Chapter 3

The Discrete Fourier Transform

At the beginning of the 19th century, French polymath Jean-Baptiste Joseph Fourier turned his attention to the way in which heat propagates inside a solid object. In the resulting memoir he introduced an analytical tool that was destined to become so successful, both in mathematics and in applied engineering, as to acquire the eponymous name of *Fourier Transform*. Fourier's intuition was that many functions (and, remarkably, even discontinuous functions) could be represented as linear combinations of simple, harmonically-related sinusoidal components.

The systematization of the mathematical aspects of Fourier's theory took over a century to complete, and produced a remarkable body of work collectively known under the name of Harmonic Analysis. At the same time, the practical applicability of the Fourier transform gained immediate recognition across all scientific domains — the sole drawback being the need to carry out cumbersome numerical calculation in order to work out the necessary coefficients. It is no surprise, then, that Fourier analysis enjoyed a renewed, explosive success as soon as electronic computation became a commodity: the well-known Fast Fourier Transform algorithm (interestingly, originally sketched by Gauss in 1805) is arguably the fundamental ingredient at the heart of today's personal communication devices.

To understand why Fourier Analysis plays such a central role in so many disciplines, and in signal processing in particular, let's consider the physical processes behind most of the interesting phenomena that we want to model or describe. Signals are time-varying quantities: you can imagine, for instance, the air pressure level produced by the singing voice, the electrical activity of a beating heart or the daily level of the tide in Venice; in all cases, the variation of a signal over time implies that a transfer of energy is taking place somewhere. Now, a time-varying value whose trend is continually increasing over time is clearly a physical impossibility in the long run: either the system will reach a maximum level and stop, or something (such as a wire, a fuse or a combustion chamber) will overheat and break. Oscillations, on the other hand, are nature's and technology's way of keeping things indefinitely in motion without incurring a meltdown; from Maxwell's wave equation to the mechanics of the vocal cords, from the motion of an engine to the ebb and flow of the tide, oscillatory behavior is the universally recurring theme for sustained activity. Sinusoidal oscillations are the purest form of such a constrained motion and, details aside, Fourier's everlasting contribution was to show that one could express any reasonably well-behaved phenomenon as the combined output of a number of harmonically related sinusoidal sources. After Fourier, virtually every mathematical object of interest began a new life as a dual entity, existing not only in time but also in frequency; and the possibility of switching the analysis viewpoint from one domain to the other according to convenience has become completely natural in all forms of theoretical and applied science.

In this chapter we will describe some key properties and results of Fourier analysis as applied to discrete-time signals. We have already mentioned in the previous chapter that, by using a vector space framework for signal processing, the Fourier transform can be described as a change of basis. This guiding principle will prove extremely useful as we navigate the subtle differences that exists between the different flavors of the transform and as we interpret their properties.

3.1 Introduction to Fourier Analysis

A Fourier transform provides an alternative representation of the data contained in a signal. Given the N-point signal shown at the top of Figure 3.1, it is completely straightforward to express it as the weighed sum of the N "atomic" time signals shown in the middle panel (signals which, incidentally, we know to represent the canonical basis for the space of size-N vectors). While less evident, the same triangular signal can also be expressed exactly as the sum of the N discrete-time oscillatory components shown in the bottom panel of the figure; the plots show these oscillation (and the shape of the underlying continuous-time sinusoids for clarity), together the weight associated to each one.

In this and the next chapter we will address the question of whether *any* discrete-time signal can be expressed exactly as a sum of oscillatory components, and study the problem of finding the weights associated to each component. The resulting set of weights is called the *spectrum* of a signal and it is an alternative representation of the signal from the so-called *frequency domain*. We will also show that when a signal admits a frequency-domain representation, the latter is invertible and we can always return to the time domain without loss of information.

To achieve all this, we will derive a Fourier transform operator for each of the three basic classes of discrete-time signals we introduced in the first chapter, making full use of the vector space these signals live in; in detail, we will look at:

- the **Discrete Fourier Transform (DFT)**, which maps a length-N signal to a set of N discrete frequency components; the transform is a change of basis in the underlying finite-dimensional vector space $\mathbb{C}^N$ and, as such, it can be easily computed numerically using very efficient algorithms.

