)$, and producing as result a third function. Informally speaking, these two operations provide a means for “combining” or “mixing” the two functions as to allow important properties, such as the convolution and correlation theorems to be presented in Section 2.7.3. In addition, convolution provides the basis for several filters, and correlation provides a means for comparing two functions. These operations are presented in the following, first with respect to continuous domains, then to discrete domains.

2.5.1 Continuous Convolution and Correlation

Let $g(t)$ and $h(t)$ be two real or complex functions. The *convolution* between these functions is the univariate function resulting from the operation defined as

$$\begin{aligned} q(\tau) &= g(\tau) * h(\tau) \\ &= (g * h)(\tau) \\ &= \int_{-\infty}^{\infty} g(t) h(\tau - t) dt. \end{aligned} \tag{2.22}$$

The *correlation* between two real or complex functions $g(t)$ and $h(t)$ is the function defined as

$$\begin{aligned} q(\tau) &= g(\tau) \circ h(\tau) \\ &= (g \circ h)(\tau) \\ &= \int_{-\infty}^{\infty} g^*(t) h(\tau + t) dt. \end{aligned} \tag{2.23}$$

As is clear from the above equations, the correlation and convolution operations are similar, except that in the latter the first function is conjugated and the signal of the free variable t in the argument of $h(t)$ is inverted. As a consequence, while the convolution can be verified to be commutative, i.e.,

$$(g * h)(\tau) = \int_{-\infty}^{\infty} g(t) h(\tau - t) dt \stackrel{a=\tau-t}{=} \int_{-\infty}^{\infty} g(\tau - a) h(a) da = (h * g)(\tau)$$

we have that the correlation is not, i.e.,

$$(g \circ h)(\tau) = \int_{-\infty}^{\infty} g^*(t) h(\tau + t) dt \stackrel{a=\tau+t}{=} \int_{-\infty}^{\infty} g^*(a - \tau) h(a) da \neq (h \circ g)(\tau)$$

However, in case both $g(t)$ and $h(t)$ are real, we have

$$(g \circ h)(\tau) = \int_{-\infty}^{\infty} g(t) h(\tau + t) dt \stackrel{a=\tau+t}{=} \int_{-\infty}^{\infty} g(a - \tau) h(a) da = (h \circ g)(-\tau)$$

In other words, although the correlation of two real functions is not commutative, we still have $(g \circ h)(\tau) = (h \circ g)(-\tau)$. In case both $g(t)$ and $h(t)$ are real and even, then $(g \circ h)(\tau) = (h \circ g)(\tau)$. For real functions, the convolution and correlation are

related as

$$g(\tau) * h(-\tau) = \int_{-\infty}^{\infty} g(t)h(t-\tau) dt \stackrel{a=t-\tau}{=} \int_{-\infty}^{\infty} g(a+\tau)h(a) da = h(\tau) \circ g(\tau)$$

If, in addition, $h(t)$ is even, we have

$$g(\tau) * h(\tau) = \int_{-\infty}^{\infty} g(t)h(t-\tau) dt = h(\tau) \circ g(\tau) = (g \circ h)(-\tau)$$

An interesting property is that the convolution of any function $g(t)$ with the Dirac delta reproduces the function $g(t)$, i.e.,

$$(g * \delta)(\tau) = \int_{-\infty}^{\infty} g(t)\delta(\tau-t) dt = \int_{-\infty}^{\infty} g(\tau)\delta(\tau-t) dt = g(\tau) \int_{-\infty}^{\infty} \delta(\tau-t) dt = g(\tau)$$

An effective way to achieve a sound conceptual understanding of the convolution and correlation operations is through graphical developments, which is done in the following with respect to the convolution. Let $g(t)$ and $h(t)$ be given by Equations 2.24 and 2.25, as illustrated in Figure 2.53.

$$g(t) = \begin{cases} 1.5 & \text{if } -1 < t \leq 0, \\ 0 & \text{otherwise,} \end{cases} \tag{2.24}$$

and

$$h(t) = \begin{cases} 2 & \text{if } 0 < t \leq 2, \\ 0 & \text{otherwise.} \end{cases} \tag{2.25}$$

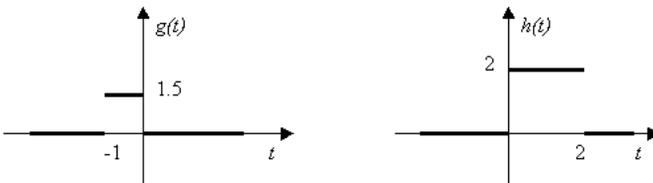


Figure 2.53: Two functions $g(t)$ and $h(t)$ to be convolved.

The first step required in order to obtain the convolution between these two functions consists in determining $h(-t)$, which is achieved, as discussed in Section 2.1.3, by reflecting $h(t)$ with respect to the y-axis, as illustrated in Figure 2.54 (a).

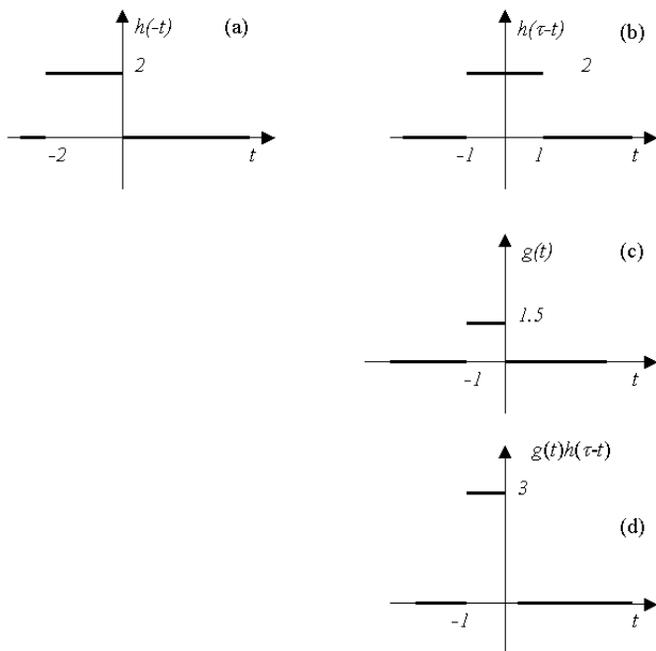


Figure 2.54: Illustration of the basic operations involved in the convolution of the functions $g(t)$ and $h(t)$. See text for explanation.

Let us now restrict our attention to a specific value of the variable τ , for example, $\tau = 1$. When this value is added to the argument of $h(-t)$, the function shifts to the right, as illustrated in Figure 2.54 (b). According to Equation 2.22, the function $h(\tau - t)$ is now multiplied by $g(t)$, shown again in Figure 2.54 (c), thus yielding the function $g(t)h(\tau - t)$ in Figure 2.54 (d).

The convolution at $\tau = 1$ is finally given by the integral of $g(t)h(\tau - t)$, which corresponds to the area below the function in Figure 2.54 (d):

$$(g * h)(\tau) = \int_{-\infty}^{\infty} g(t) h(\tau - t) dt = 3$$

Thus we have obtained the convolution value for $\tau = 1$, as shown in Figure 2.55.

By repeating the above procedure for all (and infinite) possible values of τ , we get the complete convolution shown in Figure 2.56.

The correlation can be understood in a similar manner, except for the fact that the second function is not reflected and, for complex functions, by the conjugation of the first function. Figure 2.57 shows the correlation of the above real functions, i.e., $g(t) \circ h(t)$.

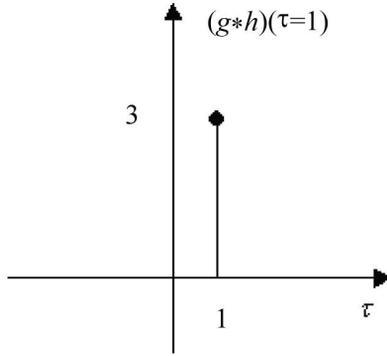


Figure 2.55: The convolution $(g * h)(\tau)$ for $\tau = 1$.

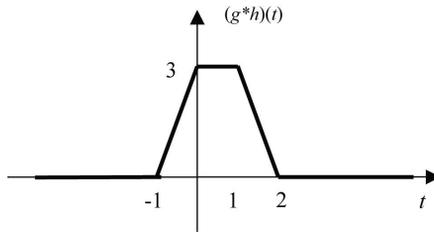


Figure 2.56: The complete convolution $(g * h)(t)$.

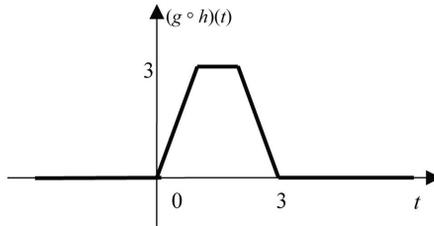


Figure 2.57: The correlation $(g \circ h)(t)$.

Let us now consider that both $g(t)$ and $h(t)$ have finite extension along the domain, i.e., $g(t), h(t) = 0$ for $t < r$ and $t > s$. Recall from Section 2.2.4 that the inner product between two functions $g(t)$ and $h(t)$ with respect to the interval $[a, b]$ is given by

$$\langle g, h \rangle = \int_a^b g^*(t) h(t) dt$$

Observe that this equation is similar to the correlation equation, except that the latter includes the parameter τ in the argument of the second function, which allows

the second function to be shifted along the x -axis with respect to the first function. As a matter of fact, for each fixed value of τ , the correlation equation becomes an inner product between the first function and the respectively shifted version of the second. This property allows us to interpret the correlation as a measure of the ‘similarity’ between the two functions with respect to a series of relative shifts between these functions. Figure 2.58 illustrates this fact. The correlation $f(t) = (g \circ h)(t)$

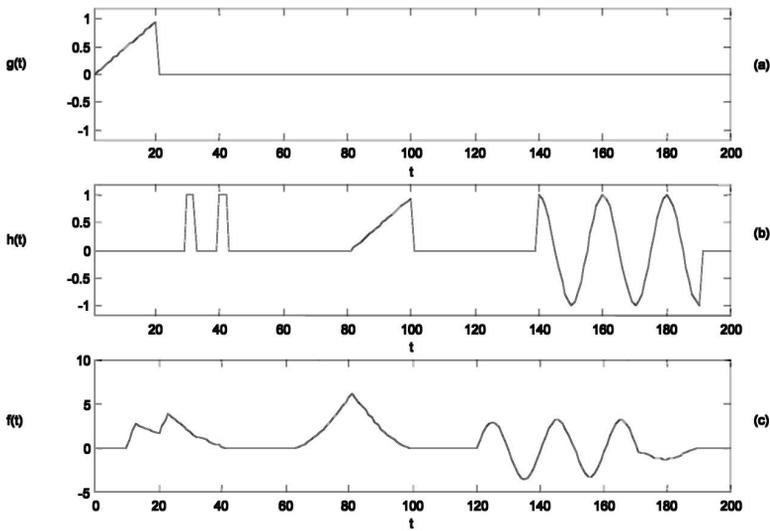


Figure 2.58: *The correlation (c) between two time limited functions $g(t)$ (a) and $h(t)$ (b) provides an indication about the similarity between the several pieces along the functions. Observe that in (c) the correlation $f(t) = (g \circ h)(t)$ is shown only for $t \geq 0$, the other half being omitted for simplicity's sake.*

slides the “template” function $g(t)$ along the function $h(t)$, calculating the inner product for each of these situations, in such a way that each correlation intensity provides an indication of the ‘similarity’ between the functions. The maximum intensity is verified for $t = 80$, which corresponds to the reference position of the sought pattern in the function $h(t)$. In other words, the correlation allows us to seek for the position where the two functions are most similar (in the sense of larger inner product).

However, it should be borne in mind that the inner products implemented by the correlation only make sense when the functions have their amplitudes properly *normalized*, or at least nearly so. For instance, if the first peak in Figure 2.58(b), centered at $t = 31$, were high enough, the correlation function would peak near $t = 10$, causing a false alarm. Among the many possible normalization schemes, it would be possible to apply the affine transformation described in Section 2.1.4

in order to map both functions to be correlated into the $[0, 1]$ interval, or to use the statistic normal transformation described in Section 2.6.2.

The convolution and correlation can be straightforwardly extended to 2D functions $g(x, y)$ and $h(x, y)$ as presented in the following:

2D Convolution:

$$(g * h)(\alpha, \beta) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g(x, y) h(\alpha - x, \beta - y) dx dy$$

2D Correlation:

$$(g \circ h)(\alpha, \beta) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g^*(x, y) h(x + \alpha, y + \beta) dx dy$$

See Chapter 3 for more detail on the application of these operations to images.

2.5.2 Discrete Convolution and Correlation

Let $g(i)$ and $h(i)$ be discrete domain (i.e., i is an integer value) functions which are zero outside $a \leq i \leq b$ and $c \leq i \leq d$, respectively, where a, b, c and d are integer values, as illustrated in Figure 2.53. Our objective in this section is to develop numerical procedures for calculating the convolution and correlation between these two functions. The discrete correlation is developed graphically in the following, in order to illustrate the basic numeric approach.

Let us start by observing that the functions g and h have lengths $N = b - a + 1$ and $M = d - c + 1$, respectively. These functions are illustrated in Figure 2.59 (a) and (b), respectively. Next, the function $h(i)$ is padded with $N - 1$ zeros at both its right and left sides, i.e., $h(i) = 0$ for $cc = c - N + 1 \leq i \leq c - 1$ and $d + 1 \leq i \leq D + N - 1 = dd$, yielding the new extended function $\tilde{h}(i)$, shown in Figure 2.59 (c).

Observe that, since $c - d < b - a$ is always verified (in fact, by construction $c < d$, implying that $c - d$ is always negative; and $a < b$, implying that $b - a$ is always positive), we ensure that $c - b < d - a$. Thus, the correlation can be organized as $(g \circ \tilde{h})(k)$ for $c - b \leq k \leq d - a$, as illustrated in Figure 2.60. The correlation is obtained by shifting the padded function $\tilde{h}(i)$ according to the above values of k , which is then multiplied by the function $g(i)$ for $a \leq i \leq b$. Finally, the results are added.

Observe that, consequently, the discrete correlation $(g \circ \tilde{h})(k)$ has length L given as

$$L = (b - c) - (a - d) + 1 = b - a + d - c + 1 = (b - a + 1) + (d - c + 1) - 1 = M + N - 1$$

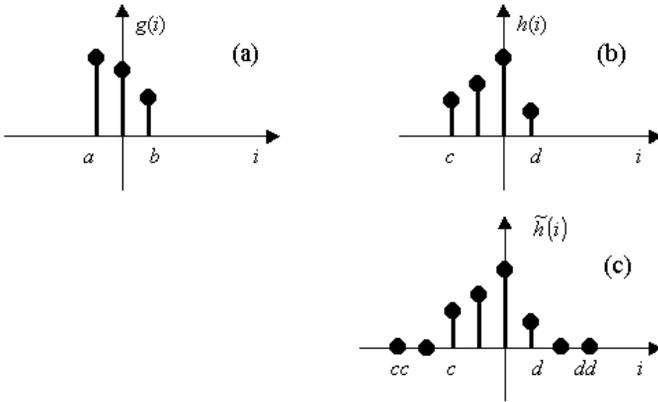


Figure 2.59: The two discrete functions $g(i)$ (a) and $h(i)$ (b) to be correlated. The padded function $\tilde{h}(i)$ (c).

