4.10 Using the above definition of the complex logarithm, show that for any complex spectrum $H(\omega)$, sampled at points ω and $\omega + \Delta\omega$, a reasonable approximation for the group delay is

$$\tau_g = \frac{2}{\Delta\omega} \operatorname{Im}\left[\frac{H(\omega+\Delta\omega) - H(\omega)}{H(\omega+\Delta\omega) + H(\omega)}\right]$$

4.11 Integrate τ_g of the single-zero, single-pole allpass filter over the Nyquist interval and find the filter's average group delay when it operates on 1000 Hz data. Interpret your results.

4.12 Show that the two couplets

$$F_1 = (Z - \sqrt{2}e^{-i3\pi/4})$$

and

$$F_2 = \sqrt{2}(Z - e^{-i3\pi/4}/\sqrt{2})$$

have the same magnitude spectrum, that F_1 is minimum phase, and that F_2 is maximum phase. Plot the zeros in the Z plane and sketch the phase lags to show that the minimum phase couplet does not have less phase delay at all frequencies.

4.13 What is the nature of the pole and zero locations of a causal digital filter?

4.14 Discuss the inverse of the allpass filter.

4.15 Show that $\ln y(Z) = \ln|y(Z)| + i\phi(Z)$.

5 The Discrete Fourier Transform

We have spent the last two chapters studying the frequency response of various operators. In each case, we performed an analytic calculation that resulted in a closed-form expression for $H(\omega)$. Traditionally, closed-form solutions—for any mathematical problem—have been prized for their intrinsic elegance and beauty. Indeed, they are a satisfying outcome for any problem. But frequently in using such solutions, we like to make graphs. In making a graph, we evaluate some function at selected points, either by hand or with the aid of a computer. If a computer is used, commonly we would evaluate the function at equally spaced points sufficiently close together to give a complete picture of the solution.

In this chapter, we apply this idea of evaluating, or sampling, to the frequency response of LSI systems and operators. This frequency sampling of $H(\omega)$ will lead us to the discrete Fourier transform, a powerful mathematical tool in digital signal processing.

Sampling the System Response in the Frequency Domain

Because sinusoids are eigenfunctions of discrete LSI systems, we have found their eigenvalues, the spectrum, extremely useful in thinking about the behavior of digital operators. In Chapter 3, we showed that the spectral response of any LSI operator can be computed from its impulse response function, h_κ, from

$$H(\omega) = \sum_{\kappa=0}^{N-1} h_\kappa e^{-i\omega\kappa} \tag{5.1}$$

Two important basic properties of $H(\omega)$ are (1) that it is a continuous function of ω, and (2) that it is periodic in ω with a period of 2π. We want to sample $H(\omega)$ at equally spaced points in frequency:

$$\omega = 2\pi\nu/M \qquad \nu = 0, 1, 2, \ldots, M-1$$

where we have limited the maximum frequency to $\omega = 2\pi$ because no new information is gained above that point because of the periodicity of $H(\omega)$. This sampling gives the M values of H:

$$H_\nu = \sum_{\kappa=0}^{N-1} h_\kappa e^{-i2\pi\nu\kappa/M} \qquad \nu = 0, 1, 2, \ldots, M-1$$

representing a transformation of N numbers, h_κ, to M other numbers, H_ν. Naturally, we would expect that a more powerful picture would emerge if this transformation could be easily inverted; that is if the h_κ could be found from the H_ν. With this objective in mind, we limit the number of frequency points M to equal N, the number of points in time. Although this restriction is not absolutely necessary, we do expect it to lead to a convenient inversion of the resulting N equations and N unknowns. Indeed, the inverse turns out to be quite simple: taking

$$H_\nu = \sum_{\kappa=0}^{N-1} h_\kappa e^{-i2\pi\nu\kappa/N} \qquad \nu = 0, 1, 2, \ldots, N-1$$

and multiplying both sides by $\exp(i2\pi\nu\kappa'/N)$ and summing on ν gives

$$\sum_{\nu=0}^{N-1} H_\nu e^{i2\pi\nu\kappa'/N} = \sum_{\kappa=0}^{N-1} \sum_{\nu=0}^{N-1} h_\kappa e^{i2\pi\nu(\kappa' - \kappa)/N} \tag{5.2}$$

Next, we introduce a fundamental relation for discrete exponential functions:

$$\sum_{\nu=0}^{N-1} e^{i2\pi\nu(\kappa' - \kappa)/N} = N\delta_{\kappa'\kappa} \tag{5.3}$$

where δ is the Kronecker delta symbol, equal to one if $\kappa' = \kappa$ and zero otherwise. This *orthogonality relation*, as Eq. (5.3) is sometimes called, is easily verified by thinking of the summation as vector addition in the complex plane: for all cases where $\kappa' \neq \kappa$, the vector diagram is a closed polygon whose resultant is zero. If $\kappa = \kappa'$, the segments of the polygon form a straight line on the real axis composed of N vectors of unit length.