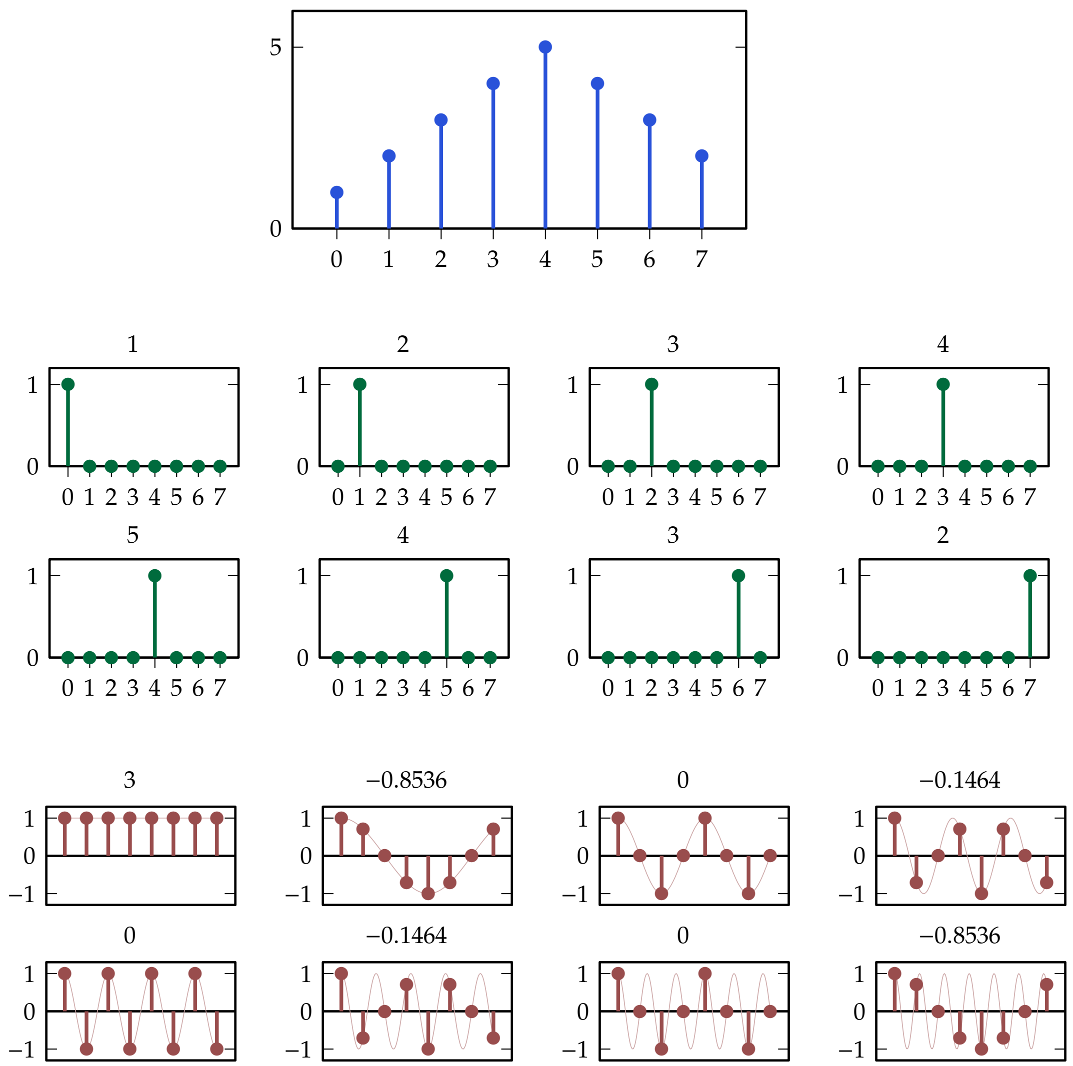

Figure 3.1: *Decomposition of a 8-point triangular signal (top panel) into atomic time units (middle panel) and atomic oscillatory units (bottom panel); the coefficient for each component is shown on top of the plot of each unit.*