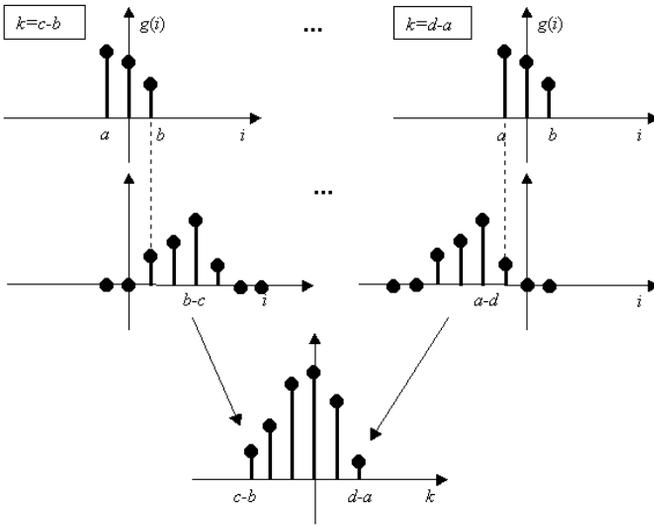


Figure 2.60: Graphical development of the discrete correlation $(g \circ h)(k)$. Since $c - d < b - a$ we have $c - b \leq k \leq d - a$. The correlation is obtained by shifting the padded function $\tilde{h}(i)$ according to the above values of k , multiplying it by the function $g(i)$, and then adding the results.

The discrete correlation $(g \circ h)(k)$ can now be expressed as

$$(g \circ h)(k) = \sum_{i=a}^b g^*(i) \tilde{h}(k+i) \quad \text{for } c - b \leq k \leq d - a. \quad (2.26)$$

By using a similar development, the *discrete convolution* $(g * h)(k)$ can be verified to be

$$(g * h)(k) = \sum_{i=a}^b g(i) \tilde{h}(k-i) \quad \text{for } c+a \leq k \leq d+b. \quad (2.27)$$

In case the adopted programming language does not allow negative indexes, the strategy presented in the box in Section 2.2.1 can be used. For discrete correlation, we have the following algorithm:

Algorithm: Discrete Correlation

1. **for** $k \leftarrow c - b$ **to** $d - a$
 2. **do**
 3. $aux \leftarrow 0$;
 4. **for** $i \leftarrow a$ **to** b
 5. **do**
 6. $aux \leftarrow \text{CONJ}(g(i - a + 1)) * h(k + i - c + 1) + aux$;
 7. $corr(1, k - c + b + 1) \leftarrow aux$;
-

where *conj* stands for the complex conjugate. For discrete convolution, we have the following algorithm:

Algorithm: Discrete Convolution

1. **for** $k \leftarrow c + a$ **to** $d + b$
 2. **do**
 3. $aux \leftarrow 0$;
 4. **for** $i \leftarrow a$ **to** b
 5. **do**
 6. $aux \leftarrow g(i - a + 1) * h(k - i - c + 1) + aux$;
 7. $conv(1, k - c - a + 1) \leftarrow aux$;
-

Observe that the above pseudo-codes, which have been prepared in order to favor intelligibility and ensure strictly positive indexing, can be further optimized.

In case the discrete functions have subsequent elements separated by Δ instead of 1, as is the situation assumed in the above developments, the discrete correlation and convolution equations should be rewritten as equations (2.28) and (2.29), respectively:

$$(g \circ h)(k) = \Delta \sum_{i=a}^b g^*(i) \tilde{h}(k+i) \quad \text{for } c-b \leq k \leq d-a, \quad (2.28)$$

$$(g * h)(k) = \Delta \sum_{i=a}^b g(i) \tilde{h}(k-i) \quad \text{for } c+a \leq k \leq d+b. \quad (2.29)$$

2.5.3 Nonlinear Correlation as a Coincidence Operator

While the correlation is frequently used as a means to compare (or match) two signals (see Figure 2.58), it presents the serious shortcoming that its result is largely affected by the signal amplitude. For instance, consider that we want to compare the two discrete signals g and h below:

$$\begin{aligned} g &= 00111100000044440000000000 \\ h &= 00000044440000000000111100 \end{aligned}$$

Figure 2.61 presents the above two sequences, shown in (a) and (b), respectively, and the result of the standard correlation between these two signals (c). Recall from Section 2.5.1 that, in principle, each peak produced by the correlation indicates a possible coincidence (or match) between portions of the signals. However, three peaks are observed in (c). Indeed, a false intermediate peak has been obtained because of the interference between the two groups of “1s” and “4s.” Moreover, the two external peaks indeed corresponding to the matches between portions of the signals present different amplitudes although they refer to partial matches of the same length (i.e., 4 values).

Much improved (actually exact) coincidence detection can be obtained by using the methodology first described in [Felsenstein et al., 1982], which involves the decomposition of the discrete signals into binary signals, yielding a nonlinear correlation. First, each signal is decomposed into a series of binary subsignals s_V , one for each of the M possible nonzero values of the elements in the original signal (observe that in the above signals $M = 2$). “1s” in one such subsignal corresponding to the value V indicate that the respective positions contained the value V , as illustrated below for the above discrete signal $g(t)$:

$$\begin{aligned} g &= 00111100000044440000000000 \\ g_1 &= 00111100000000000000000000 \\ g_4 &= 00000000000011110000000000 \end{aligned}$$

Once both signals have been decomposed, each pair of subsignals is correlated by using the standard linear correlation, yielding $s_V(t) = g_V(t) \circ h_V(t)$. The final coincidence $u(t)$ is obtained simply by adding all such correlations, i.e., $u(t) = \sum_{V=1}^M s_V(t)$. Figure 2.61 (d) presents the coincidence between the two above signals $g(t)$ and $h(t)$ obtained by using the above described methodology. It is clear that now exactly two peaks, each with maximum value of 4, have been obtained as a precise identification of the position and extension of the matches between the two discrete signals. In addition to being fully precise, the coincidence operator can be performed in $O(N \log(N))$ by using the correlation theorem in Section 2.7.3 and the fast Fourier transform to calculate the involved correlations. Further improvements to this technique have been described in [Cheever et al., 1991].

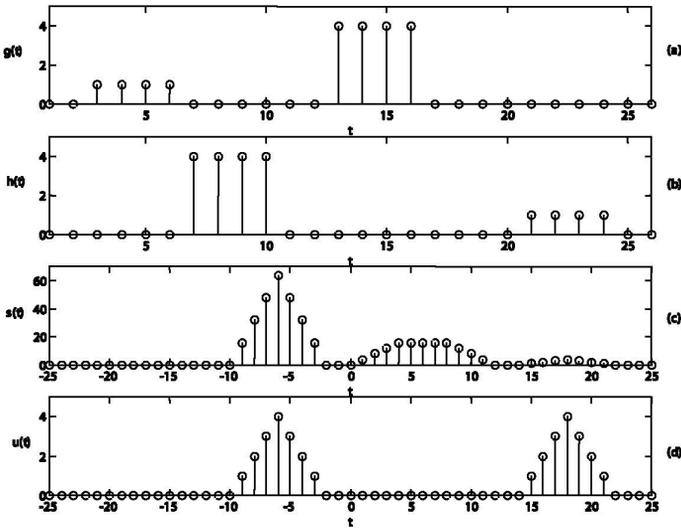


Figure 2.61: *The discrete signals $g(t)$ (a) and $h(t)$ (b). While the result of the standard correlation between these two signals (c) provides peaks with different heights for each group of “1s” and “4s,” as well as false alarms between the two peaks, the coincidence operator allows a fully precise result (d).*

To probe further: *Correlation and Convolution*

The correlation and convolution operations are usually covered in signal and image processing textbooks, such as [Brigham, 1988; Gonzalez & Woods, 1993; Morri-son, 1994; Oppenheim & Schaffer, 1975, 1989].

2.6 Probability and Statistics

Probability and statistics play a key role in image and shape analysis because of the variability of shapes and images as well as the diverse types of noise and artifacts, which can often be statistically modeled. Concepts and techniques from these two important areas are also used to treat situations of which we do not have complete knowledge. For instance, a face recognition system can be trained to recognize a set of specific faces, but it is not generally informed about all other possible faces. The current section starts by presenting the key concepts of events and probabilities

2.7 Fourier Analysis

The Fourier series and transform, the objects of the present section, represent some of the most important and interesting approaches in applied mathematics, allowing particularly useful applications in signal analysis, image processing and analysis, computer vision, differential statistics, dynamic systems, vibration analysis, psychology, acoustics and telecommunications, to name a few. Part of the importance of the Fourier approach arises from the fact that it allows a representation of a broad class of functions in terms of a linear combination of sine, cosine or complex exponential basic functions. Moreover, unlike most alternative representations of functions in terms of an orthogonal kernel, the Fourier approach exhibits an inherent and special compatibility with the signals typically found in nature, especially regarding their oscillatory and highly correlated features. As a matter of fact, the Fourier transform allows the compaction of signals by performing a decorrelation process similar and almost as effective as the statistically optimal Karhunen-Loève transform. Besides, the Fourier transform has frequently been related to the human perceptual system. For instance, our inner ears can be understood as performing spectral analysis in a way that is directly related to the Fourier series. In addition, several models of human visual perception have considered the Fourier transform as an essential component underlying processing and analysis. One of the main topics covered in this book, namely the numerical approaches to multi-resolution curvature estimation (Chapter 7), is founded on the useful but less frequently used properties of the Fourier transform, known as the derivative property. In addition, much insight about other important transforms, such as the cosine and wavelet transform, can be gained by treating them in terms of the Fourier transform.

Because of its special relevance to image and shape processing and analysis, the Fourier approach is treated to a considerable depth in this section. Although we have addressed most of the information needed for the proper understanding of the concepts and applications developed in this book, the issues related to the Fourier transform are particularly broad and can by no means be covered in an exhaustive manner here. Fortunately, there are several excellent textbooks, to which the reader is referred, providing clear and comprehensive treatment of more sophisticated related topics (see the *To Probe Further* box at the end of this section).

After some brief historical remarks, this section introduces the Fourier series, the continuous Fourier transform, its properties, frequency filtering concepts, and the discrete Fourier transform. It is observed that the sound understanding of the continuous Fourier transform, especially its properties, is *essential* for the proper application of the discrete Fourier transform to practical problems, which is discussed in some detail in the final sections of this chapter.

2.7.1 Brief Historical Remarks

The history of Fourier analysis (see, for instance [Davis & Hersh, 1999; Gullberg, 1997]) can be traced back to pioneering approaches by L. D'Alembert (1717–

1783), L. Euler (1707–1783) and D. Bernoulli (1700–1782) to the solution of the wave equation, such as that governing a vibrating string, which is a differential equation of the type

$$\frac{\partial^2 \Psi(x, t) \phi}{dx^2} = \alpha \frac{\partial^2 \Psi(x, t) \phi}{dt^2},$$

where x is position, t time, and α a constant. For the first time, Bernoulli's approach represented the initial position of the string in terms of an infinite sum of sine functions with varying frequencies—which represents the main underlying concept in the Fourier series. However, it was left to Euler to find a convenient formula for calculating the now-called *Fourier coefficients*. The interesting study of representing functions as a series of sines and cosines was resumed much later by Joseph Fourier (1768–1830), while dealing with the heat equation, a differential equation of the type

$$\frac{\partial^2 \Psi(x, t) \phi}{dx^2} = \alpha \frac{\partial \Psi(x, t) \phi}{dt},$$

as reported in his “Théorie Analytique de la Chaleur.” The main contribution of Fourier was to prove, although in an inconsistent fashion, that functions defined by pieces could also be represented in terms of infinite series of sines and cosines, the now famous *Fourier series* representation.

2.7.2 The Fourier Series

The Fourier series (or expansion) of a periodic function $g(t)$, with period $2L$, whenever it exists, is given by

$$g(t) = a_0 + \sum_{n=1}^{\infty} \left[a_n \cos\left(\frac{n\pi t}{L}\right) + b_n \sin\left(\frac{n\pi t}{L}\right) \right],$$

where

$$\begin{aligned} a_0 &= \frac{1}{2L} \int_{-L}^L g(t) dt \\ a_n &= \frac{1}{L} \int_{-L}^L g(t) \cos\left(\frac{n\pi t}{L}\right) dt \\ b_n &= \frac{1}{L} \int_{-L}^L g(t) \sin\left(\frac{n\pi t}{L}\right) dt, \end{aligned} \tag{2.42}$$

where $n = 1, 2, \dots$

These values are known as *Fourier coefficients*, and the involved sine and cosine functions are known as *kernel functions*. The frequency f of the sine and cosine

functions in the above equations can be immediately obtained by equating the argument of those functions, i.e., $\frac{n\pi t}{L}$, with $2\pi ft$, since the frequency of the functions $\cos(2\pi ft)$ or $\sin(2\pi ft)$ is, by definition, f . Therefore

$$\frac{n\pi t}{L} = 2\pi ft \Leftrightarrow f = \frac{n}{2L}.$$

Figure 2.72 illustrates the function $\cos(2\pi ft)$ for $f = 1, 2, 3, 4, 5$ and 6 .

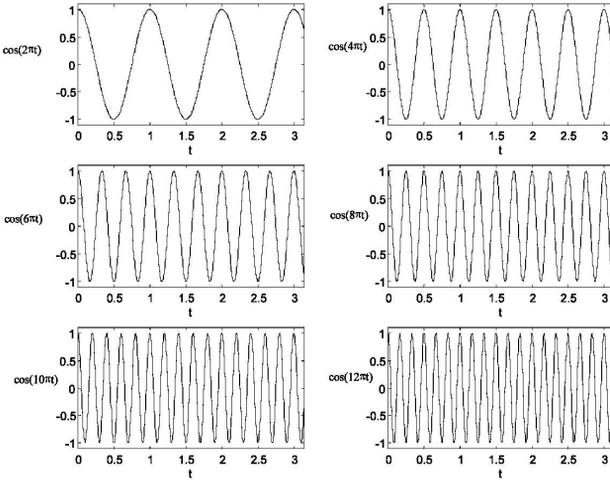


Figure 2.72: Several instances of the function $\cos(2\pi ft)$ respectively defined by $f = 1, 2, 3, 4, 5$ and 6 .

Observe that the kernel function for a specific value of n has period $T = \frac{1}{f} = \frac{2L}{n}$, implying that any of the kernel functions are periodic of period $2L$. Since both the function g , and the kernel functions have the same period $2L$, the products between these functions in equation (2.42) are also periodic of period $2L$ (the product of two periodical functions is a periodic function with the same period), and the integrals in equation (2.42) can be calculated not only between $-L$ and L , but along any interval with extent $2L$ along the time domain, as illustrated below:

$$a_n = \frac{1}{L} \int_{-L}^L g(t) \cos\left(\frac{n\pi t}{L}\right) dt = \frac{1}{L} \int_0^{2L} g(t) \cos\left(\frac{n\pi t}{L}\right) dt = \frac{1}{L} \int_{-L+2}^{L+2} g(t) \cos\left(\frac{n\pi t}{L}\right) dt.$$

It is clear from the above developments that the Fourier series of a function $g(t)$ represents a means for expressing this function as a linear combination of sine and cosine functions of distinct frequencies. It should be observed that, as implied by equation (2.42), the term a_0 corresponds to the average (or “direct current”—DC)

value of the original function along the period $2L$. Although in principle limited to periodic functions, it should be observed that any function defined over a finite domain can be expanded as a periodic function, as illustrated in the box entitled *Fourier Series*. The reader is referred to specialized literature (e.g., [Tolstov, 1976]) for the convergence conditions of the Fourier series.

An informal way to understand the Fourier series is as a “cookie recipe.” If the function is understood as the “cookie,” the Fourier coefficients can be understood as the amount of each ingredient (i.e., the amplitude of each sine and cosine functions of several frequencies) that have to be added in order to produce the cookie (i.e., the function). Table 2.7 summarizes this analogy.

Cookie Recipe	Fourier Series
The cookie.	The function.
Ingredients (i.e., flour, sugar, chocolate, etc.).	The kernel functions (i.e., sines and cosines with several frequencies).
The amount of each ingredient.	The amplitude of each kernel function, specified by the respective Fourier coefficients.
The cookie is obtained by adding together the specific amounts of ingredients.	The function is obtained as the addition of the kernel functions weighted by the Fourier coefficients (i.e., a linear combination).