Returning to the evaluation of Eq. (5.2), this orthogonality property of discrete exponentials gives us

$$\sum_{\nu=0}^{N-1} H_\nu e^{i2\pi\nu\kappa'/N} = \sum_{\kappa=0}^{N-1} h_\kappa N\delta_{\kappa'\kappa} = Nh_{\kappa'}$$

or, dropping the prime on κ,

$$h_\kappa = \frac{1}{N} \sum_{\nu=0}^{N-1} H_\nu e^{i2\pi\nu\kappa/N}$$

which is the desired inverse transformation from the H_ν to the h_κ. This pair of equations, which we rewrite for reference,

DFT
$$H_\nu = \sum_{\kappa=0}^{N-1} h_\kappa e^{-i2\pi\nu\kappa/N} \tag{5.4a}$$

IDFT
$$h_\kappa = \frac{1}{N} \sum_{\nu=0}^{N-1} H_\nu e^{i2\pi\nu\kappa/N} \tag{5.4b}$$

is collectively called the *discrete Fourier transform*. Sometimes, one is more specific and calls Eq. (5.4a) the discrete Fourier transform (the DFT), and Eq. (5.4b) is called the inverse discrete Fourier transform (the IDFT). Our discussion was motivated by thinking of the h_κ as the real impulse response of a digital LSI system and then the H_ν are sampled complex values of its frequency response. As a mathematical entity, the DFT can be thought of as a more general mapping of N complex numbers into N other complex numbers. We hasten to issue an alert: DFT definitions are not completely standard and you may find some that differ from Eqs. (5.4). In particular, the factor of N can appear in either equation (or even be symmetrically disposed by a factor $\sqrt{N}$), and the forward and inverse definitions can be reversed. This lack of standard definitions stems from an important property of the DFT—the duality of Eqs. (5.4)—where time and frequency play indistinguishable mathematical roles. Any property connecting time to frequency will have a corresponding statement connecting frequency to time.

The importance of the DFT to digital signal processing stems from two properties: first, it has the obvious relationship to the frequency domain; and second, the transformation has a high degree of symmetry, permitting a fast computing scheme called the *fast Fourier transform* or FFT. There are many ways of designing FFT algorithms, all with different advantages and limitations. Details can be found in many references, such as Elliott (1988), Press *et al.* (1986), IEEE (1979), and Rabiner and Gold (1975). But, you need not write your own FFT program. They are readily available in most scientific computing subroutine packages, they are even available in firmware on some microcomputers, and there are FFTs available on chips. Many FFT algorithms require data whose length is a power of two, but we will see that this is hardly ever a limitation. Problems 5.14, 5.15, and 5.16 discuss the computer computation of DFTs, both by direct calculation and by an FFT algorithm. To take advantage of these fast computing algorithms, we do need to fully understand the properties of the DFT and its use in performing LSI operations in the frequency domain.

Properties of the DFT

The DFT is a restricted version of the Z transform; it is the Z transform sampled at equally spaced points around the unit circle. Consequently, most of the important properties of the DFT will be familiar to us from our Z transform experience.

Perhaps the most basic property of the DFT is its periodicity. The periodicity of the spectrum of sampled time domain data has been a central point in our discussion of digital signal processing. But now, sampling in the frequency domain by the DFT has forced periodicity in the time domain:

$$h_{\kappa+N} = \frac{1}{N}\sum H_\nu e^{i2\pi\nu(\kappa+N)/N} = \frac{1}{N}\sum H_\nu e^{i2\pi\nu\kappa/N}$$

$$h_{\kappa+N} = h_\kappa \qquad \text{because} \qquad e^{i2\pi\nu} = 1$$

This periodicity in both domains can be represented in two ways, as shown in Fig. 5.1. Thinking of the DFT points as lying on a circle is frequently helpful when thinking of the properties of the transformation.

But what about this, perhaps unexpected, perhaps even unwanted, periodicity imposed on the time domain data? There are some applications, occurring relatively rarely in digital signal processing problems, in which we do indeed have repetitive time domain data. However, in the majority of applications, the time domain data are not periodic, leading us to an intriguing dilemma that strikes at the very heart of digital signal processing: in order to justify digitizing data without aliasing, its spectrum must be band limited. If the data is band limited in frequency, the data must be either periodic or infinitely long; in practice, neither is ever exactly true. In subsequent chapters, we will explore the consequences of this inconsistency between reality and Fourier theory in more detail. But beware, it will forever haunt us. For now we will present less subtle properties of the DFT.

For starters, we note that because of the periodicity of the DFT, the summation in Eqs. (5.4) can start at any point on the circle and continue through one period. For example, it is common to write these equa-

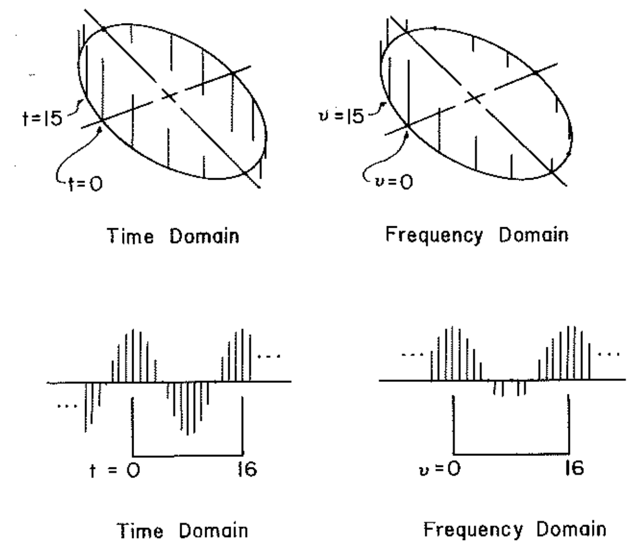


Figure 5.1 Showing the repetitive nature of the DFT data in both domains. At the top, the periodicity is represented by arranging the data on circles. At the bottom, the periodicity is shown by implying infinitely long periodic sequences. Two sets of figures are actually required in each domain: one for the real part of the data and one for the imaginary part.