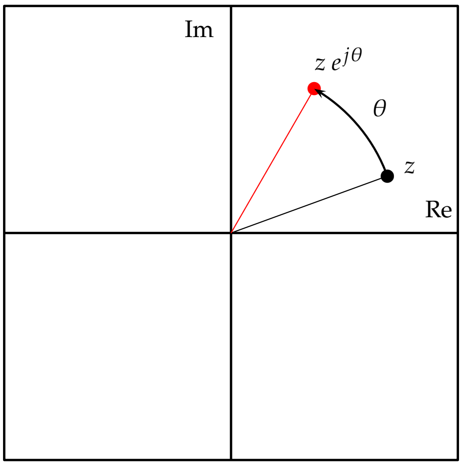

Figure 3.2: *Rotation of a point on the complex plane via multiplication by a phase factor $e^{j\theta}$.*

- the **Discrete Fourier Series (DFS)**, which maps an N-periodic sequence to a set of N discrete frequency components; the DFS is mathematically identical to the DFT, except that the periodicity of the signals is taken into account explicitly; the DFS is also a change of basis in $\mathbb{C}^N$.
- the **Discrete-Time Fourier Transform (DTFT)**, which maps an infinite sequence to a 2π-periodic function of a real-valued frequency variable; this transform can also be interpreted mathematically as a change of basis, as we will see in detail, connecting $\ell_2(\mathbb{Z})$, the space of finite-energy sequences, to $L_2([-\pi, \pi])$, the space of square-integrable functions over the $[-\pi, \pi]$ interval. The DTFT is a theoretical analysis tool that will be used to prove fundamental signal processing results that hold in the most general cases.

3.2 The Complex Exponential

Regardless of the underlying signal space, the Fourier transform decomposes a discrete-time signal into a superposition of suitably scaled discrete-time oscillatory components. The prototypical oscillation of choice (that is, the basic ingredient of all transforms) is the *discrete-time complex exponential*, namely a sequence $\mathbf{w}$ of the form

$$w[n] = A\,e^{j(\omega n + \phi)} \tag{3.1}$$

where $A \in \mathbb{R}$ is the amplitude, ϕ is the phase offset (in radians) and $\omega \in \mathbb{R}$ is the oscillation's frequency, also expressed in radians. Note that, although it is convenient to think of the index n as a measure of "time", such time is a-dimensional and therefore the units for the frequency are simply radians (and not, say, radians per second).