Table 2.7: *The analogy between a cookie recipe and the Fourier series.*

Consider the following example:

Example: *Fourier Series*

Calculate the Fourier series of the rectangular function $g(t)$ given by

$$g(t) = \begin{cases} 1 & \text{if } -a \leq t < a, \\ 0 & \text{otherwise.} \end{cases} \quad (2.43)$$

Solution:

Since this function is not periodic, the first step consists in transforming it into a periodic function $h(t)$. A suitable period is $2L = 4a$, i.e., $L = 2a$, which yields $h(t) = g(t)$ for $-2a \leq t < 2a$ and $h(t) = h(t + 4ak)$, $k = \dots, -2, -1, 0, 1, 2, \dots$ [Figure 2.73](#) illustrates this situation with respect to $a = 1$.

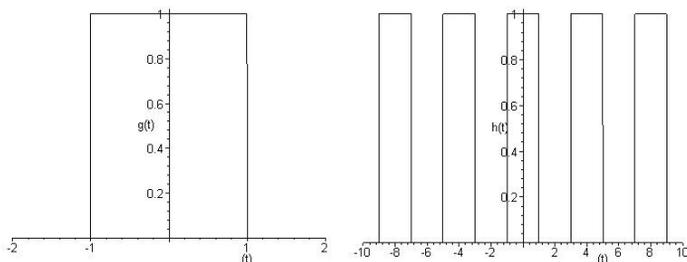


Figure 2.73: The non-periodic original function $g(t)$ and its periodic version $h(t)$, for $a = 1$.

Now the Fourier coefficients can be calculated as

$$a_0 = \frac{1}{2L} \int_{-L}^L h(t) dt = \frac{1}{4a} \int_{-2a}^{2a} h(t) dt = \frac{1}{4a} \int_{-a}^a 1 dt = \frac{1}{2},$$

$$\begin{aligned} a_n &= \frac{1}{L} \int_{-L}^L h(t) \cos\left(\frac{n\pi t}{L}\right) dt = \frac{1}{2a} \int_{-2a}^{2a} h(t) \cos\left(\frac{n\pi t}{2a}\right) dt = \frac{1}{2a} \int_{-a}^a \cos\left(\frac{n\pi t}{2a}\right) dt = \\ &= \frac{1}{2a} \left[\frac{2a}{n\pi} \sin\left(\frac{n\pi t}{2a}\right) \right]_{-a}^a \stackrel{a=1}{=} \frac{1}{n\pi} \left[\sin\left(\frac{n\pi}{2}\right) - \sin\left(-\frac{n\pi}{2}\right) \right] = \\ &= \frac{1}{n\pi} \left[\sin\left(\frac{n\pi}{2}\right) + \sin\left(\frac{n\pi}{2}\right) \right] = \frac{2}{n\pi} \sin\left(\frac{n\pi}{2}\right) = \text{sinc}\left(\frac{n}{2}\right), \end{aligned}$$

$$\begin{aligned} b_n &= \frac{1}{L} \int_{-L}^L h(t) \sin\left(\frac{n\pi t}{L}\right) dt = \frac{1}{2a} \int_{-2a}^{2a} h(t) \sin\left(\frac{n\pi t}{2a}\right) dt = \frac{1}{2a} \int_{-a}^a \sin\left(\frac{n\pi t}{2a}\right) dt = \\ &= -\frac{1}{2a} \left[\frac{2a}{n\pi} \cos\left(\frac{n\pi t}{2a}\right) \right]_{-a}^a \stackrel{a=1}{=} -\frac{1}{n\pi} \left[\cos\left(\frac{n\pi}{2}\right) - \cos\left(-\frac{n\pi}{2}\right) \right] = \\ &= -\frac{1}{n\pi} \left[\cos\left(\frac{n\pi}{2}\right) - \cos\left(\frac{n\pi}{2}\right) \right] = 0, \end{aligned}$$

where $\text{sinc}(x) = \sin(\pi x)/(\pi x)$. It is interesting to observe that a_n is zero for $n = 2, 4, 6, \dots$, and that the obtained coefficients are independent of the parameter a which, however, reappears in the function reconstruction

$$h(t) = \frac{1}{2} + \sum_{n=1}^{\infty} \left[\text{sinc}\left(\frac{n}{2}\right) \cos\left(\frac{n\pi t}{2a}\right) \right] \quad (2.44)$$

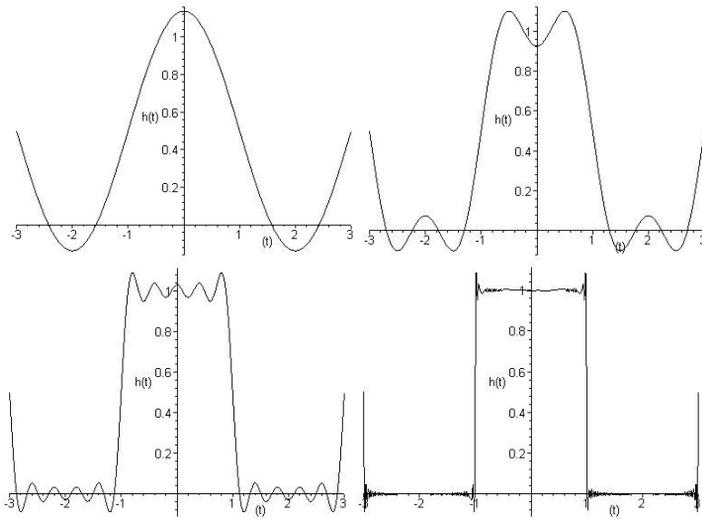


Figure 2.74: Reconstruction of the rectangular function by including an increasing number of components.

It is important to bear in mind that the above coefficients and series representation is specific to the periodical function $h(t)$, and not to $g(t)$, which can nevertheless be extracted from any of the periods of $h(t)$. At continuous points, the Fourier series tends to the function $h(t)$ as more and more terms are included—as a matter of fact, the exact convergence is only typically obtained for an infinite number of terms. At each discontinuity point P of $g(t)$, the Fourier series tends to the mean value of the respective left and right limit values, but also implies an oscillation of the series expansion around P , which is known as *Gibbs phenomenon* (see, for instance, [Kreyszig, 1993]). Figure 2.74 presents four series approximations to the above rectangular function considering increasing numbers of terms and $a = 1$.

The Fourier series can be represented in a more compact fashion by using the complex exponential as a kernel function. Given a function $g(t)$, the respective complex Fourier coefficients are given by equation (2.46), which allows the original function to be recovered by using equation (2.45).

$$g(t) = \sum_{n=-\infty}^{\infty} \left[c_n \exp \left\{ \frac{jn\pi t}{L} \right\} \right] \quad (2.45)$$

$$c_n = \frac{1}{2L} \int_{-L}^L g(t) \exp \left\{ -\frac{jn\pi t}{L} \right\} dt \quad (2.46)$$

where $n = \dots, -2, -1, 0, 1, 2, \dots$

As observed for the real Fourier series, the integration in equation (2.46) can be performed over any full period $2L$ of $g(t)$. For instance:

$$c_n = \frac{1}{2L} \int_0^{2L} g(t) \exp\left\{-\frac{jn\pi t}{L}\right\} dt \tag{2.47}$$

Consider the following example:

Example: Complex Fourier Series

Calculate the complex Fourier series of the function $g(t)$ in equation (2.43).

Solution:

First, the function $g(t)$ has to be made periodic, for instance by imposing period $2L = 4a$, as before, yielding $h(t)$. The complex coefficients can be calculated by using equation (2.46):

$$\begin{aligned} c_n &= \frac{1}{2L} \int_{-L}^L h(t) \exp\left(-\frac{jn\pi t}{L}\right) dt = \frac{1}{4a} \int_{-2a}^{2a} h(t) \exp\left(-\frac{jn\pi t}{2a}\right) dt = \\ &= \frac{1}{4a} \int_{-a}^a \exp\left(-\frac{jn\pi t}{2a}\right) dt = \frac{1}{4a} \left[-\frac{2a}{jn\pi} \exp\left(-\frac{jn\pi t}{2a}\right) \right]_{-a}^a \stackrel{a=1}{=} \\ &= -\frac{1}{j2n\pi} \left[\exp\left(-\frac{jn\pi}{2}\right) - \exp\left(\frac{jn\pi}{2}\right) \right] = \\ &= -\frac{1}{j2n\pi} \left[\cos\left(\frac{n\pi}{2}\right) - j \sin\left(\frac{n\pi}{2}\right) - \cos\left(\frac{n\pi}{2}\right) - j \sin\left(\frac{n\pi}{2}\right) \right] = \\ &= \frac{1}{n\pi} \sin\left(\frac{n\pi}{2}\right) = \frac{1}{2} \operatorname{sinc}\left(\frac{n}{2}\right) \end{aligned}$$

It should be observed that these coefficients are zero for $n = \pm 2, \pm 4, \dots$, and that $c_0 = 0.5$.

The function $h(t)$ can then be represented as

$$h(t) = \frac{1}{2} \sum_{n=-\infty}^{\infty} \left[\operatorname{sinc}\left(\frac{n}{2}\right) \exp\left\{\frac{jn\pi t}{2a}\right\} \right]$$

which is equivalent to that in equation (2.44).

2.7.3 The Continuous One-Dimensional Fourier Transform

Let $g(t)$ be a complex and not necessarily periodic function. In case its *Fourier transform* exists, it is given by

$$G(f) = \mathfrak{F} \{g(t)\} = \int_{-\infty}^{\infty} g(t) \exp \{-j2\pi ft\} dt, \quad (2.48)$$

where the variables t and f are usually called *time* and *frequency*, respectively. The *inverse Fourier transform* of $G(f)$, which returns $g(t)$, is given by

$$g(t) = \mathfrak{F}^{-1} \{G(f)\} = \int_{-\infty}^{\infty} G(f) \exp \{j2\pi ft\} df. \quad (2.49)$$

The original function and its inverse are usually represented as the *Fourier pair*:

$$g(t) \leftrightarrow G(f).$$

Observe that both $g(t)$ and $G(f)$ are, in general, complex. The function $G(f)$ is usually expressed in one of the two following representations (i) *real* and *imaginary* parts, i.e., $\text{Re}\{G(f)\}$ and $\text{Im}\{G(f)\}$, and (ii) *magnitude* (or *modulus*) and *phase*, given by

$$|G(f)| = \sqrt{[\text{Re}\{G(f)\}]^2 + [\text{Im}\{G(f)\}]^2}$$

and

$$\Phi \{G(f)\} = \arctan \left\{ \frac{\text{Im}\{G(f)\}}{\text{Re}\{G(f)\}} \right\}.$$

In addition, observe that if $g(t) \leftrightarrow G(f)$, then $g(-t) \leftrightarrow G(-f)$ and, in case $g(t)$ is real we also have $g(-t) \leftrightarrow G^*(f)$.

Since the existence of the Fourier transform is verified in practice for most functions, this topic is not covered here and the reader is referred to the literature (e.g., [Brigham, 1988]) for theoretical conditions for its existence. It is important to observe that there are several alternative definitions for the Fourier transform and its inverse [Brigham, 1988] to be found in the literature, all of which should be compatible with the Laplace transform and the energy conservation principle, i.e., Parseval's theorem.

There is an interesting analogy between the Fourier transform and the Fourier series in the sense that equation (2.48) can be understood as producing the continuous Fourier coefficients $G(f)$, which can then be used to represent the function through the inverse Fourier transform in equation (2.49). This similarity becomes evident when equations (2.45) and (2.46) are compared to equations (2.48)

and (2.49), respectively, except for the change of the sum symbol in equation (2.45) into integral in equation (2.49). In other words, the Fourier transform and its inverse can be thought of as playing the role of analysis and synthesis of the original signal, respectively. The difference between the Fourier series and transform is that in the latter the “Fourier coefficients” $G(f)$ merge to form the continuous Fourier transform of the original function instead of a series of discrete values as is the case with the Fourier series. Indeed, it can be shown (e.g., [Brigham, 1988]) that the Fourier transform can be understood as a limit situation of the Fourier series where the spacing between the Fourier coefficients tends to zero. It is precisely this fact that allows the Fourier transform to be defined for non-periodical functions (i.e., functions with infinite period).

Being directly related to the Fourier series, the Fourier transform can also be understood in terms of the “cookie recipe” analogy introduced in Section 2.7.2. That is to say, the continuous Fourier coefficients $G(f)$ provide the amount of each ingredient (again the complex exponential kernel functions) that must be linearly combined, through equation (2.49) in order to prepare the “cookie,” i.e., the original signal. The *spectral composition* of the signal is usually represented in terms of the *power spectrum* $P_g\{f\}$ of the original signal $g(t)$, which is defined as:

$$P_g\{f\} = |G(f)|^2 = G(f)^*G(f).$$

An important property of the power spectrum is that it does not change as the original function is shifted along its domain, which is explored by the so-called Fourier descriptors for shape analysis (see Section 6.5).

Consider the following example:

Example: Fourier Transform I: Periodic Functions

Calculate the Fourier transform and power spectrum of the function

$$g(t) = \exp\{-t\}, \quad 0 \leq t < \infty.$$

First, we apply equation (2.48):

$$\begin{aligned} \mathfrak{F}\{g(t)\} &= \int_0^{\infty} \exp\{-t\} \exp\{-j2\pi ft\} dt = \int_0^{\infty} \exp\{-t(j2\pi f + 1)\} dt = \\ &= -\frac{1}{1 + j2\pi f} [\exp\{-t(j2\pi f + 1)\}] \Big|_0^{\infty} = \\ &= -\left(\frac{1}{1 + j2\pi f}\right)[0 - 1] = \frac{1}{1 + j2\pi f} \left(\frac{1 - j2\pi f}{1 - j2\pi f}\right) = \frac{1 - j2\pi f}{1 + (2\pi f)^2} = G(f). \end{aligned}$$

Thus, the Fourier transform $G(f)$ of $g(t)$ is a complex function with the following real and imaginary parts, shown in Figure 2.75:

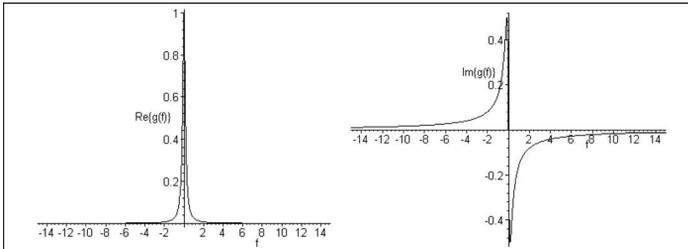


Figure 2.75: The real and imaginary parts of $G(f)$.

$$\operatorname{Re}\{G(f)\} = \frac{1}{1 + (2\pi f)^2} \quad \text{and} \quad \operatorname{Im}\{G(f)\} = \frac{-2\pi f}{1 + (2\pi f)^2}.$$

Alternatively, in the magnitude and phase representation, we have

$$|G(f)| = \sqrt{\frac{1 + 4\pi^2 f^2}{[1 + (2\pi f)^2]^2}} \quad \text{and} \quad \Phi\{G(f)\} = \arctan\{-2\pi f\}.$$

The power spectrum can be calculated as

$$P_g\{f\} = G(f)^* G(f) = \frac{1 - j2\pi f}{1 + (2\pi f)^2} \frac{1 + j2\pi f}{1 + (2\pi f)^2} = \frac{1 + (2\pi f)^2}{(1 + (2\pi f)^2)^2} = |G(f)|^2$$

and is illustrated in Figure 2.76.