tions as

$$H_\nu = \sum_{-N/2}^{N/2-1} h_\kappa e^{-i2\pi\nu\kappa/N} \tag{5.5a}$$

$$h_\kappa = \frac{1}{N}\sum_{-N/2}^{N/2-1} H_\nu e^{i2\pi\nu\kappa/N} \tag{5.5b}$$

so that the indices run from $-N/2$ through zero to $N/2-1$. For lack of better terminology, we will call Eqs. (5.4) the one-sided format and Eqs. (5.5) the centered format. The only difference is where one starts counting elements on the DFT circle. For example, the one-sided format gives

$$\mathrm{DFT}(1,0,0,1) = (2, 1+i, 0, 1-i) \tag{5.6}$$

where the centered format of the same sequence would read

$$\mathrm{DFT}(0,1,1,0) = (0, 1-i, 2, 1+i) \tag{5.7}$$

Since the one-sided format uses only positive indices, it is frequently more convenient for computer implementation, whereas the centered version is more useful for discussing the symmetry properties that we take up next.

One may think it strange that the DFT transforms N real numbers h into $2N$ numbers, resulting from the real and imaginary parts of the I terms of H_ν. This situation is resolved by the symmetry of the transforma tion that results when h_κ is real. In the more general case, h_κ can b complex; the DFT then transforms N complex h_κ into N complex H_ν. T investigate the symmetry properties of the DFT, we take the comple conjugate of Eq. (5.5a) to get

$$H_\nu^* = \sum_{-N/2}^{N/2-1} h_\kappa^* e^{i2\pi\nu\kappa/N}$$

and then relabel ν to $-\nu$:

$$H_{-\nu}^* = \sum_{-N/2}^{N/2-1} h_\kappa^* e^{-i2\pi\nu\kappa/N}$$

The previous two equations show that

$$H_{-\nu}^* \leftrightarrow h_\kappa^* \qquad (5.8a$$

$$H_\nu^* \leftrightarrow h_{-\kappa}^* \qquad (5.8b$$

Also, it is easy to see that

$$H_{-\nu} \leftrightarrow h_{-\kappa} \qquad (5.8c$$

where the double arrow indicates taking the DFT when going from κ, i time, to ν, in frequency, or taking the IDFT in the opposite direction. Fo the special case when h_κ is real, Eqs. (5.8) say that

$$H_{-\nu}^* = H_\nu \qquad (5.9$$

a property of H that is sometimes called *Hermitian*; it means that H has a symmetric real part and an antisymmetric imaginary part. Thus, H only has N independent components. If h is purely imaginary, Eqs. (5.8) say that

$$H_{-\nu}^* = -H_\nu \qquad (5.10$$

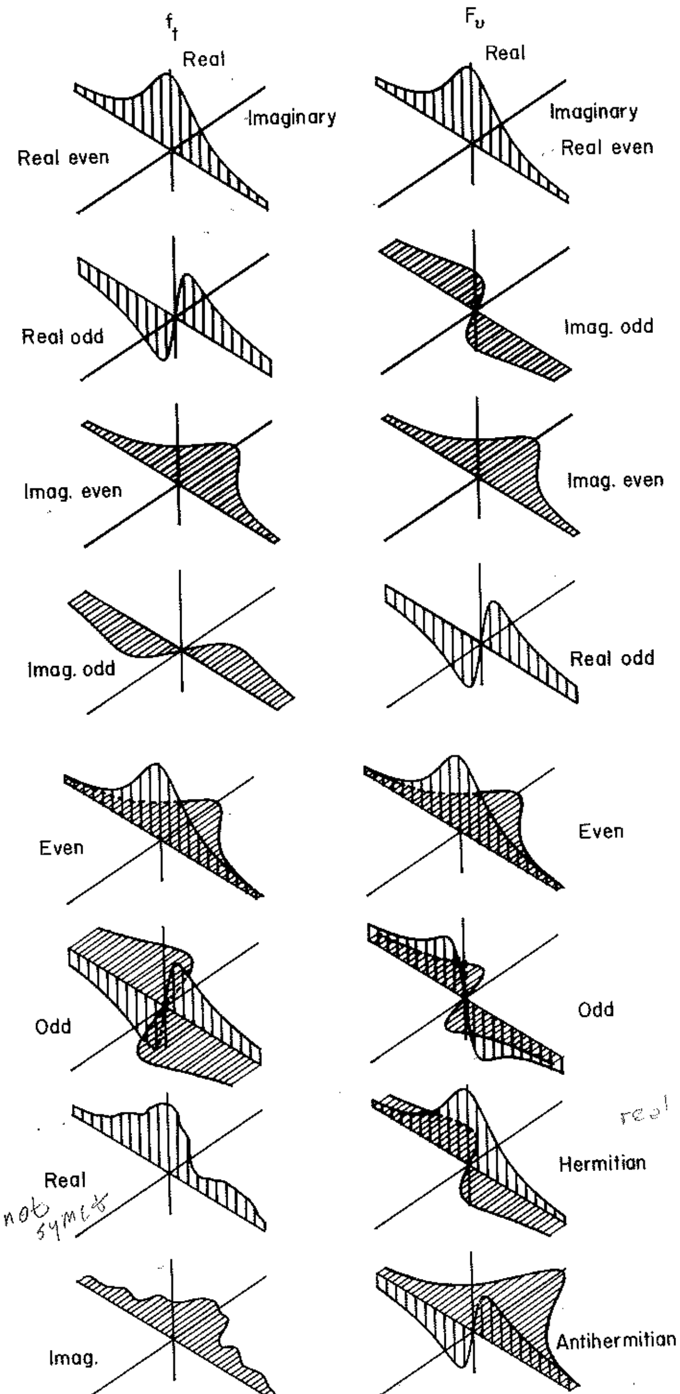
a property of H that is sometimes called *anti-Hermitian*; it means that H has an antisymmetric real part and a symmetric imaginary part.