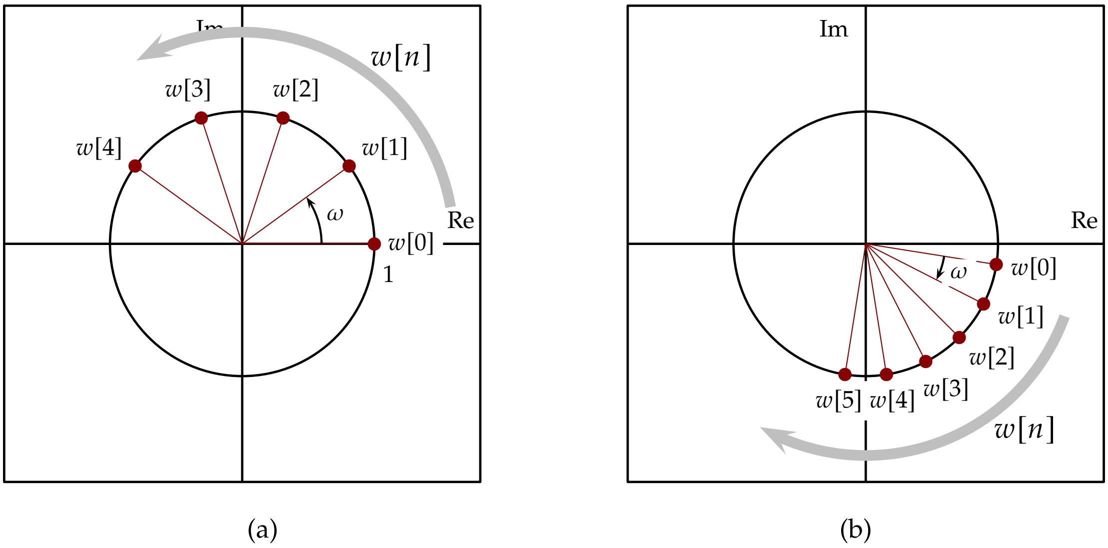

Figure 3.3: *Complex exponential sequences on the complex plane:*
(a) $w[n] = e^{j\omega n}$ with $\omega = 2\pi/10$ (positive frequency);
(b) $w[n] = e^{j\omega n + \theta}$ with $\omega = -2\pi/20$ and $\theta = 2\pi/40$ (negative frequency).

As we have already explored,simple complex algebra shows that

$$w[n+1] = w[n]\, e^{j\omega}; \tag{3.2}$$

that is, we can imagine $\mathbf{w}$ as the output of a "complex exponential generating machine" that, at each step, takes the previous sample and multiplies it by the pure phase factor $e^{j\omega}$, a complex number whose magnitude is equal to one. Multiplication of a complex point z by a phase factor $e^{j\theta}$ corresponds to rotating z around the origin by an angle θ as shown in Figure 3.2; the rotation is counterclockwise if θ is positive and clockwise if θ is negative. With this, the sequence $\mathbf{w}$ can be plotted on the complex plane as a series of points on a circle of radius $|A|$ centered on the origin; each point is at an angular distance of ω from the previous one. Two examples of complex exponential sequences are shown for a few values of n in Figure 3.3, using positive and negative frequencies and different phase offsets. As you can see, the complex exponential perfectly captures the concept of a point rotating in circles, i.e. the most fundamental type of oscillatory movement.

3.2.1 Why *complex-valued* oscillations?

The choice of a complex-valued signal as the prototypical oscillation may appear needlessly... *complex* at first and, in fact, the Fourier transform of real-valued sequences could be derived entirely in the real-valued domain using only standard trigonometric functions. There are however several major advantages in using complex exponentials.

The math is simpler: in a nutshell, by using complex exponentials trigonometry boils down to complex algebra. For instance, how many times have you struggled with the correct angle-sum formula? You remember that $\cos(\alpha + \beta)$ will be equal to a sum of products of sines and cosines but in what order and with what sign? Using complex algebra, however,

$$\begin{aligned}\cos(\alpha + \beta) &= \mathrm{Re}\{e^{j\alpha}\, e^{j\beta}\}\\ &= \mathrm{Re}\{(\cos\alpha + j\sin\alpha)(\cos\beta + j\sin\beta)\}\\ &= \cos\alpha\cos\beta - \sin\alpha\sin\beta.\end{aligned}$$

No need to remember formulas means fewer chances of making mistakes.

The notation is more compact: oscillatory signals originate in the circular movement of a rotating point and the position of the point always possesses two degrees of freedom. We can choose to encode this information using the point's real-valued vertical and horizontal coordinates on a Cartesian plane (that is, using the scaled cosine and sine of its angle); or we can encode the position as a point on the complex plane using polar coordinates expressed as a complex exponential. While the two representations are equivalent, the latter is much more compact.

In the digital world, signals can be complex: indeed, why not? Digital signals are just sequences of numbers that will be processed by general-purpose computational units, and therefore these sequences can certainly be complex-valued. While the interfaces to and from the physical world will necessarily handle real values only, *internally* a DSP system will often be easier to design if complex-valued operations are allowed; this is particularly true in the case of communication systems. By starting off with complex exponentials as the prototypical oscillation, we are already equipped with a more versatile tool for frequency analysis.

3.2.2 Properties of complex oscillations

In discrete time, things are a bit different with respect to the properties of classic sinusoidal functions of a real variable.