Although the Fourier transform of a complex function is usually (as in the above example) a complex function, it can also be a purely real (or imaginary) function. On the other hand, observe that the power spectrum is always a real function of the frequency. Consider the following example

Example: Fourier Transform II: Aperiodic Functions

Calculate the Fourier transform of the function

$$g(t) = \begin{cases} 1 & \text{if } -a \leq t < a \\ 0 & \text{otherwise.} \end{cases}$$

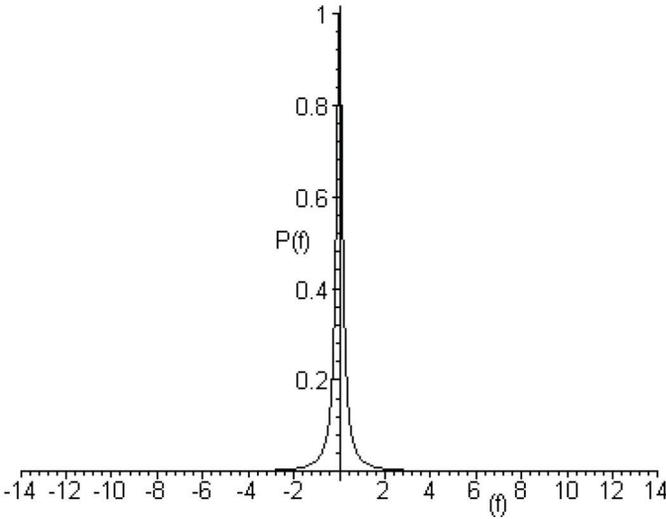


Figure 2.76: The power spectrum of the function $g(t)$.

Applying equation (2.48)

$$\begin{aligned} \mathfrak{F}\{g(t)\} &= \int_{-\infty}^{\infty} g(t) \exp\{-j2\pi ft\} dt = \int_{-a}^a 1 \cdot \exp\{-j2\pi ft\} dt = \left[\frac{-1}{j2\pi f} \exp\{-j2\pi ft\} \right]_{-a}^a = \\ &= \left(\frac{-1}{j2\pi f} \right) [\exp\{-j2\pi fa\} - \exp\{j2\pi fa\}] = \\ &= \left(\frac{-1}{j2\pi f} \right) [\cos(2\pi fa) - j \sin(2\pi fa) - \cos(2\pi fa) - j \sin(2\pi fa)] = \\ &= \frac{\sin(2\pi fa)}{\pi f} = 2a \frac{\sin(2\pi fa)}{2\pi fa} = 2a \operatorname{sinc}(2af), \end{aligned}$$

which is the purely real function shown in [Figure 2.77](#), together with its respective power spectrum, considering $a = 1$.

The Dirac delta function is particularly useful in Fourier analysis. Its transform can easily be calculated as:

$$\mathfrak{F}\{\delta(t)\} = \int_{-\infty}^{\infty} \delta(t) \exp\{-j2\pi ft\} dt = \int_{-\infty}^{\infty} \delta(t) \exp\{-j2\pi f \cdot 0\} dt = \int_{-\infty}^{\infty} \delta(t) dt = 1.$$

Since the respective inverse can be verified to exist, we have:

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

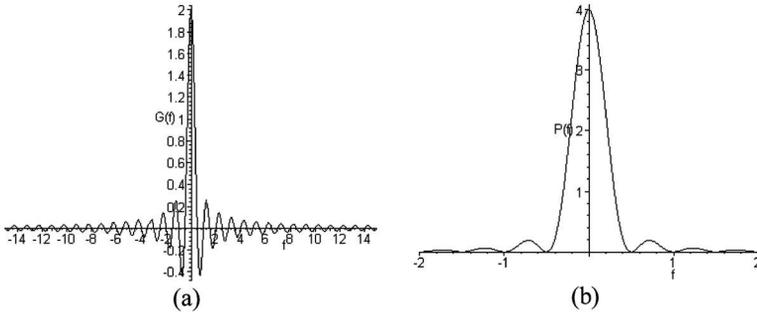


Figure 2.77: The purely real Fourier transform of the function $g(t)$ (a) and its respective power spectrum (b) for $a = 1$. Observe the different scales for the x -axis.

Another especially relevant function in Fourier analysis is the Gaussian, which defines the following Fourier pair:

$$g_{\sigma}(t) = \frac{1}{\sigma \sqrt{2\pi}} \exp \left\{ -\frac{1}{2} \left(\frac{t}{\sigma} \right)^2 \right\} \leftrightarrow G(f) = \exp \left\{ -\frac{1}{2} \left(\frac{t}{\sigma_f} \right)^2 \right\}, \quad (2.50)$$

where $\sigma_f = \frac{1}{2\pi\sigma}$. Therefore, the Fourier transform of a normalized Gaussian function is a non-normalized Gaussian function.

The Fourier transform exhibits a series of extremely useful and practical properties in signal processing and analysis, which are also essential for properly applying the discrete Fourier transform. The most important of such properties are presented and exemplified in the following sections.

Symmetry

Let

$$g(t) \leftrightarrow G(f).$$

Then

$$G(t) \leftrightarrow g(-f).$$

This property provides an interesting possibility for obtaining new transforms directly from previously known transforms, as illustrated in the following example. We have already seen that $g(t) = \delta(t) \leftrightarrow G(f) = 1$. By using the above property, we have that

$$G(t) = 1 \leftrightarrow g(-f) = \delta(-f) = \delta(f),$$

implying the new Fourier pair $1 \leftrightarrow \delta(f)$.

Time Shifting

Let

$$g(t) \leftrightarrow G(f).$$

Then

$$g(t - t_0) \leftrightarrow G(f) \exp \{-j2\pi f t_0\}.$$

Thus, the effect of shifting a function in the time domain implies that the respective Fourier transform is modulated by the complex exponential with frequency equal to the time shift value. Observe that:

$$|G(f) \exp \{-j2\pi f t_0\}|^2 = |G(f)|^2 |\exp \{-j2\pi f t_0\}|^2 = |G(f)|^2,$$

i.e., the power spectrum is not modified by time shiftings of the original function.

Example 1: Given $\delta(t) \leftrightarrow 1$, calculate the Fourier transform of $\delta(t - t_0)$. This is immediately provided by the above time shifting property as $\delta(t - t_0) \leftrightarrow \exp \{-j2\pi f t_0\}$.

Example 2: Given $g(t) = \exp \{-|t|\} \leftrightarrow \frac{2}{1+(2\pi f)^2} = G(f)$, calculate the Fourier transform of $h(t) = g(t - 2) = \exp \{-|t - 2|\}$. Again, by applying the time shifting property we obtain

$$\exp \{-|t - 2|\} \leftrightarrow \frac{2 \exp \{-4\pi j f\}}{1 + (2\pi f)^2},$$

that is graphically shown in [Figure 2.78](#).

Time Scaling

Let

$$g(t) \leftrightarrow G(f).$$

Then

$$g(at) \leftrightarrow \frac{1}{|a|} G\left(\frac{f}{a}\right).$$

Therefore, a compression (extension) of the function along the time domain implies an extension (compression) of the respective Fourier transform. This property is particularly interesting from the point-of-view of multiscale analysis (especially wavelets), since it relates the Fourier transform of scaled versions of a signal.

Example: Given $g(t) = \exp \{-|t|\} \leftrightarrow \frac{2}{1+(2\pi f)^2} = G(f)$, calculate the Fourier transform of $h(t) = g(3t) = \exp \{-|3t|\}$. By applying the above property:

$$\exp \{-|3t|\} \leftrightarrow \frac{2}{3} \frac{1}{1 + \left(\frac{2}{3}\pi f\right)^2},$$

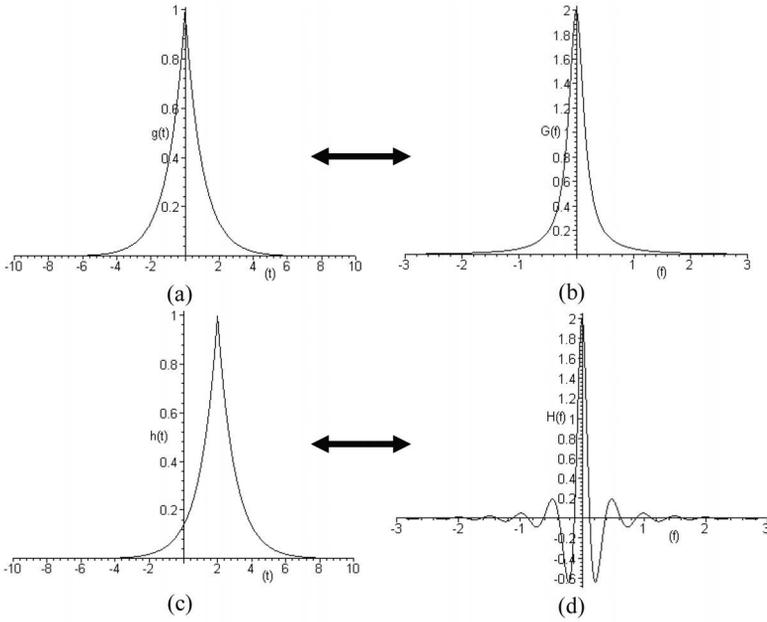


Figure 2.78: The function $g(t) = \exp\{-|t|\}$ (a) and its respective Fourier transform (b). A time shifted version of the function $g(t) = \exp\{-|t|\}$ (c) and the real part of its respective Fourier transform (d).

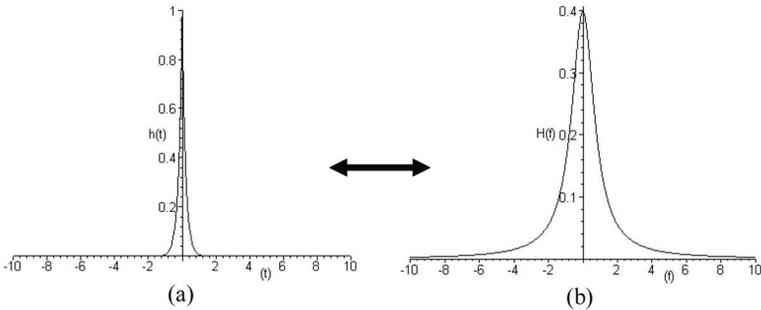


Figure 2.79: The scaled version of the function $g(t) = \exp\{-|t|\}$ (a) and its respective Fourier transform (b). Compare with Figure 2.78. The compression in the time domain implies an expansion in the frequency domain.

which is illustrated in Figure 2.79. It is clear from the above example that the effect of compressing a function in the time domain implies that its respective Fourier transform expands in the frequency domain, and viceversa.

Frequency Shifting

Let

$$g(t) \leftrightarrow G(f).$$

Then

$$g(t) \exp\{j2\pi f_0 t\} \leftrightarrow G(f - f_0).$$

This property can be understood similarly to the time shifting property presented above.

Frequency Scaling

Let

$$g(t) \leftrightarrow G(f).$$

Then

$$\frac{1}{|a|} g\left(\frac{t}{a}\right) \leftrightarrow G(af).$$

This property can be understood similarly to the time scaling property presented above.

Linearity

The Fourier transform is linear, i.e.,

$$\mathfrak{F}\{ag(t) + bh(t)\} = a\mathfrak{F}\{g(t)\} + b\mathfrak{F}\{h(t)\}.$$

This important property of the Fourier transform is very useful in practice. It can also be used to calculate new Fourier transforms. For instance, it allows the Fourier transform of the sine and cosine transform to be easily calculated as follows.

Since

$$\sin(2\pi f_0 t) = \frac{\exp\{j2\pi f_0 t\} - \exp\{-j2\pi f_0 t\}}{2j},$$

we have

$$\begin{aligned}\mathfrak{F}\{\sin(2\pi f_0 t)\} &= \mathfrak{F}\left\{\frac{\exp\{j2\pi f_0 t\} - \exp\{-j2\pi f_0 t\}}{2j}\right\} = \\ &= \frac{1}{2j}\mathfrak{F}\{\exp\{j2\pi f_0 t\}\} - \frac{1}{2j}\mathfrak{F}\{\exp\{-j2\pi f_0 t\}\} \\ &= \frac{1}{2j}(\delta(f - f_0) - \delta(f + f_0)).\end{aligned}$$

Since the inverse can be verified to exist, we have the Fourier pair

$$\sin(2\pi f_0 t) \leftrightarrow \frac{j}{2}\delta(f + f_0) - \frac{j}{2}\delta(f - f_0).$$

In a similar fashion

$$\begin{aligned}\mathfrak{F}\{\cos(2\pi f_0 t)\} &= \mathfrak{F}\left\{\frac{\exp\{j2\pi f_0 t\} + \exp\{-j2\pi f_0 t\}}{2}\right\} = \\ &= \frac{1}{2}\mathfrak{F}\{\exp\{j2\pi f_0 t\}\} + \frac{1}{2}\mathfrak{F}\{\exp\{-j2\pi f_0 t\}\} \\ &= \left(\frac{1}{2}\delta(f - f_0) + \frac{1}{2}\delta(f + f_0)\right)\end{aligned}$$

and, therefore $\cos(2\pi f_0 t) \leftrightarrow \frac{1}{2}\delta(f + f_0) + \frac{1}{2}\delta(f - f_0)$.

The Convolution Theorem

This important property of the Fourier transform is expressed as follows.

Let

$$g(t) \leftrightarrow G(f) \quad \text{and} \quad h(t) \leftrightarrow H(f).$$

Then

$$(g * h)(t) \leftrightarrow G(f)H(f)$$

and

$$g(t)h(t) \leftrightarrow (G * H)(f),$$

where $g(t)$ and $h(t)$ are generic complex functions. See [Sections 2.7.4](#) and [7.2](#) for applications of this theorem.

The Correlation Theorem

Let $g(t)$ and $h(t)$ be real functions defining the Fourier pairs $g(t) \leftrightarrow G(f)$ and $h(t) \leftrightarrow H(f)$. Then $(g \circ h)(t) \leftrightarrow G^*(f)H(f)$.

The Derivative Property

Let the generic Fourier pair $g(t) \leftrightarrow G(f)$ and a be any non-negative real value. Then

$$\frac{d^a g(t)}{dt^a} \leftrightarrow D_a(f)G(f), \quad (2.51)$$

where $D_a(f) = (j2\pi f)^a$. This interesting property, which is used extensively in the present book (see Section 7.2), allows not only the calculation of many derivatives in terms of the respective Fourier transforms, but also the definition of fractionary derivatives such as $\frac{d^{0.5}g(t)}{dt^{0.5}}$ and $\frac{d^\pi g(t)}{dt^\pi}$. This property can also be used to calculate integrals, which is done by using $a < 0$.

Example: Given $g(t) = \cos(2\pi f_0 t) \leftrightarrow \frac{1}{2}(\delta(f + f_0) + \delta(f - f_0)) = G(f)$, calculate the first derivative of $g(t)$. By applying the above property:

$$\begin{aligned} g'(t) &\leftrightarrow \frac{(j2\pi f)}{2} (\delta(f + f_0) + \delta(f - f_0)) \\ &= \frac{(j2\pi)}{2} (f\delta(f + f_0) + f\delta(f - f_0)) \\ &= \frac{(j2\pi)}{2} (f_0\delta(f + f_0) - f_0\delta(f - f_0)) \\ &= (j2\pi f_0) \frac{j}{2} (-\delta(f + f_0) + \delta(f - f_0)) \\ &\leftrightarrow -2\pi f_0 \sin(2\pi f_0 t). \end{aligned}$$

Parseval's Theorem

Let

$$g(t) \leftrightarrow G(f).$$

Then

$$\int_{-\infty}^{\infty} |g(t)|^2 dt = \int_{-\infty}^{\infty} |G(f)|^2 df.$$

This property indicates that the Fourier transform preserves the “energy” of the function. As a matter of fact, this is an ultimate consequence that the norm of a signal is preserved by an orthonormal transformation.

Parity-Related Properties

The Fourier transform of a function is determined by the parity and nature of the function $g(t)$ to be transformed (i.e., real, imaginary, or complex). Some of the most useful of such properties are summarized in Table 2.8.

$g(t)$	$G(f)$
Real and even	Real and even
Real and odd	Imaginary and odd
Imaginary and even	Imaginary and even
Imaginary and odd	Real and odd

Table 2.8: Some of the parity properties of the Fourier transform.