Further special symmetrics result if h_κ or H_ν is an even or odd function of their index. For example, if h_κ is real and even about $\kappa = 0$, in the case of a zero-phase real operator, then Eq. (5.8b) says that

$$H_\nu^* \leftrightarrow h_{-\kappa}^* = h_{-\kappa} = h_\kappa$$

or

$$H_\nu^* = H_\nu$$



ure 5.2 Symmetry properties of the DFT using the centered format whereby the indices t l ν run from $-N/2$ to $N/2-1$. For even N, the center of symmetry occurs at t and $\nu = 0$.

ıe ıe a) :d) r s e d e r y r

and

$$H_\nu = H_{-\nu}$$

That is, H is purely real and even about $\nu = 0$. A useful summary of all such symmetry properties, which are likewise easily verified, is shown in Fig. 5.2.

Special Values of the DFT

Next, we consider the transformation equations, Eqs. (5.5), for some special values of their indices. For $\nu = 0$, Eq. (5.5a) reads

$$H_0 = \sum_{-N/2}^{N/2-1} h_\kappa \tag{5.11}$$

The zero-frequency value of the spectrum is the sum of the time terms. When divided by N, this sum becomes the average value of the time sequence—the dc value in electrical circuit terms. Similarly, for $\kappa = 0$, Eq. (5.5b) reads

$$h_0 = (1/N) \sum_{-N/2}^{N/2-1} H_\nu \tag{5.12}$$

sometimes simply called the first value of the time sequence.

Another special value occurs one-half way around the DFT circle:

$$H_{N/2} = \sum h_\kappa e^{-i2\pi N\kappa/2N} = \sum e^{-i\pi\kappa} h_\kappa$$

or

$$H_{N/2} = \sum_{-N/2}^{N/2-1} (-1)^\kappa h_\kappa \tag{5.13}$$

which is called the Nyquist value of H. Likewise, for h,

$$h_{N/2} = \frac{1}{N} \sum H_\nu e^{i2\pi\nu N/2N} = \frac{1}{N} \sum e^{i\pi\nu} H_\nu$$

$$h_{N/2} = \frac{1}{N} \sum_{-N/2}^{N/2-1} (-1)^\nu H_\nu \tag{5.14}$$

Another important relation, called *Parseval's* theorem, readily follows from the definition of the DFT:

$$N \sum_{-N/2}^{N/2-1} |h_\kappa|^2 = \sum_{-N/2}^{N/2-1} |H_\nu|^2 \tag{5.15}$$

The sum of the squares of the amplitude in a signal is sometimes called the total energy of the signal in analogy with an electrical potential across a load.

These symmetry properties and special values that we have discussed in Eqs. (5.8) through Eq. (5.15) are of great practical value in checking and using DFT calculations. Note that many of these properties can be easily demonstrated for the DFT shown in Eq. (5.7).

The Phase-Shift Theorem

The value of the frequency domain is due in large part to providing an alternate viewpoint to the time domain. The usefulness of this alternate view hinges critically on developing connecting relationships between the two domains. We have just discussed relationships between numbers calculated in each domain. Next, we wish to explore relationships between operations in both domains. These operations, intimately associated with the DFT's granddaddy—the Z transform—will play a central role in using the DFT in digital signal processing.