Periodicity. First of all, perhaps surprisingly, not all complex exponential sequences are periodic in n. Without loss of generality, consider a sequence with zero phase offset $w[n] = e^{j\omega n}$; for $\mathbf{w}$ to be periodic, there must exist an integer N so that

$$w[n] = w[n + N], \quad \forall n \in \mathbb{Z}.$$

The above expression is equivalent to $e^{j\omega N} = 1$, that is, there must exist an integer M such that

$$\omega N = 2\pi M.$$

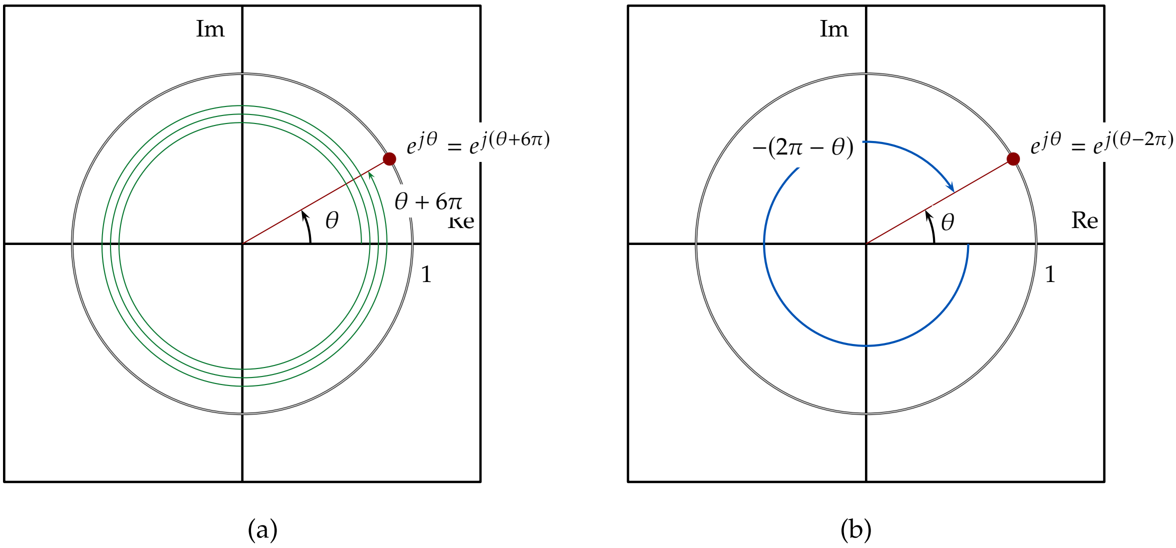

Figure 3.4: *Same point, many aliases: (a) adding multiples of* 2π *to a pure phase term does not change its position; (b) a positive (counterclockwise) displacement by* θ *is equivalent to a negative (clockwise) displacement by* $(2\pi - \theta)$

Periodicity therefore requires the frequency to be of the form

$$\omega = \frac{M}{N} 2\pi, \quad M, N \in \mathbb{Z} \tag{3.3}$$

or, in other words, in discrete time the only periodic oscillations are those *whose frequency is a rational multiple of* 2π.

Aliasing. Another peculiarity of discrete-time oscillations is that there is a limit on "how fast" we can go. To understand why, let's start by recalling that a pure phase term is always 2π-periodic, in the sense that

$$e^{j\theta} = e^{j(\theta+2k\pi)} \quad \forall k \in \mathbb{Z}.$$

This inherent phase ambiguity is called aliasing and stems from the simple fact that a point on the unit circle has an infinite number of possible "names", as shown in Figure 3.4-(a) — etymologically, "alias" is Latin for "otherwise". Applied to discrete-time complex exponential sequences, this clearly implies an upper limit on the rotational speed of a point around the unit circle, since adding multiples of 2π to the value of the frequency will not change the values of the samples:

$$e^{j(\omega+2k\pi)n} = e^{j\omega n} e^{j(kn)2\pi} = e^{j\omega n} \quad \forall k \in \mathbb{Z};$$

consequently, we can limit the range of distinct angular speeds to a representative interval of size 2π. To choose the most suitable interval, consider what happens if we gradually increase the frequency of a discrete-time complex exponential starting from zero:

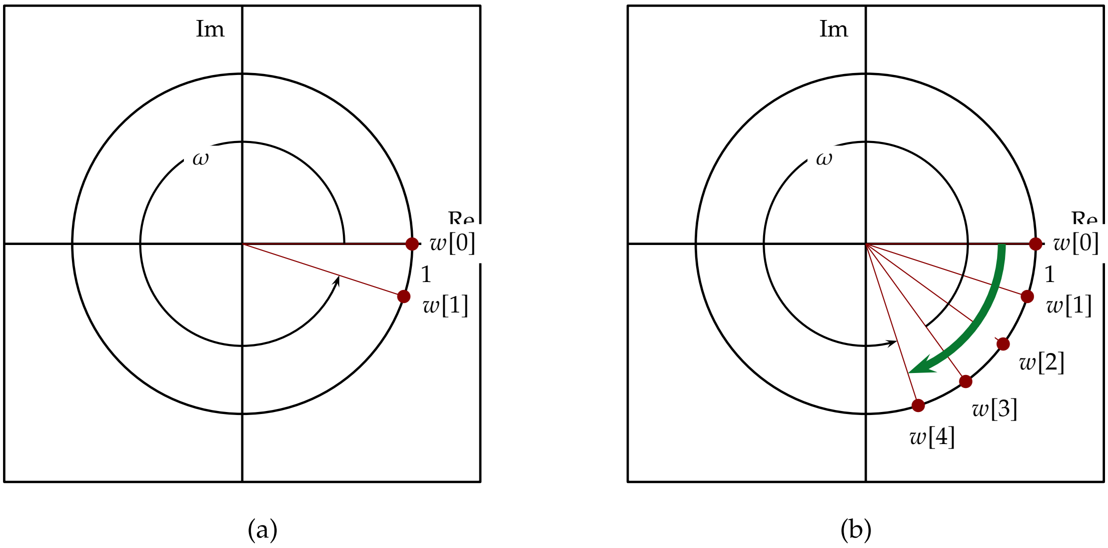

Figure 3.5: *Complex-exponential sequence at angular speed $\omega = 2\pi - \theta$, with θ small: (a) at each step, the point's displacement is larger than π; (b) the movement is more "economical" if one assumes a negative frequency $\omega' = -\theta$.*

- for $0 \leq \omega < \pi$ we have a counterclockwise motion with increasing angular speed, i.e., we cover the full circle in fewer and fewer steps.

- for $\omega = \pi$ we have the maximum possible forward speed of a discrete-time complex exponential; this corresponds to a sequence whose values are alternating between $+1$ and -1, which represents the maximum displacement attainable by successive points on the unit circle (antipodal points); we cover the full circle in 2 steps.

- for $\pi < \omega < 2\pi$ at each step the point on the unit circle moves by more than π, as shown in Figure 3.5. Such a large counterclockwise motion is more "economically" explained by a *clockwise* motion by an angle $2\pi - \omega < \pi$, i.e., the motion is better described by a *negative* frequency whose magnitude is less than π.

- for $\omega \geq 2\pi$ we can subtract a suitable multiple of 2π to ω until we fall into one of the three preceding cases.

The reference interval of choice for angular frequencies is therefore $[-\pi, \pi]$.

Note that, as we increase ω beyond π, we obtain a perceived reversal of direction and a *decreasing* angular speed. This aliasing phenomenon is well known in cinematography where it is called the *wagonwheel effect*; you can experience it in full by watching an old western movie: if a stagecoach enters the scene, you will see that its multi-spoked wheels seem to spin alternately backwards and forward as the speed of the vehicle changes. For a more detailed discussion of this optical illusion, see Appendix B.

3.3 The Discrete Fourier Transform (DFT)

Let's place ourselves in $\mathbb{C}^N$, the space of complex-valued signals of finite length N; what are all the possible sinusoidal signals in this space that span a *whole* number of periods over N points? We will now show that:

- there are exactly N such sinusoids
- their frequencies are all harmonically related, i.e. they are all multiples of the fundamental frequency $2\pi/N$:

$$\omega_k = \frac{2\pi}{N}k, \quad k = 0, 1, \ldots, N-1; \tag{3.4}$$

- the set of N length-N complex exponentials at frequencies ω_k form a set of orthogonal vectors and therefore a basis for $\mathbb{C}^N$.