Discrete and Periodical Functions

First, consider the sampling function $\Psi_{\Delta t}(t)$ defined as the sum of equally spaced (by Δt) Dirac deltas, i.e., $\Psi_{\Delta t}(t) = \sum_{n=-\infty}^{\infty} \delta(t - n\Delta t)$, which defines the Fourier pair

$$\Psi_{\Delta t}(t) = \sum_{n=-\infty}^{\infty} \delta(t - n\Delta t) \leftrightarrow \frac{1}{\Delta t} \Psi_{1/\Delta t}(f) = \frac{1}{\Delta t} \sum_{n=-\infty}^{\infty} \delta\left(f - n\frac{1}{\Delta t}\right),$$

which is illustrated in Figure 2.80.

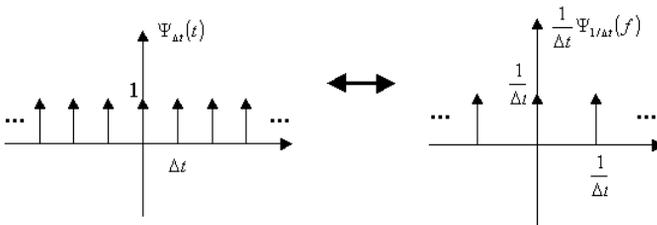


Figure 2.80: The sampling function (a) and its respective Fourier transform (b), which is also a sampling function, but with different features.

Now, let $g(t)$ be a non-periodic function entirely contained in a finite interval $r \leq t < s$ along its domain, such as

$$g(t) = \begin{cases} 1 & \text{if } -a \leq t \leq a, \\ 0 & \text{otherwise,} \end{cases}$$

where $r < -a$ and $a < s$. Let also $G(f)$ be the respective Fourier transform of $g(t)$, in this case $G(f) = 2a \operatorname{sinc}(2af)$. A periodic version $h(t)$ of the function $g(t)$, with period $2L > 2a$ can be obtained by convolving $g(t)$ with the sampling function $\Psi_{2L}(t) = 2L \sum_{n=-\infty}^{\infty} \delta(t - 2nL)$, i.e., $h(t) = g(t) * [2L\Psi_{2L}(t)]$. The coefficient $2L$ adopted for the sampling function avoids the otherwise implied scaling of the Fourier transform by a factor of $\frac{1}{2L}$. The functions $g(t)$ and $h(t)$ are illustrated in Figure 2.81 (a) and (c), respectively, considering $a = 1$ and $L = 2$. Since $g(t)$ is

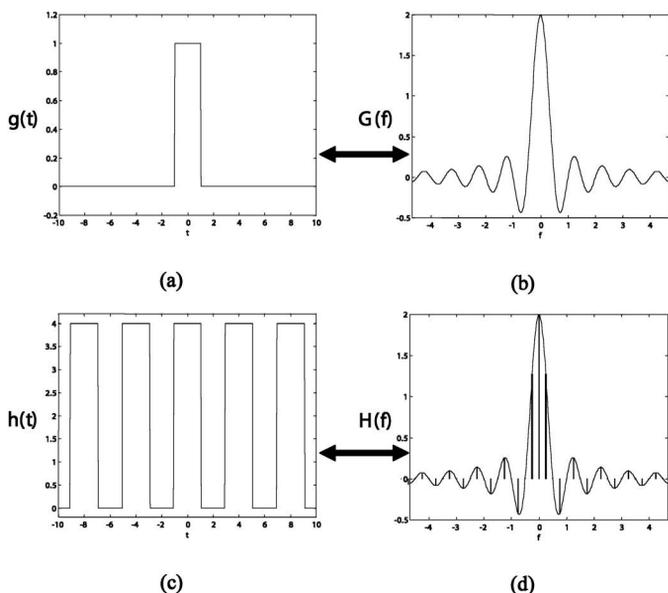


Figure 2.81: The function $g(t)$ (a) and its Fourier transform $G(f)$ (b). The periodical version $h(t) = g(t) * \Psi_{2L}(t)$ of $g(t)$, for $a = 1$ and $L = 2$ (c), and its respective Fourier transform $H(f)$ (d).

contained in a limited interval along its domain and $2L > 2a$, this process corresponds to copying $g(t)$ at the values of t that are multiples of the period $2L$. By the convolution theorem, the Fourier transform of $h(t)$ is given by the product between $G(f)$ and the Fourier transform of the sampling function, i.e.,

$$H(f) = G(f) [\Psi_{1/2L}(f)] = \sum_{i=-\infty}^{\infty} G\left(f - i\frac{1}{2L}\right) \delta\left(f - i\frac{1}{2L}\right).$$

The periodical function $h(t)$ and its respective Fourier transform $H(f)$ are shown in Figures 2.81 (b) and (d), respectively, considering $a = 1$ and $L = 2$.

A completely similar effect is observed by sampling the function $g(t)$, implying the respective Fourier transform to be periodical. The above results are summarized

below:

$g(t)$	$G(f)$
Periodical	Discrete
Discrete	Periodical

It should be observed that the Fourier transform can also be applied to periodical functions, producing as a result a necessarily *discrete* Fourier transform, i.e., a collection of Dirac deltas along the frequency domain. As a matter of fact, the Fourier transform of a periodical function $h(t)$ can be verified to produce a Fourier transform that is identical to the Fourier series of $h(t)$ [Brigham, 1988]. In other words, *for periodical functions the Fourier series becomes equal to the Fourier transform*, and the resulting transform or series is always *quantized* (or *discrete*) in the frequency space. In this sense, the Fourier series can be thought of as a particular case of the Fourier transform when the input function is periodical.

2.7.4 Frequency Filtering

One of the many important practical applications of the Fourier transform is as a means for implementing filters. To *frequency filter* a specific function or signal is henceforth understood as modifying its Fourier coefficients in a specific fashion. In this section, we consider the following three main types of filters: low-pass, high-pass and band-pass. The implementation of such filters in the frequency domain, however, is common to all these types and is achieved by multiplying the Fourier transform of the analyzed signal with a *filtering function*, and taking as result the inverse Fourier transform. That is to say, if $h(t)$ is the function to be filtered, with respective continuous Fourier transform $H(f)$, and $V(f)$ is the filtering function, the filtered version of $h(t)$, henceforth represented as $q(t)$, can be obtained as

$$q(t) = \mathfrak{F}^{-1} \{H(f) V(f)\}. \quad (2.52)$$

By considering the convolution theorem, such a filtering process can be verified to correspond to convolving, in the time domain, the function $h(t)$ with the inverse Fourier transform $v(t)$ of the filtering function $V(F)$. It is observed that there are, at least in principle, no restrictions to the type of filtering function (e.g., continuous, strictly positive, differentiable, etc.). Let us now consider each of the three types of filters individually.

As we understand from its name, a *low-pass filter* acts by attenuating the magnitude of the high frequency components in the signal, while the low frequency components are allowed to pass. Therefore, the respective filter function is expected to decrease for high values of frequency magnitude. It is important to note that such effect is relative, i.e., what matters is to attenuate the high frequency components *relative* to the low frequency components, even if all components are attenuated or magnified as a consequence. [Figure 2.82](#) presents two possible low-pass filtering functions.

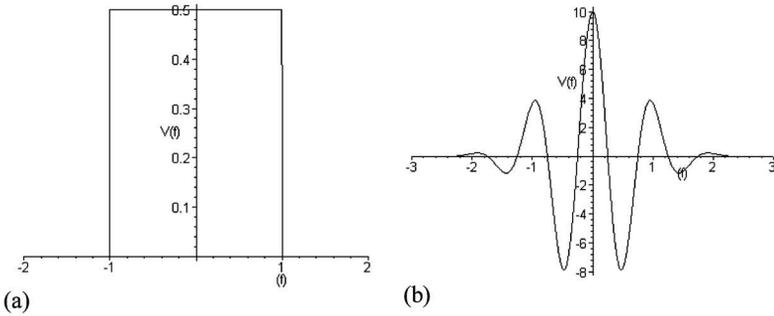


Figure 2.82: Two possible low-pass filtering functions.

Observe that the low-pass filter in Figure 2.82 (a) attenuates all frequencies, but the attenuation is smaller for the lower frequencies. Low-pass filtering tends to produce functions that are *smoother* and *more intensely correlated* than the original function $h(t)$ (see Section 2.6.5).

A typical low-pass filtering function is the zero-mean Gaussian (see Section 2.1.4). It is interesting to relate the Gaussian filtering function to its inverse Fourier transform, since this allows us to understand the filtering effect in terms of the standard deviation σ of the Gaussian respectively defined in the time domain (the higher this value, the more intense the low-pass filtering effect). Recall from Section 2.7.3 and equation (2.50) that the Gaussian in the frequency domain has as parameter $\sigma_f = 1/(2\pi\sigma)$. The henceforth adopted Gaussian filtering function $V(f)$ and its respective inverse Fourier transform (which is a Gaussian in the strict sense), are given in terms of the following Fourier transform pair:

$$g_\sigma(t) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left\{-\frac{1}{2}\left(\frac{t}{\sigma}\right)^2\right\} \leftrightarrow V(f) = \exp\{-2(\pi\sigma f)^2\}.$$

Observe that the above Gaussian filter function $V(f)$ always varies between 0 and 1. Figure 2.83 illustrates the process of Gaussian low-pass filtering.

The Fourier transform $H(f)$ (b) of the function $h(t)$ to be filtered (a) is multiplied by the filtering function $V(f)$ (c), which in this case is the Gaussian $V(f) = \exp\{-2(\pi\sigma f)^2\}$ with $\sigma = 0.1$, and the filtered function (d) is obtained by taking the inverse Fourier transform of $H(f)V(f)$. The effect of this filtering process over the original function, a cosine function corrupted by additive uniform noise, is clear in the sense that the higher frequency components of $h(t)$, i.e., the sharp oscillations along the cosine function, have been substantially attenuated, although at the expense of a substantial change in the amplitude of $h(t)$. An additional discussion about Gaussian filtering, in the context of contour processing, is presented in Section 7.2.3.

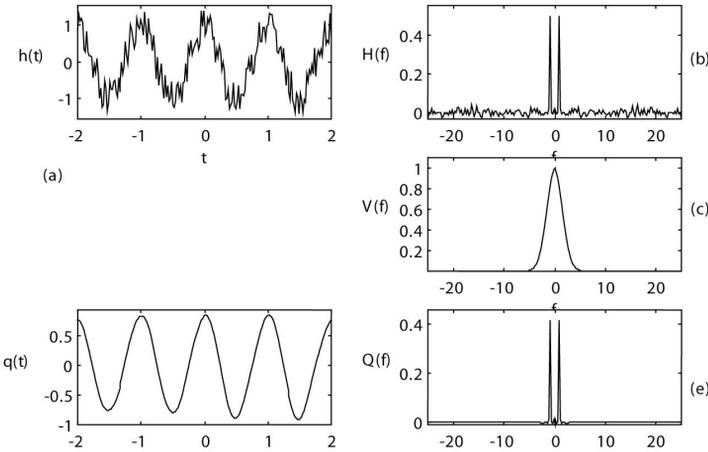


Figure 2.83: The function $h(t)$ to be low-pass filtered (a), its respective Fourier transform (b), the filtering function (in Fourier domain) (c), the filtered function $q(t)$ (d), and its respective Fourier transform (e).

The second class of filters, known as *high-pass filters*, act conversely to the low-pass filters, i.e., by attenuating the magnitude of the low frequency components of the signal, while the higher frequency components are allowed to pass. Such an attenuation should again be understood in *relative* terms. An example of high-pass filter is the *complemented Gaussian* $V(f)$, defined as

$$g_{\sigma}(t) = \delta(t) - \frac{1}{\sigma \sqrt{2\pi}} \exp \left\{ -\frac{1}{2} \left(\frac{t}{\sigma} \right)^2 \right\} \leftrightarrow V(f) = 1 - \exp \left\{ -2(\pi \sigma f)^2 \right\}.$$

It is interesting to observe that the complemented Gaussian filter function always varies between 0 and 1. This function is illustrated in Figure 2.84 for $\sigma = 0.25$.

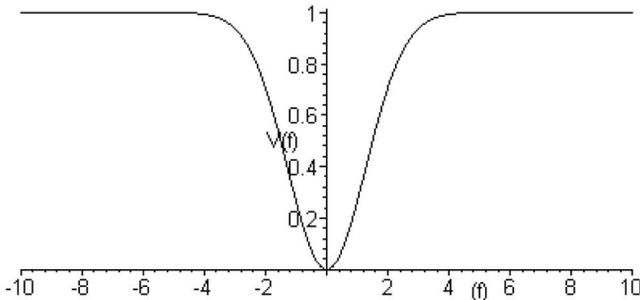


Figure 2.84: The complemented Gaussian function for $\sigma = 0.25$.

As illustrated in Figure 2.85, a high-pass filter tends to accentuate the most abrupt variations in the function being filtered, i.e., the regions where the derivative magnitude is high (in image processing and analysis, such abrupt variations are related to the image contrast). In other words, high-pass filtering reduces the

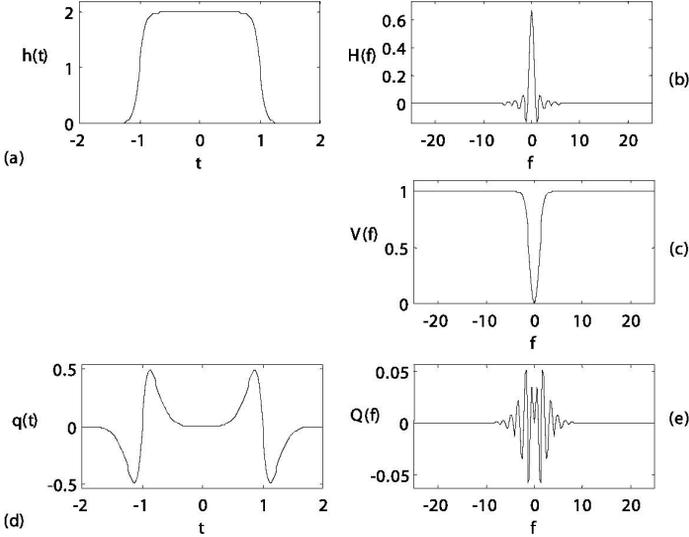


Figure 2.85: The function $h(t)$ to be high-pass filtered (a), its respective Fourier transform (b), the filtering function (c), the filtered function (d), and its respective Fourier transform (e).

correlation and redundancy degree in the original signal.

A particularly interesting type of high-pass filter in the context of the present book is related to the derivative property of the Fourier transform (see Section 2.7.3). We have already seen that, in order to obtain the first derivative of a function $h(t)$, all that we need to do is to multiply its Fourier transform by the purely imaginary function $D_1(f) = (j2\pi\sigma f)$ and take the inverse transform. As shown in Figure 2.86, this filter function presents the general shape of a high-pass filter, attenuating the low-frequency components relative to the higher frequency components. Figure 2.87 illustrates the use of this function in order to differentiate a function $h(t)$. Since the differentiation can substantially enhance high frequency noise, such an operation is usually performed by using as filter function the product of the function $D_1(f) = (j2\pi\sigma f)$ by a Gaussian function.