The first of these operations is directly associated with the delay operator Z^τ. To delay a sequence τ units in time, its Z transform is multiplied by this delay operator. The DFT of a delayed sequence is similarly related to the DFT of the original sequence by

$$\mathrm{DFT}(h_{\kappa-\tau}) = e^{-i2\pi\nu\tau/N}\,\mathrm{DFT}(h_\kappa) \tag{5.16}$$

which is a relation, easily verified by direct substitution, called the *phase shift theorem*; a shift in time is a phase factor in the frequency domain. As an example, we consider the 4-long real symmetric sequence $(0,0,1,0)$; its magnitude spectrum is all one's $(1,1,1,1)$ and its phase spectrum is zero. We can write phase of the DFT $(0,0,1,0) = (0,0,0,0)$. If the one is shifted through the sequence by progressive delays, the spectrum is multiplied by

$$e^{-i\,\pi\tau/2}$$

which leaves the magnitude spectrum unchanged and introduces a linear phase with a negative slope of 90° per unit of time shift. We can write

phase of the DFT $(0,0,0,1) = (180°, 90°, 0, -90°)$
phase of the DFT $(1,0,0,0) = (360°, 180°, 0, -180°)$
phase of the DFT $(0,1,0,0) = (540°, 270°, 0, -270°)$

This phase-shift theorem has many important applications. For example, the one shifting through the 4-long sequence might represent wave propagation to the right in time. During propagation, the wave's magnitude

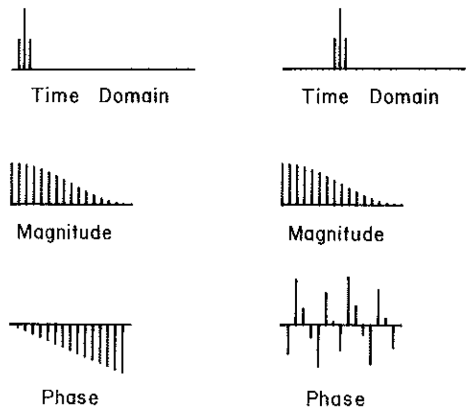


Figure 5.3 Two examples of the phase-shift theorem. At the top left, a time domain signal is shifted one unit from the symmetric position. Its phase spectrum, at the bottom left, shows a phase that varies linearly from 0 to $-\pi$ over one-half the Nyquist interval. At the top right, the same signal is shown shifted 9 units from the symmetrical position. The magnitude spectrum is unchanged; its phase spectrum varies so rapidly that unwrapping, to show the linear phase dependency, is difficult. Only one-half of the frequency spectrum of the 32-point DFT is shown because the real time domain signal produces a symmetric magnitude spectrum and an antisymmetric phase spectrum.

spectrum remains unchanged while its phase spectrum undergoes clockwise rotation. The calculation of DFT phases can run into two difficulties, particularly on natural data, that frequently make the phase spectrum look like noisy nonsense. First, the phase may be the computed modulo 2π, or π, so that phases that are changing rapidly are difficult to unwrap. Second, the phase is usually calculated from the arc tangent of the ratio of the imaginary part of the spectrum to the real part. If the real part meanders about zero, wild fluctuations result in the phase angle. Examples are shown in Fig. 5.3.

The Convolution Theorem

Like the phase-shift theorem, the convolution theorem reflects a fundamental property of the Z transform. Convolution in time corresponds to multiplying Z transforms, and therefore, it also corresponds to multiplying DFT spectra. We know that the most general operation an LSI system can perform on a time sequence is a convolution; clearly, the convolution theorem promises to be a major link between the two domains.

We prove the DFT convolution theorem by multiplying two spectra and transforming back to time. Capital letters will be used for frequency

domain data and lowercase letters for time domain data:

$$F_\nu = \sum_\kappa f_\kappa e^{-i2\pi\kappa\nu/N}$$

$$G_\nu = \sum_{\kappa'} g_{\kappa'} e^{-i2\pi\kappa'\nu/N}$$

The product of the spectra becomes

$$F_\nu G_\nu = \sum_\kappa \sum_{\kappa'} f_\kappa g_{\kappa'} e^{-i2\pi\nu(\kappa'+\kappa)/N}$$

and taking the IDFT of the product gives

$$\begin{aligned}\mathrm{IDFT}(F_\nu G_\nu)_\tau &= \frac{1}{N}\sum_\nu e^{i2\pi\nu\tau/N}\sum_\kappa\sum_{\kappa'} f_\kappa g_{\kappa'} e^{-i2\pi\nu(\kappa'+\kappa)/N}\\ &= \frac{1}{N}\sum_\kappa\sum_{\kappa'} f_\kappa g_{\kappa'} \sum_\nu e^{-i2\pi\nu(\kappa'+\kappa-\tau)/N}\\ &= \frac{1}{N}\sum_\kappa\sum_{\kappa'} f_\kappa g_{\kappa'} N\delta_{\kappa',\tau-\kappa}\end{aligned}$$

where we have used the orthogonality property of Eq. (5.3) to evaluate the sum over ν. Finally, the δ symbol only contributes the $\tau-\kappa$ term from the sum over τ':

$$\mathrm{IDFT}(F_\nu G_\nu)_\tau = \sum_\kappa f_\kappa g_{\tau-\kappa} \tag{5.17a}$$

Likewise, if we started by multiplying two time sequences and transformed them into the frequency domain, we would see that

$$\mathrm{DFT}(f_\tau g_\tau)_n = \frac{1}{N}\sum_\nu F_\nu G_{n-\nu} \tag{5.17b}$$

so that convolving in one domain corresponds to multiplication in the other domain.

A good reason for taking time to prove the DFT convolution theorem is that inspection of the proof reveals that all the summations used are around the DFT circle; the convolution sums in Eqs. (5.17) are called *circular convolution*.

The distinction between circular convolution and *linear convolution*, the type that we have considered until now, is an important detail of DFT data processing. Perhaps the best way to elaborate on the difference is by means of an example. We select $(1,3,0,2)*(1,0,2,2)$. Under circular convolution,

f g

$$(1,3,0,2)*(1,0,2,2) = (7,6,10,7)$$

while under linear convolution,

$$(1,3,0,2)*(1,0,2,2)=(1,3,2,10,6,4,4)$$

The results are, of course, completely different—they even have different lengths. Hand calculations of short circular convolutions are best performed by repeating one factor and then shifting from the zero lag position so as to maintain complete overlap; schematically, the above case would look like

$$\begin{array}{c} (\ 2\ 0\ 3\ 1\ ) \rightarrow \\ \cdots\ \ 1\ 0\ 2\ 2\ 1\ 0\ 2\ \cdots \end{array}$$

Linear convolution can be included within the circular convolution framework by appending zeros to eliminate the overlap between cycles. A schematic representing this looks like

$$\begin{array}{c} (\ 0\ 0\ 0\ 0\ 2\ 0\ 3\ 1\ ) \rightarrow \\ \cdots\ 1\ 0\ 2\ 2\ 0\ 0\ 0\ 0\ 1\ 0\ 2\ 2\ 0\ 0\ 0\ 0\ \cdots \end{array}$$

It follows that the advantage of high-speed FFT algorithms can be used for linear convolution: the two factors are appended with zeros so as to be of equal length, at least twice as long as the longer of the two. The product of the DFTs of each sequence is then transformed back into the time domain.

Symmetry Properties

Time $\longleftrightarrow$	Frequency
f_t^*	$F_{-\nu}^*$
f_{-t}^*	F_ν^*
f_{-t}	$F_{-\nu}$

Convolution Relations

Time $\longleftrightarrow$	Frequency
$f*g$	$F_\nu G_\nu$
$f*g_{-t}$	$F_\nu G_{-\nu}$
$f_{-t}*g_{-t}$	$F_{-\nu}G_{-\nu}$
$f_t*g_{-t}^*$	$F_\nu G_\nu^*$
$f_t*g_t^*$	$F_\nu G_{-\nu}^*$
$f_{-t}*g_{-t}^*$	$F_{-\nu}G_\nu^*$
$f_{-t}*g_t^*$	$F_{-\nu}G_{-\nu}^*$
$f_{-t}^**g_{-t}^*$	$F_\nu^*G_\nu^*$
$f_{-t}^**g_t^*$	$F_\nu^*G_{-\nu}^*$
$f_t^**g_t^*$	$F_{-\nu}^*G_{-\nu}^*$

Figure 5.4 Symmetry properties of the DFT and their application to 10 forms of the convolution theorem.

For sufficiently long sequences, the speed of the FFT more than compensates for what seems, at first sight, like extra calculations.

Clearly, the convolution theorem will play a central role in the remainder of our discussions. It is not surprising then, that application of the symmetry properties of the DFT discussed earlier in the chapter leads to some equally interesting forms of the convolution theorem. Figure 5.4 reviews these symmetry properties and lists 10 forms of the convolution theorem resulting from them. In the next section, we study one particular form.

Cross-Correlation and Autocorrelation

A version of the fourth form of the convolution theorem listed in Fig. 5.4, obtained by a simple relabeling, is

$$\mathrm{DFT}(f^*(-)*g)=F^*G \tag{5.18}$$

It has particular significance. First, let us write out the convolution indicated on the left-hand side; the τth element is

$$f^*(-)*g=\sum_t f^*_{-(\tau-t)}g_t=\sum_t f^*_{(t-\tau)}g_t$$

We define the particular convolution sum to be the complex *cross-correlation* of f^* with g and use the notation

$$f^*\otimes g=\sum_t f^*_{t-\tau}g_t \tag{5.19a}$$

By a simple change of variables

$$t'=t-\tau$$

we can see, after dropping the prime notation, that

$$f^*\otimes g=\sum_t f_t^*g_{t+\tau} \tag{5.19b}$$

Clearly, the cross-correlation is similar to convolution; the difference is that the order is not reversed in the cross-correlation computation as it is for convolution. For example, to compute the cross-correlation between the two 4-long sequences of the preceding section,

$$(1,3,0,2)\otimes(1,0,2,2)$$

the computational schematic looks like

$$\begin{array}{c} (\ 1\ 3\ 0\ 2\ ) \\ \cdots\ 1\ 0\ 2\ 2\ 1\ 0\ 2\ 2\ \cdots \end{array}$$

where neither sequence is reversed. The result is

$$(1,3,0,2)\otimes(1,0,2,2)=(5,8,8,9)$$

Also

$$(1,0,2,2)\otimes(1,3,0,2)=(5,9,8,8)$$

Comparison of these two results show that factors of the cross-correlation do not commute, but they are time reversed versions of one another. (Remember that these results are on the DFT circle; time reversal implies a cyclic progression in the opposite sense.) That is, if