The filters under the category known as *band-pass* act by relatively accentuating the frequency components along a specific portion of the frequency domain. Therefore, a low-pass filter can be understood as a particular case of a band-pass filter centered at zero frequency. Gaussian functions with non-zero means in the frequency domain provide a good example of band-pass filter functions. Figure 2.88 illustrates the filtering of the function $h(t) = \cos(2\pi f_1 t) + \cos(2\pi f_2 t)$, $f_2 = 4f_1$,

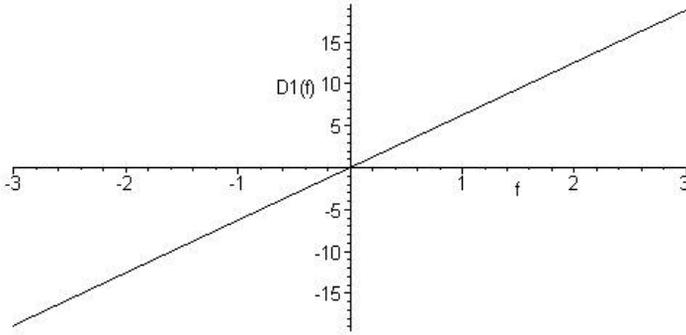


Figure 2.86: *The first derivative filter function, which is a purely imaginary function.*

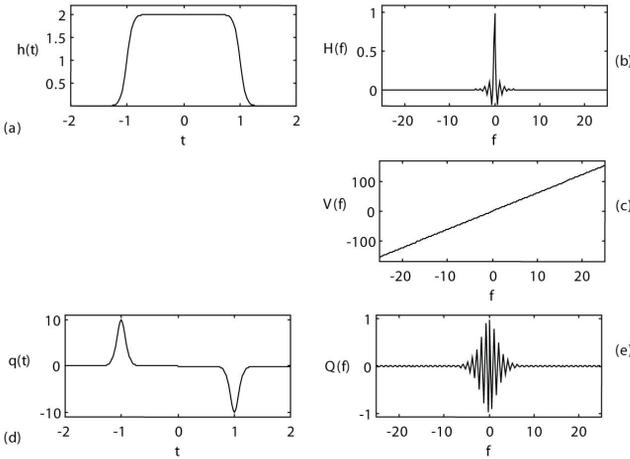


Figure 2.87: *The function $h(t)$ to be high-pass filtered by the first derivative filter (a), its respective Fourier transform (b), the filtering function (c), and the filtered function (d) and its respective Fourier transform (e). Observe that $V(f)$ is a pure imaginary function.*

by using as two band-pass Gaussian filtering functions centered respectively at $-f_2$ and f_2 , as shown in Figure 2.88 (c). Since the filter removes almost completely the lower frequency component (i.e., $\cos(2\pi f_1 t)$), the resulting filtered function consists almost exclusively of $\cos(2\pi f_2 t)$. It should be observed that, in this specific example, a similar effect could have been obtained by using a zero-mean complemented Gaussian narrow enough to attenuate the low frequencies.

Although filters are usually applied with a specific objective, such as smoothing a function, situations arise where some undesirable filtering has already taken place

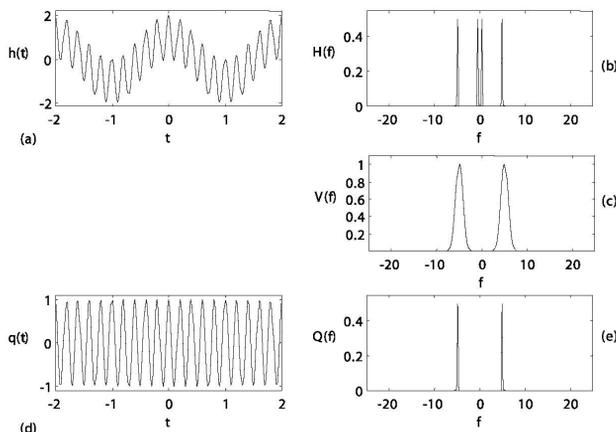


Figure 2.88: The function $h(t)$ to be band-pass filtered (a), its respective Fourier transform (b), the filter function (c), and the filtered function (d) and its respective Fourier transform (e).

and we want to recover the original function. Such a problem, which is relatively common in image processing and analysis, is called *deconvolution* (this follows from the fact that the filtering process can be alternatively understood as a convolution in the time space). If the original function $h(t)$ was filtered by a function $V(f)$, yielding $q(t)$, we may attempt to recover the original function by dividing the Fourier transform $Q(f)$ of the filtered function by the filter function $V(f)$ and taking the inverse Fourier transform as the result. Thus, the sought recovered function would be obtained as $h(t) = \mathfrak{F}^{-1}\{Q(f)/V(f)\}$. However, this process is not possible whenever $V(f)$ assumes zero value. In practice, the situation is complicated by the presence of noise in the signal and numeric calculation. Consequently, effective deconvolution involves more sophisticated procedures such as Wiener filtering (see, for instance, [Castleman, 1996]).

2.7.5 The Discrete One-Dimensional Fourier Transform

In order to be numerically processed by digital computers, and to be compatible with the discrete signals produced by digital measuring systems, the Fourier transform has to be reformulated into a suitable discrete version, henceforth called *discrete Fourier transform—DFT*.

First, the function g_i to be Fourier transformed is assumed to be a uniformly sampled (spaced by Δt) series of measures along time, which can be modelled in terms of multiplication of the original, continuous function $\tilde{g}(t)$ with the sampling function $\Psi_{\Delta t}(t) = \sum_{i=-\infty}^{\infty} \delta(t - i\Delta t)$. Second, by being the result of some measuring process (such as the recording of a sound signal) the function g_i is assumed to have *finite duration* along the time domain, let us say from time $a = i_a\Delta t$ to $b = i_b\Delta t$.

The function g_i is henceforth represented as:

$$g_i = \tilde{g}(i\Delta t).$$

Observe that the discrete function g_i can be conveniently represented in terms of the vector $\vec{g} = (g_{i_a}, g_{i_a+1}, \dots, g_{i-1}, g_i, g_{i+1}, \dots, g_{i_b-1}, g_{i_b})$. Figure 2.89 illustrates the generic appearance (i.e., sampled) of the function g_i .

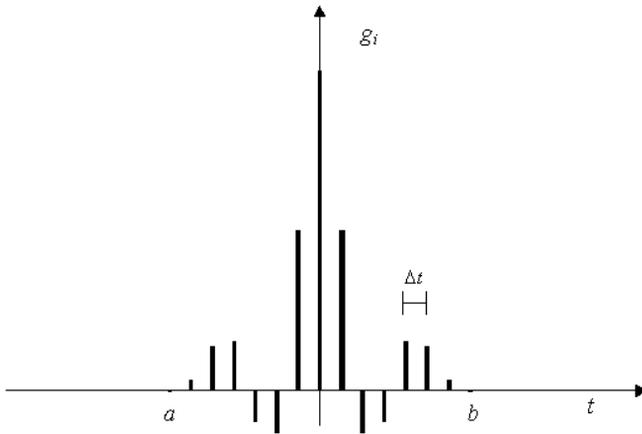


Figure 2.89: Example of a sampled function g_i to be Fourier transformed.

As seen in the previous section, the fact that the function g_i is discrete implies that the DFT output $H(f)$ is always periodical of period $\frac{1}{\Delta t}$. This important property is illustrated in Figure 2.90, which also takes into account the fact that the resulting Fourier transform is discrete (see below).

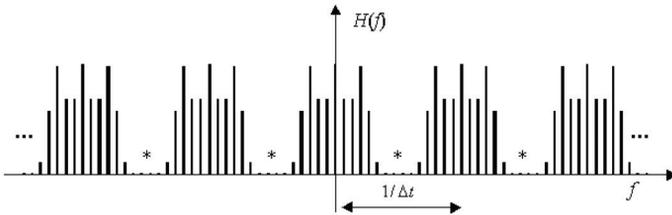


Figure 2.90: The DFT output function $H(f)$ is always discrete and periodic of period $1/\Delta t$.

Figure 2.90 also allows us to immediately derive an important result in signal processing, known as the *sampling theorem*, which relates the maximum frequency of a sampled signal g_i that can be represented in terms of the adopted sampling step Δt . Since the function $H(f)$ is periodic of period $\frac{1}{\Delta t}$, and the period centered at zero frequency extends from $-f_{\max}$ to f_{\max} , we have that $f_{\max} = \frac{1}{2\Delta t}$, which is

known as the *Nyquist rate*. Therefore, any higher frequency contained in the original continuous signal $\tilde{g}(t)$ will not be properly represented in the Fourier transform of the respective sampled signal g_i . Indeed, such too high frequencies will imply an overlapping of each of the borders of the basic period, identified by asterisks in Figure 2.90, a phenomenon known as *aliasing*. The best strategy to reduce this unwanted effect is to use a smaller value for Δt .

To be represented in a digital computer (e.g., as a vector), the Fourier transform output function $H(f)$ has to be sampled in the frequency space (the samples are assumed to be uniformly spaced by Δf), i.e., multiplied by the sampling function $\Psi_{\Delta f}(t) = \sum_{i=-\infty}^{\infty} \delta(t - i\Delta f)$, implying the periodical extension $h(t)$ (see the previous section) of g_i . Consequently, *the DFT input $h(t)$ is always a periodical extension of g_i with period $\frac{1}{\Delta f}$* . This fact is illustrated in Figure 2.91.

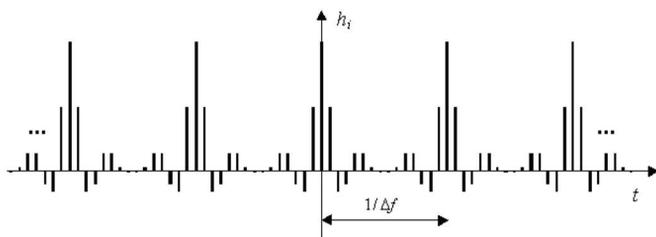


Figure 2.91: *The DFT input function h_i is always discrete and periodic of period $1/\Delta f$.*

As seen in the previous section, the fact that the input function is periodical implies that the DFT represents a numerical approximation of the *Fourier series* of $h(t)$. These important facts are summarized below:

<i>Discrete input function g_i implies the DFT output $H(f)$ to be periodical of period $\frac{1}{\Delta t}$.</i>
<i>Discrete DFT output $G(f)$ implies the input function g_i to become a periodical function $h(t)$ with period $\frac{1}{\Delta f}$.</i>
<i>The DFT input (and also the output) function can be represented as a vector.</i>
<i>The DFT corresponds to a numerical approximation of the Fourier series.</i>

Since each period of $h(t)$ has length $\frac{1}{\Delta f} = b - a = 2L$ and the samples are equally spaced by Δt , and by assuming that the period $2L$ is an integer multiple of Δt , the number N of sampling points representing the input function g_i along one period is given by:

$$N = \frac{1/\Delta f}{\Delta t} = \frac{1}{\Delta t \Delta f}.$$

Observe that we have $N = \frac{1}{\Delta t \Delta f}$ instead of $N = \frac{1}{\Delta t \Delta f + 1}$ because we want to avoid repetition at the extremity of the period, i.e., the function is sampled along the interval $[a, b)$. The number M of sampling points in any period of the output function $H(f)$ is similarly given by

$$M = \frac{1/\Delta t}{\Delta f} = \frac{1}{\Delta t \Delta f}.$$

By considering $N = M$, i.e., the number of sampling points representing the input and output DFT functions are the same (which implies vectors of equal sizes in the DFT), we have

$$N = M = \frac{1}{\Delta t \Delta f}. \quad (2.53)$$

Since the input function is always periodical, the DFT can be numerically approximated in terms of the Fourier series, which can be calculated by considering any full period of the input function $h(t)$. In order to be numerically processed, the Fourier series given by equation (2.47) can be rewritten as follows. First, the integral is replaced by the sum symbol and the continuous functions are replaced by the above sampled input and output functions. In addition, this sum is multiplied by Δt because of the numerical integration and the relationship $n = 2Lf$ is taken into account, yielding

$$\begin{aligned} G_k &= G(k\Delta f) \\ &= c_{n=2Lf} = \frac{1}{2L} \int_0^{2L} h(t) \exp\left\{-\frac{j\pi n t}{L}\right\} \\ &= \Delta t \frac{1}{2L} \sum_{i=0}^{N-1} h(i\Delta t) \exp\left\{-\frac{j\pi(2Lf)(i\Delta t)}{L}\right\} \\ &= \Delta t \frac{1}{2L} \sum_{i=0}^{N-1} h(i\Delta t) \exp\{-j2\pi(k\Delta f)(i\Delta t)\}. \end{aligned}$$

Observe that we have considered the time interval $[0, 2L)$, in order to avoid redundancies. By considering the input function as having period $2L = \frac{1}{\Delta f}$, we obtain

$$H_k = H(k\Delta f) = \Delta t \Delta f \sum_{i=0}^{N-1} h(i\Delta t) \exp\{-j2\pi(k\Delta f)(i\Delta t)\}.$$

Now, from equation (2.53) we have $\Delta t \Delta f = \frac{1}{N}$, which implies that

$$H_k = H(k\Delta f) = \frac{1}{N} \sum_{i=0}^{N-1} h(i\Delta t) \exp\left\{-\frac{j2\pi i k}{N}\right\}. \quad (2.54)$$

This equation, which is commonly known as the *discrete Fourier transform equation*, allows us to numerically estimate the Fourier series of the periodical function $h(t)$. It is easily verified that the computational execution of equation (2.54) for each specific value of k demands N basic steps, being therefore an algorithm of complexity order $O(N)$. Since the complete Fourier series involves N calculations of this equation (i.e., $k = 0, 1, \dots, N - 1$), the overall number of basic operations in the DFT algorithm is of $O(N^2)$.

2.7.6 Matrix Formulation of the DFT

Equation (2.54) can be compactly represented in matrix form, which is developed in the following. By defining the abbreviations:

$$w_{k,i} = \exp \left\{ -\frac{j2\pi ik}{N} \right\}, \quad h_i = h(i \Delta t), \quad \text{and} \quad H_k = H(k \Delta f),$$

equation (2.54) can be rewritten as:

$$H_k = \frac{1}{N} \sum_{i=0}^{N-1} w_{k,i} h_i. \tag{2.55}$$

Before proceeding with the derivation of the matrix form of the DFT, it is interesting to have a closer look at the discretized kernel function $w_{k,i} = \exp \left\{ -\frac{j2\pi ik}{N} \right\}$ (see Figure 2.92). Let us introduce $w_{ki} = w_{k,i}$ and observe that $w_{k,i} = w_{i,k}$; for instance $w_4 = w_{1,4} = w_{4,1} = w_{2,2}$. From Section 2.1, it is easy to see that the complex exponential kernel function $w_{i,k}$ in the above equation can be understood as the sequence of complex points uniformly distributed along the unit circle in the Argand plane. For instance, for $N = 8$, $i = 1$ and $k = 0, 1, \dots, N - 1$, we have the result shown in Figure 2.92.

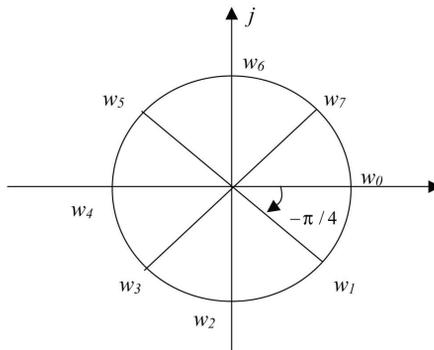


Figure 2.92: The sampled Fourier kernel function $w_{k,i} = \exp \left\{ -\frac{j2\pi ik}{N} \right\}$ for $N = 8$ and $i = 1$.

It is clear from such a construction and graphical representation that $w_{ki} = w_{k,i}$ is periodic of period N , i.e., $w_{ki} = w_{ki+pN}$, $p = \dots, -2, -1, 0, 1, 2, \dots$. It is also interesting to observe that larger values of k imply larger angular spacings between the sampled kernel values and larger frequencies for the sampled complex exponential function. The reader can easily verify that for $k > \frac{N}{2}$ the sampled complex values $w_{k,i}$ rotate in the reverse sense (i.e., counterclockwise). For instance, for $k = 7$ we have $w_{7,1} = w_7$; $w_{7,2} = w_{14} = w_6$; $w_{7,3} = w_{21} = w_5$; and so on. Ultimately, this is the reason why high speed wheels in western movies seem to rotate backwards (see also the sampling theorem in Section 2.7.5).