$$f\otimes g=R_t$$

then

$$g\otimes f=R_{-t}$$

An important special case of cross-correlation occurs when $g=f$. Then, Eq. (5.19a) becomes the *autocorrelation* of f, which reads

$$f^*\otimes f=\sum_t f^*_{t-\tau}f_t \tag{5.20a}$$

and

$$f^*\otimes f=\sum_t f^*_t f_{t+\tau} \tag{5.20b}$$

showing that the autocorrelation is symmetric in its lag variable τ. Furthermore, Eq. 5.18 now says

$$\mathrm{DFT}(f^*\otimes f)=F^*F=|F|^2 \tag{5.21}$$

or, in words, the DFT of the autocorrelation is the DFT power spectrum. It is possible to prove that the autocorrelation is a maximum at zero lag. For this reason, sometimes it is normalized to unity at $\tau=0$ by writing

$$f^*\otimes f=\sum f^*_t f_{t+\tau}/\sum f_t f_t$$

The DFT power spectrum of a sequence and its autocorrelation contain the same information. In the frequency domain, it is clear that the sequence's phase information is lost. All sequences with the same power spectrum have the same autocorrelation, even though the sequences differ. For example, $(4,0,-1)$ and $(2,3,-2)$ have the same autocorrelation. Autocorrelation will play an important role in later chapters in our discussions of inverse filtering, spectral factorization, and power spectrum estimation.

Both cross-correlation and autocorrelation are easily described in the Z plane. Since cross-correlation differs from convolution in its time reversal,

it is clear that the complex cross-correlation between two sequences, $A(Z)$ and $B(Z)$, is given by

$$A^*\otimes B=A^*(1/Z)B(Z)$$

and the autocorrelation of a sequence A is

$$A^*\otimes A=A^*(1/Z)A(Z) \tag{5.22}$$

which is the power spectrum in the Z plane introduced in Eq. (4.7).

In this chapter, we have found that the concept of the DFT, a discrete time, discrete frequency Fourier transform, arises quite naturally from the desire to sample the continuous spectrum of a discrete time sequence. We have only presented the fundamental properties of the DFT, the mechanics we might say. Before we can profitably turn the wheels of this DFT machinery and apply it to practical problems, we need a deeper insight into the subtleties required to relate DFT data to the real world. This relationship can only be brought to light within the conceptual framework of continuous time and continuous frequency. In the next chapter, we will develop the appropriate Fourier transform for this case and then use it to illustrate the role of the DFT in data processing.

Problems

5.1 Clearly, the DFT of Eq. (5.4a) is not a shift invariant operation on the h_κ. Show, however, that for $N=4$ Eq. (5.4a) can be written as the matrix operation

$$\begin{pmatrix} \\ H \\ \\ \end{pmatrix}=\begin{pmatrix}1 & 1 & 1 & 1\\ 1 & W & W^2 & W^3\\ 1 & W^2 & W^4 & W^6\\ 1 & W^3 & W^6 & W^9\end{pmatrix}\begin{pmatrix} \\ h \\ \\ \end{pmatrix}$$

where $W=\exp(-i2\pi/\mathrm{N})$. A matrix whose elements satisfy $A_{\nu\kappa}=A^{(\nu-1)(\kappa-1)}$ is called a Van Der Monde matrix. Show that the above matrix satisfies this definition. Show that the inverse to the above matrix just has each element inverted, that is the inverse equation is

$$\begin{pmatrix} \\ h \\ \\ \end{pmatrix}=\frac{1}{N}\begin{pmatrix}1 & 1 & 1 & 1\\ 1 & W^{-1} & W^{-2} & W^{-3}\\ 1 & W^{-2} & W^{-4} & W^{-6}\\ 1 & W^{-3} & W^{-6} & W^{-9}\end{pmatrix}\begin{pmatrix} \\ H \\ \\ \end{pmatrix}$$

5.2 It is sometimes useful to have a stock list of DFTs for reference. Show that the following are DFT transforms (we are using the centered format):

$$\mathrm{DFT}(0,0,1,0) = (1,1,1,1)$$
$$\mathrm{DFT}(1,1,1,1) = (0,0,4,0)$$
$$\mathrm{DFT}(0,0,1,1) = (0,1+i,2,1-i)$$
$$\mathrm{DFT}(0,-1,1,1) = (1,1+2i,1,1-2i)$$
$$\mathrm{DFT}(0,1,1,0) = (0,1-i,2,1+i)$$
$$\mathrm{DFT}(0,-1,0,1) = (0,2i,0,-2i)$$
$$\mathrm{DFT}(0,-i,0,i) = (0,-2,0,2)$$

5.3 Identify the dc and the Nyquist values of the DFTs in Problem 5.2. Show that they satisfy Eqs. (5.12) and (5.13). Show that the DFTs satisfy the symmetry properties for even and odd functions.

5.4 The third and fifth sequences listed in Problem 5.2 differ by a shift of one unit. Convert the DFT spectra shown into magnitude and phase, showing that these spectra differ only by an overall phase factor.

5.5 Multiply the spectra of examples three and five from Problem 5.2. Compute the IDFT of the resulting product and show that it is equal to $(0,0,1,1) * (0,1,1,0)$ as an example of the convolution theorem.