But now it is time to return to the matrix representation of the DFT. It follows directly from the definition of the product between matrices (equation (2.11)) that equation (2.55) can be effectively represented as the product between the $N \times N$ matrix $W_N = [w_{i,k}]$ and the $N \times 1$ vector $\vec{h} = [h_i]$, $i = 0, 1, \dots, N - 1$, resulting the $N \times 1$ vector $\vec{H} = [H_i]$, i.e.,

$$\vec{H} = \frac{1}{N} W_N \vec{h}. \quad (2.56)$$

Hence we have obtained an elegant representation of the DFT in terms of a simple matrix multiplication. In addition to providing such a compact representation, this formulation makes it clear that the DFT is indeed a *linear transformation*. Since such transformations are completely characterized by the nature of the transformation matrix, it is important to have a closer look at the matrix W_N and its properties. To begin with, we observe that this matrix is symmetric, but not Hermitian (see Section 2.2.5). Next, it is easy to verify that

$$W_N (W_N^*)^T = NI,$$

which means that the matrix W_N is *almost* unitary (Section 2.2.5). As a consequence, we can write

$$\vec{H} = \frac{1}{N} W_N \vec{h} \Rightarrow (W_N^*)^T \vec{H} = \frac{1}{N} (W_N^*)^T W_N \vec{h} \Rightarrow (W_N^*)^T \vec{H} = \frac{1}{N} (NI) \vec{h} \Rightarrow \vec{h} = (W_N^*)^T \vec{H}.$$

But, as the matrix W_N is symmetric, so is its conjugate, and we can write

$$\vec{h} = (W_N^*)^T \vec{H} \Rightarrow \vec{h} = W_N^* \vec{H}.$$

This equation, which provides a means for recovering the input vector \vec{h} from the DFT output \vec{H} , corresponds to the *inverse discrete Fourier transform*, henceforth abbreviated as *IDFT*, which clearly is also a linear transformation.

Consider the following example:

Example: DFT

Calculate the DFT of the sampled ($\Delta t = 1$) and periodic function $h(t)$ defined as

$$h(t) = \delta(t - 1) \quad \text{and} \quad h(t) = h(t + 4).$$

As always, it is interesting to start by visualizing the involved function, which is shown in Figure 2.93.

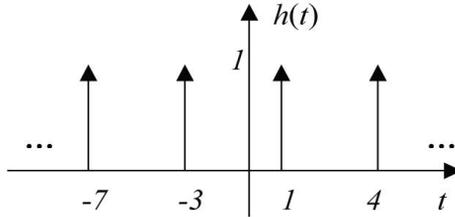


Figure 2.93: Sampled function to be Fourier transformed.

It is clear that $h(t)$ is periodic of period 4. We need to consider only the period defined from 0 to $N - 1$, yielding $\vec{h} = (0, 1, 0, 0)^T$. Now, the DFT can be calculated by using equation (2.55) as

$$\begin{aligned} \vec{H} &= \frac{1}{N} W_N \vec{h} = \frac{1}{N} \begin{bmatrix} w_{0,0} & w_{0,1} & w_{0,2} & w_{0,3} \\ w_{1,0} & w_{1,1} & w_{1,2} & w_{1,3} \\ w_{2,0} & w_{2,1} & w_{2,2} & w_{2,3} \\ w_{3,0} & w_{3,1} & w_{3,2} & w_{3,3} \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \\ &= \frac{1}{N} \begin{bmatrix} w_0 & w_0 & w_0 & w_0 \\ w_0 & w_1 & w_2 & w_3 \\ w_0 & w_2 & w_4 & w_6 \\ w_0 & w_3 & w_6 & w_9 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \end{aligned}$$

The above matrix can be straightforwardly obtained by considering the graphical representation of the sampled complex exponential function for $N = 4$ shown in Figure 2.94.

Therefore

$$\vec{H} = \frac{1}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \frac{1}{4} \begin{bmatrix} 1 \\ -j \\ -1 \\ j \end{bmatrix}$$

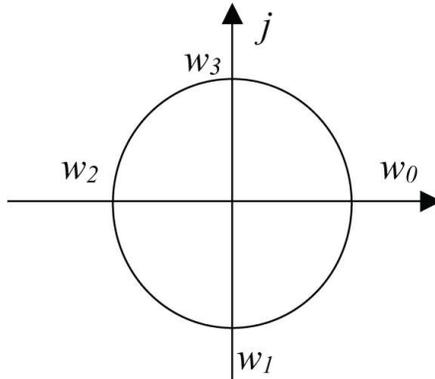


Figure 2.94: The graphical representation of the 4×4 Fourier matrix elements.

and the original signal can be immediately recovered as

$$\vec{h} = W_N^* \vec{H} = \frac{1}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & j \end{bmatrix} \begin{bmatrix} 1 \\ -j \\ -1 \\ j \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

2.7.7 Applying the DFT

In the previous developments we assumed that the function $\tilde{g}(t)$ had already been obtained from g_i by the use of a sampling and time-limiting processes. It is now time to have a closer look at how such sampled and time-limited signals can be extracted in practical situations. This problem is particularly relevant to fully understanding several situations in image processing and analysis, such as in the process of acquiring digital images by using a camera or a scanner. The basic steps involved in obtaining g_i from $\tilde{g}(t)$ are illustrated in [Figure 2.95](#).

The above diagram also makes it clear that the acquisition process presents as parameters the total number N of observations and the time interval Δt between successive observations, as well as the initial and final times, a and b , respectively, with $b - a = 2L$. Let us clarify this process by considering a practical example. Suppose we want to analyze the sound of a flute which is being played continuously at a specific pitch and intensity. We can use a microphone connected to an A/D converter interfaced to a digital computer in order to obtain the sampled and time-limited signal g_i . We start recording the signal at time a and stop at time b . Once the original continuous signal has been time limited and sampled, it is ready

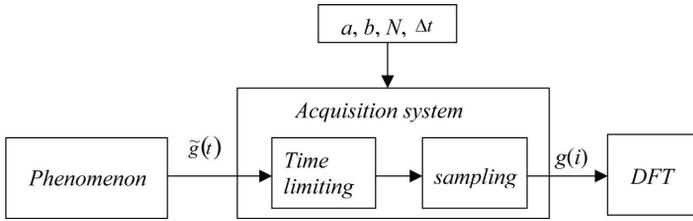


Figure 2.95: A continuous signal $\tilde{g}(t)$ has to be time-limited and previously sampled, yielding the approximated signal g_i , before being processed by the DFT. The signal g_i has N sample points, spaced by Δt , extending from time a to b , with respective extension $b - a = 2L$.

to be processed by the DFT. By choosing different values for these parameters, we can obtain different representations of the continuous signals $\tilde{g}(t)$ and, as discussed next, to have drastic effects over the quality of the obtained results. For instance, the choice of an improper value of Δt (e.g., too large) may not allow the proper representation of the involved high frequencies. In addition, observe that by considering equation (2.53), the frequency interval (and resolution) is automatically determined for each specific value of N and Δt as $\Delta f = \frac{1}{N\Delta t} = \frac{1}{b-a} = \frac{1}{2L}$.

Let us consider the acquisition process more carefully in terms of the hypothetical function $\tilde{g}(t) = \cos(2\pi t) + \cos(4\sqrt{2}\pi t)$, which is not completely unlike the signal that would be produced by a wood flute¹. This signal clearly involves two frequencies, i.e., $f_1 = 1$ Hz and $f_2 = 2\sqrt{2}$ Hz. Figure 2.96 shows this continuous signal together with its respective Fourier transform, given by

$$\tilde{G}(f) = 0.5 [\delta(f + f_1) + \delta(f - f_1) + \delta(f + f_2) + \delta(f - f_2)].$$

The fact that the observed signal has to be limited along time can be modelled by multiplying $\tilde{g}(t)$ by a windowing function $\phi(t)$ such as the rectangular function (see Section 2.1.4):

$$\phi(t) = r(t) = \begin{cases} 1 & \text{if } -a \leq t < a, \\ 0 & \text{otherwise,} \end{cases}$$

whose Fourier transform was calculated in Section 2.7.3. Observe that although we have adopted $a = b$ for simplicity's sake, the more generic situation involving $a \neq b$ can be easily developed by the reader. Figure 2.97 presents the signal $\tilde{g}(t)$ after being windowed by the above function, considering $a = 2$ seconds, as well

¹The Fourier representation of sounds such as those produced by a flute is known to involve frequencies that are multiples of the smallest frequency (the so-called *fundamental*). However, this law of multiples can become distorted as a consequence of non-linearities involved in sound production. In addition, the Fourier coefficients respective to the multiple frequencies, unlike in our hypothetical example, tend to decrease with the frequency.

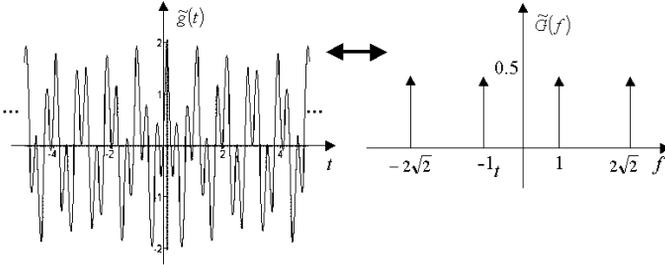


Figure 2.96: A hypothetical continuous signal $\tilde{g}(t)$ produced by a wood flute being played continuously at a specific pitch and intensity (a) and its respective Fourier transform $\tilde{G}(f)$ (b).

as its respective Fourier transform, which is obtained by convolving the Fourier transforms of $\tilde{g}(t)$ and $\phi(t)$, i.e., $\tilde{G}(f) * \Phi(f)$.

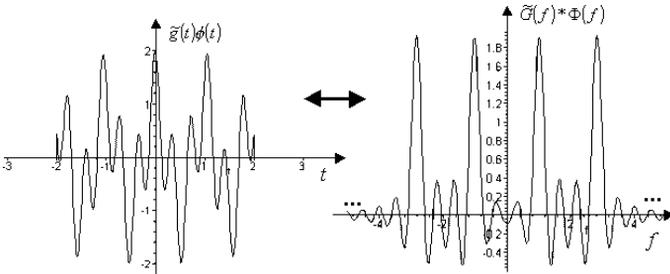


Figure 2.97: The windowed version of $\tilde{g}(t)$ and its respective Fourier transform given by $\tilde{G}(f) * \Phi(f)$.

It is clear from the oscillating nature of the Fourier transform of the rectangular function, i.e., $\tilde{G}(f)$, that the windowing of the signal $\tilde{g}(t)$ implies a ripple effect onto its respective Fourier transform. This unwanted effect can be minimized by using larger values of a or by using a smoother windowing function such as the Gaussian. As a matter of fact, observe that when a tends to infinity, its Fourier transforms tend to the Dirac delta function, and no ripple effect is implied.

Now that we have obtained a time-limited version $\tilde{g}(t) \phi(t)$ of the possibly infinite original signal $\tilde{g}(t)$, it has to be uniformly sampled before it can be represented as a vector suitable to be used in the DFT equation (2.55). As discussed in Section 2.7.5, such a sampling can be obtained by multiplying the function $\tilde{g}(t) \phi(t)$ by the sampling function $\Psi_{\Delta t}(t) = \sum_{i=-\infty}^{\infty} \delta(t - i \Delta t)$, i.e., the acquired signal finally can be represented as $g(t) = \tilde{g}(t) \phi(t) \Psi_{\Delta t}(t)$.

Figure 2.98 illustrates the sampled version of $\tilde{g}(t) \phi(t)$, assuming $\Delta t = 0.05$ second, as well as its respective Fourier transform $G(f) = \tilde{G}(f) * \Phi(f) * \left(\frac{1}{\Delta t}\right) \Psi_{1/\Delta t}(f)$. Since the convolution is commutative, the order of the sampling and time-limiting operations become irrelevant.

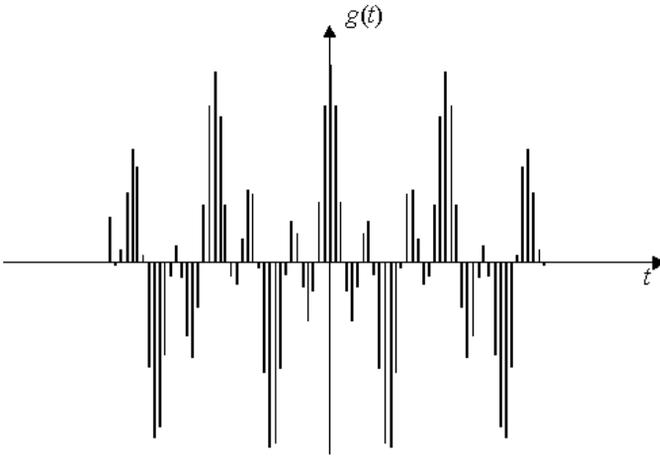


Figure 2.98: *The sampled function $g(t) = \tilde{g}(t) \phi(t) \Psi_{\Delta t}(t)$, assuming $\Delta t = 0.05$ second, and its respective Fourier transform $G(f)$.*

It is clear that the number N of samples not cancelled by the windowing function can be calculated as

$$N = \text{floor} \left\{ \frac{2a}{\Delta t} \right\}.$$

In the above case, since $\Delta t = 0.05$ and $a = 2$, we have $N = \frac{2a}{\Delta t} = \frac{4}{0.05} = 80$ sampling points. We also have from equation (2.53) that $\Delta f = \frac{1}{N\Delta t} = 0.25$, and from the sampling theorem that $f_{\text{max}} = \frac{1}{2\Delta t} = 10$ Hz. As discussed in Section 2.7.5, the sampling process implies the Fourier transform of g_i to be periodical with period $\frac{1}{\Delta t} = 20$ Hz. It can be verified that the Fourier transform $G(f)$ provides a good approximation for the original Fourier transform $\tilde{G}(f)$, except for the ripple effect caused by the windowing operation.

The DFT of g_i can now be determined by applying equation (2.54), i.e.,

$$G(k \Delta f) = \frac{1}{N} \sum_{i=0}^{N-1} g(i \Delta t) \exp \left\{ -\frac{j2\pi ik}{N} \right\}.$$

A complete period (starting at zero frequency) of the respective DFT output $H(k \Delta f)$ is shown in Figure 2.99.

A more careful analysis of the Dirac delta approximations obtained in Figure 2.99 indicates that the lower frequency peaks (marked with asterisks), with respect to $f_1 = 1$ Hz, have been better represented (there is less ripple around it

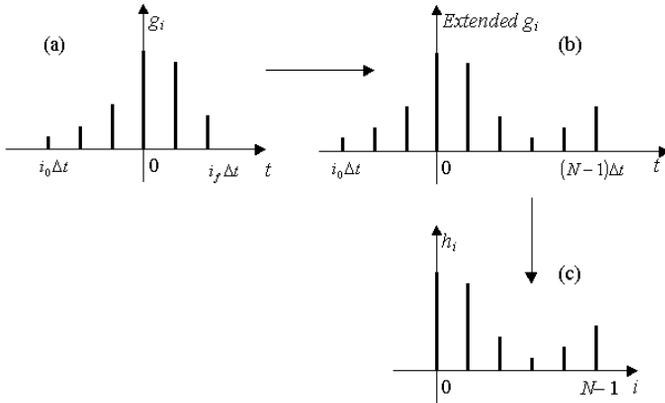


Figure 2.99: The DFT output function $H(k \Delta f)$. The pair of lower frequency peaks for $f_1 = 1$ Hz are indicated by asterisks.

and the amplitude is exactly as expected, i.e., 0.5) than the deltas for $f_2 = 2\sqrt{2}$ Hz. This is because f_1 is an integer multiple of $\Delta f = \frac{1}{N\Delta t} = 0.25$ Hz, while f_2 is not. Indeed, a complete cancellation of the ripple effect is observed in such a multiple situation, because the zero crossings of $\tilde{G}(f)$ can be verified to coincide with the sampling points. However, this welcomed effect cannot usually be guaranteed in practical situations, and the unwanted rippling effect has to be somehow alleviated, for instance by using a smoother window function. Figure 2.100 illustrates this possibility considering as windowing function $\phi(t) = \exp\{-t^2\} r(t)$, i.e., the product of a Gaussian with the rectangular function (a truncated Gaussian function).