5.6 Prove Parseval's theorem, Eq. (5.15), and verify that the examples of Problem 5.2 satisfy it.

5.7 Our definition of the cross-correlation started with the convolution $f^*(-) * g$. Show that if you start with $f * g^*(-)$ and proceed in a similar fashion, the resulting cross-correlation is the time-reversed version of our definition.

5.8 Compute both the linear and the circular autocorrelation of the following sequences: $(2,5,2)$, $(4,4,1)$, and $(1,4,4)$.

5.9 Prove the phase-shift theorem of Eq. (5.16) by direct substitution into the DFT equations.

5.10 In the paragraph following Eqs. (5.4), we spoke of the duality of the DFT equations. Find the statement that is dual to the phase-shift theorem stated in Eq. (5.16).

5.11 Figure 5.4 shows 10 forms of the convolution theorem. Use these relationships to write down at least 5 similar expressions that relate autocorrelation in the time domain to multiplication in the frequency domain.

5.12 In Chapter 1, we noted that convolutions can be written as matrix operations. Show that for circular convolution the matrix must be a *circulant*, that is, a special Toeplitz structure where each row is a circular shift of the one above. As a consequence of this circulant structure, show that the eigenvalues of the matrix are given by the DFT of the first row. That is,

$$\mathbf{C}\mathbf{V}_i = \lambda_i \mathbf{V}_i$$

where $\mathbf{C}$ is the circulant matrix, $\mathbf{V}$ are the eigenvectors, and λ are the eigenvalues. How are the DFTs of the other rows related to the eigenvalues?

5.13 Discuss the time reversed version of a real minimum phase sequence within the DFT formalism.

5.14 It is important to realize that the DFT is a simple computation that may be done directly when the extra speed of the fast Fourier transform is not required. Use the direct calculation coded below to verify the transforms given in Problem 5.2. In this program, the complex input array is $X + iY$ and the output is $R + iI$. Of course, there is no limitation on their length N.

```
      REAL X(N), Y(N), R(N), I(N)
      DO 11 K = 1, N
      W = 2 * PI * (K-1)/N
      XO = X(1)    YO = Y(1)
      DO 9 J = 2, N
      P = W * (J-1)
      C = COS(P)    S = SIN(P)
      XO = XO + C * X(J) + S * Y(J)
    9 YO = YO + C * Y(J) - S * X(J)
      R(K) = XO
   11 I(K) = YO
```

5.15 The above program requires $2N^2$ evaluations of the sine and cosine functions while only N different values are required. A table lookup approach will solve this problem, and will be even more efficient when multiple DFT computations of the same length are needed because the table need only be computed once. Modify the above program to operate by table lookup.

5.16 The neat and efficient fast Fourier transform algorithm listed below (after Claerbout, 1976) uses complex arithmetic. Rewrite the program to use only real numbers for use on computers not permitting complex arithmetic. The complex array X is both the input and the output of length N, which must be a power of two. The flag F should be 1 for the forward transform and -1 for the inverse transform. Check the operation of your program against the transforms of Problem 5.2 and the results of the program in Problem 5.14.

```
      SUBROUTINE FFT (X, N, F)
      COMPLEX X(N), A, T, W
      J = 1
   Sc S = SQRT(N ** (F-1))
      DO 14 I = 1, N  Lx
      IF (I.GT.J) GOTO 9
 temp T = S * X(J)
      X(J) = S * X(I)
      X(I) = T
    9 M = N/2
   10 IF (J.LE.M) GOTO 14
      J = J - M
      M = M/2
      IF (M.GE.1) GOTO 10
   14 J = J + M
      L = 1
   16 D = 2 * L
      DO 23 M = 1, L
      A = -(0.0, 1.0) * PI * F * (M-1)/L
      W = CEXP(A)
      DO 23 I = M, N, D
      T = W * X(I + L)
      X(I + L) = X(I) - T
   23 X(I) = X(I) + T
      L = D
      IF (L.LT.N) GOTO 16
```

6 The Continuous Fourier Integral Transform

Digital signal processing has become extremely important in recent years because of digital electronics. In treating the processing of analog signals, which require any reasonable amount of computation, it is usually beneficial to digitize the signals and to use digital computers for their subsequent processing. The advantage results from both the extremely high computation speeds of modern digital computers and the flexibility afforded by computer programs that can be stored in software or firmware or hardwired into the design. Low-cost, large-scale integrated circuits, and more recently VLSI circuits, have made this approach beneficial even for devices limited to special-purpose computing applications or restricted by throwaway economics. This computational asset has been a major impetus for thinking of signals as discrete time sequences. An additional advantage of representing signals in discrete time has been their pleasant mathematical simplicity; continuous-time theory requires far more advanced mathematics than the algebra of polynomials, some simple trigonometric function theory, and the behavior of the geometrical series that we have employed. The digital revolution seduces us into viewing every situation in its terms; still, we are haunted by the concept of underlying continuous relationships. We need to know the effect of digitizing continuous-time signals and the essential difference between these digitized signals and those signals that are inherently digital from the start. Are continuous-time and discrete-time versions simply alternate models of the real physical world from which we are free to chose? Some say yes; yet, there are essential differences.