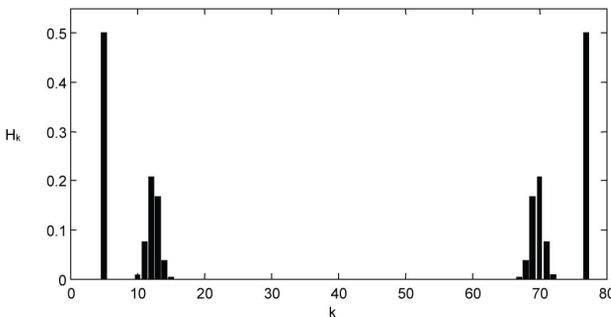


Figure 2.100: DFT obtained by windowing $\tilde{g}(t)$ with the truncated Gaussian function $\phi(t) = \exp\{-t^2\} r(t)$.

The ripple attenuation is readily verified, though at the expense of a decrease in the amplitude of the coefficients related to $f_2 = 2\sqrt{2}$ Hz. The reader is referred to the literature (e.g., [Ingle & Proakis, 1997; Kamen & Heck, 1997; Papoulis,

1984]) for a more detailed discussion about several windowing functions and their respective properties.

Another important practical aspect related to the DFT application concerns the fact that in most DFT algorithms (and especially many FFT algorithms—see Section 2.7.8) the function g_i is assumed to initiate at time zero, i.e., the origin of the time axis, and extend up to $N - 1$, so that g_i can be represented as the vector $\vec{h} \mid h_i = g(i \Delta t), i = 0, 1, \dots, N - 1$. If this is indeed the case, the DFT execution is immediate, being only needed to use the number of samples N . However, special attention is required when dealing with functions extending into the negative portion of the time domain, such as that considered in the above examples. It should be borne in mind that the time zero while acquiring the signal is a relative reference that is important and must be taken into account, otherwise the time shifting effect described in Section 2.7.3 will imply a modulation of the respective Fourier transform.

Let us now consider the situation where the signal extends into the negative portion of its domain in more detail. Basically, what is needed is to move the left portion of g_i , i.e., that in the negative portion of the time axis (excluding the zero), to the right-hand side of the function (see Figure 2.101). If $g_i = g(i \Delta t)$ starts at

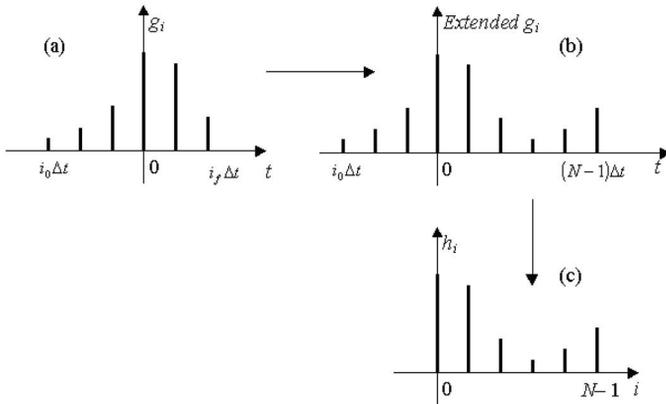


Figure 2.101: Most DFT (and FFT) algorithms require the input function $h(t)$ to be defined from 0 to $N - 1$. In case the function g_i , from which h_i is obtained, extends into the negative time domain (a), it needs to be conveniently reorganized (b) and (c). The vector \vec{h} to be used as input to the DFT can then be obtained by taking the sampled points in the non-negative region of the time axis.

i_a and terminates at i_b , i.e., $i = i_a, \dots, -1, 0, 1, \dots, i_b$ (observe that $i_a < 0$), the above mentioned translation operation can be implemented by using the following algorithm:

1. $N \leftarrow i_b - i_a + 1;$
2. **for** $k \leftarrow i_a$ **to** -1
3. **do**
4. $g_{(k+N)\Delta t} = g_{k\Delta t};$

Once this extended version of g_i is obtained, all that remains to be done is to copy it into the DFT input vector $\vec{h} \mid h_i = g(i\Delta t), i = 0, 1, \dots, N - 1$, which, as discussed in this section, will be considered as being periodical of period N . This process is illustrated in Figure 2.101.

Figure 2.102 presents the DFT input function $h(i)$ (a) and its respective Fourier transform (b) as typically produced by DFT algorithms.

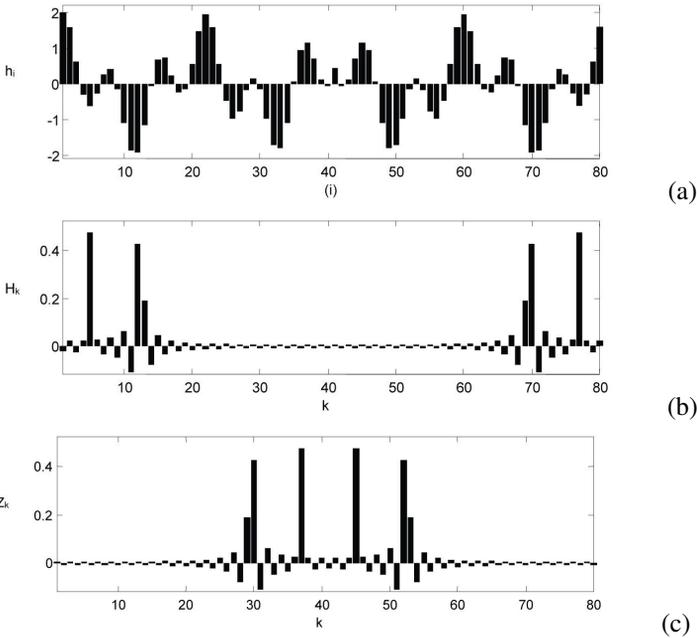


Figure 2.102: The input (a) and output (b) functions for a typical DFT algorithm. The obtained output can be represented with the zero frequency at its middle (c).

It is clear that the leftmost point in the DFT output is at zero frequency. In case we want to have the zero frequency in the middle of function, a transposition process similar to the above described has to be applied, yielding the reorganized signal Z_k . A possible strategy for doing so is presented in terms of the following algorithm:

1. $n \leftarrow \text{FLOOR}((N-1)/2);$
2. **for** $k \leftarrow 0$ **to** $N - n - 1$
3. **do**
4. $Z_{k+n} \leftarrow H_k;$
5. **for** $k \leftarrow N - n$ **to** $N - 1$
6. **do**
7. $Z_{k+n-N} \leftarrow H_k;$

Let us conclude this section by characterizing two alternative practical situations typically met while applying the DFT, which are presented in the following:

Alternative 1: Imposed frequency sampling interval.	
1	Estimate the maximum frequency f_{\max} to be found in the continuous signal $\tilde{g}(t)$. In case this is not possible, assume the highest possible value for f_{\max} .
2	Calculate $\Delta t = \frac{1}{2f_{\max}}$.
3	Define a suitable frequency sampling interval Δf . The frequency resolution is then $\frac{\Delta f}{2}$.
4	Determine the number of samples (i.e., the dimension of the DFT input vector) as $N = \frac{1}{\Delta t \Delta f}$.
Alternative 2: Imposed number of samples.	
1	Estimate the maximum frequency f_{\max} to be found in the continuous signal $\tilde{g}(t)$. In case this is not possible, assume the highest possible value for f_{\max} .
2	Calculate $\Delta t = \frac{1}{2f_{\max}}$.
3	Define a suitable value for N . The higher this value, the slower the DFT.
4	Determine the frequency sampling interval as $\Delta f = \frac{1}{N \Delta t}$.

The choice between these two alternatives will depend on practical constraints. First, it should be observed that the DFT parameters Δt , Δf and N are linked by equation (2.53), i.e., $N = \frac{1}{\Delta t \Delta f}$. Thus, having chosen two of these parameters, the third is automatically defined. Ideally, we would wish to have both Δf and Δt as small as possible, in order to allow maximum frequency resolution and the largest maximum representable frequency. However, the problem is that both these facts imply higher values for N , and thus the DFT becomes slower. The first alternative, by defining N as a consequence of the choice of Δf , is therefore suitable for situations where there is not too much concern about the execution time. The second alternative should be used otherwise.

Before proceeding to the next section, it is important to note that, although in this section all the Fourier transforms were purely real (because the considered time signals were all real and even), this is by no means the general case. In other words, the DFT results generally involve both real and imaginary parts. In addition,

although we were limited to purely real input functions, it is also possible to use complex functions. The above presented developments, however, are immediately applicable to such situations.

2.7.8 The Fast Fourier Transform

The *fast Fourier transform*, hence *FFT*, should be understood simply as a numeric method for computing the DFT in a fast manner, its result being identical except for round-off noise (i.e., the noise produced by the limited precision in representing numbers and performing operations in digital computers). The effectiveness of this technique—indeed of this class of techniques, since there are many FFT algorithms—cannot be overlooked, making a real difference in practice, especially when the number of samples representing the signals is relatively large. Since the time savings allowed by the FFT is considerable, the matrix calculation of the DFT in equation (2.56) is rarely used in practice.

The main advantage of the FFT methods arises from the fact that they remove many of the redundancies in the DFT matrix calculation (see [Brigham, 1988], for instance). As a matter of fact, the number of basic operations (i.e., additions and multiplications) involved in the FFT algorithm is of order $O(N \log N)$, while (as seen in Section 2.7.5), the standard DFT implies $O(N^2)$.

Several FFT algorithms, including the classical Cooley and Tukey's approach, require the value of N to be an integer power of two, such as 32, 64, 128 and so on. In such cases, the method is said to be of *radix 2*. It is also possible to have alternative radices, such as 4, 5, 6 and so on [Brigham, 1988]. In the general case, N must be an integer power of the radix. In practice, this requirement of having $N = (\text{radix})^k$, $k = 0, 1, 2, \dots$ can be easily met by using the smallest value of k such as the vector size is smaller than 2^k and filling up the unused positions in the DFT input vector \vec{h} with zeros. However, observe that this procedure may cause a discontinuity and, consequently, introduce oscillations in the recovered signal because of the Gibbs effect. A method to alleviate this problem consists in filling up the first half of the unused portion of the vector with the same value as the last in the original function, and the second half as the first value in the original function (recall that the DFT implies that the function represented by this vector is periodical).

The simplicity and small number of numerical operations implied by the FFT, involving mostly complex products, has motivated the whole family of new devices known as *digital signal processors*, namely circuits or integrated circuits capable of processing the FFT very quickly by using dedicated hardware components (such as multipliers). Such a tendency has allowed the FFT to be processed at very high speeds allowing real-time applications for most situations.

In spite of the FFT importance and usefulness, we do not present an algorithm for its calculation in the present book for the following three reasons: (a) the proper explanation of FFT algorithms is relatively extensive; (b) there are excellent books covering the FFT (for instance, see [Brigham, 1988]); and (c) nowadays, it is un-

likely that the reader will have to implement the FFT, since it is broadly available in mathematical environments (e.g., MATLAB[®], Scilab[®], Maple[®] and so on) and also as ready-to-use libraries and objects to be used in programming languages and environments.

2.7.9 Discrete Convolution Performed in the Frequency Domain

One of the most frequent applications of the discrete Fourier transform is as a means of performing fast convolution or correlation between two sampled functions or signals. Such operations are typically applied to compare two signals (correlation) or to filter a signal (see Section 2.7.4). While Section 2.5.2 has presented equations and algorithms for performing convolution and correlation between two discrete signals, the calculation of such operations in the time domain can be too slow, especially when the signals involve many samples. The computational advantage of performing convolution in the frequency domain by using the FFT is still more substantial for higher dimensional signals, such as images.

We have already seen in Section 2.7.7 how to perform the DFT over discrete signals. In addition to the procedures presented, the calculation of the convolution (or correlation) by using the FFT (or DFT) demands particular attention to the fact that the DFT input signal is necessarily periodical. Therefore, the convolution in the frequency domain implies the input signal to be treated as if it were a closed loop or ring. As an example, consider that we want to low-pass filter a discrete signal h_i by using a Gaussian as the filter function. As we have already seen, this corresponds to convolving the signal with the inverse Fourier transform of the Gaussian, which is also a Gaussian whose discrete and time-limited version is henceforth represented as g_i . The convolution procedure involves multiplying the signal h_i by time shifted versions of the function g_i —recall that since the Gaussian is even we have $g_{-i} = g_i$. Because in the DFT the signals are always periodical, when such time-displaced Gaussian functions are near the period extremities, they tend to wrap over to the other side of the signal, which is known as *circular convolution*. In such a way, the signal extremities interact with one another. While this is sometimes exactly what is required from the convolution (such as in the case of closed contours in Section 5.2.1), the circular convolution implemented by the DFT will not generally produce the same results as those obtained in the time domain by using the algorithms described in Section 2.5.2. Fortunately, the noncircular convolution can be easily calculated by the DFT simply by padding the functions with N zeros between each subsequent period of the signals being convolved (for additional information, see [Castleman, 1996]). The same approach can be applied to obtain noncircular correlation.

An interesting and relevant situation where the DFT is used to calculate the (generally noncircular) correlation occurs in statistics. We have seen in Section 2.6.5 that the cross-correlation and autocorrelation of stochastic signals can provide important information about the respective signals (or processes). Indeed, observe

that the cross- and autocorrelation can be performed in the Fourier domain by using the correlation theorem (Section 2.7.3). As a matter of fact, the autocorrelation of a signal can be obtained as the inverse Fourier transform of its power spectrum. This possibility is illustrated for the case of images in Section 2.6.5. Additional information about the use of the Fourier transform in statistics can be obtained in [Papoulis, 1962; Therrien, 1992].

To probe further: *Fourier Analysis*

The importance of Fourier analysis as a theoretical and applied tool has been fully substantiated by an ever growing number of related references. Nice coverages of Fourier series, including many examples, can be found in [Tolstov, 1976] and [Kreyszig, 1993]. Interesting introductions to the Fourier transform, including applications, are provided by [James, 1995] and [Sneddon, 1995], and [Spiegel, 1995] includes several exercises (many of which are solved) on Fourier analysis. Good references on Fourier analysis from the perspective of signal processing include [Lynn, 1984; Oppenheim & Schaffer, 1975; Papoulis, 1984] and [Burrus et al., 1994] provides many computer-based exercises on signal processing, including discrete Fourier transforms and fast Fourier transforms. Additional hands-on approaches to digital signal processing include [Ingle & Proakis, 1997; Kamen & Heck, 1997]. The classical reference on the fast Fourier transform is [Brigham, 1988]. A good reference on Fourier analysis from the perspective of linear systems can be found in [Frederick & Carlson, 1971], and [Körner, 1996] includes a series of more advanced topics on Fourier analysis. References on the application of Fourier analysis to imaging and computer vision include [Castleman, 1996; Gonzalez & Woods, 1993; Schalkoff, 1989].

2.8 Graphs and Complex Networks

Graphs¹ are discrete data structures involving nodes (or vertices) and links (or edges), so that emphasis is placed on the *connectivity*. Despite their intrinsic simplicity, graphs are particularly general in the sense that most other discrete data structures can be derived from them. For instance, lists, queues, trees, lattices, among many other structures, are but particular instances of graphs. As such, graphs are the natural choice for representing and modeling most (or even all) systems which involve components and relationships between these components. Figure 2.103 shows a simple graph composed of 8 nodes and 10 edges.

Examples of representations of real-world systems in terms of graphs include

¹Though graphs and networks are understood as different structure in graph theory, with networks being normally associated to flows, for historic reasons these two terms have been used indistinctively in the area of complex networks. In this book graphs and networks are treated as synonyms.