With this basis, we are able to express *any* signal in $\mathbb{C}^N$ as a linear combination of N harmonically-related sinusoids; the set of N coefficients in the linear combination are called the *Discrete Fourier Transform* of the signal, which can be easily computed algorithmically for any input data vector.

3.3.1 The Fourier Basis for $\mathbb{C}^N$

The fundamental oscillatory signal, the discrete-time complex-exponential $e^{j\omega n}$, is equal to 1 for $n = 0$; therefore, if the signal is to span a whole number of periods over N points, we must have

$$e^{j\omega N} = 1.$$

In the complex field, the equation $z^N = 1$ has N distinct solutions, given by the N roots of unity

$$z_k = e^{j\frac{2\pi}{N}k}, \quad k = 0, \ldots, N-1,$$

and so the N possible frequencies that fulfill the N-periodicity requirements are those given in (3.4). We can now use these frequencies to define a set $\{\mathbf{w}_k\}_k$ containing N signals of length N, where

$$w_k[n] = e^{j\frac{2\pi}{N}kn} \quad n, k = 0, 1, \ldots, N-1. \tag{3.5}$$

The real and imaginary parts of $\mathbf{w}_k$ for $N = 32$ and for some values of k are plotted in Figures 3.6 to 3.11; note how $\mathbf{w}_k = \mathbf{w}^*_{N-k}$.

The vectors in $\{\mathbf{w}_k\}$ are mutually orthogonal; to show this, we start from the definition of inner product in $\mathbb{C}^N$

$$\langle \mathbf{x}, \mathbf{y} \rangle = \sum_{n=0}^{N-1} x^*[n]y[n],$$

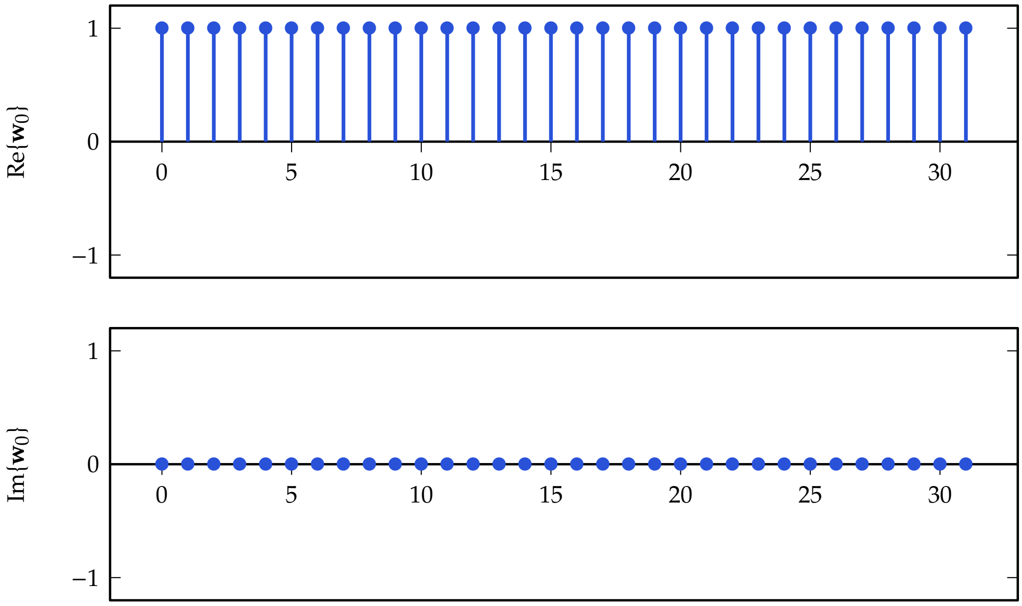

Figure 3.6: *Fourier basis vector* $\mathbf{w}_0$.

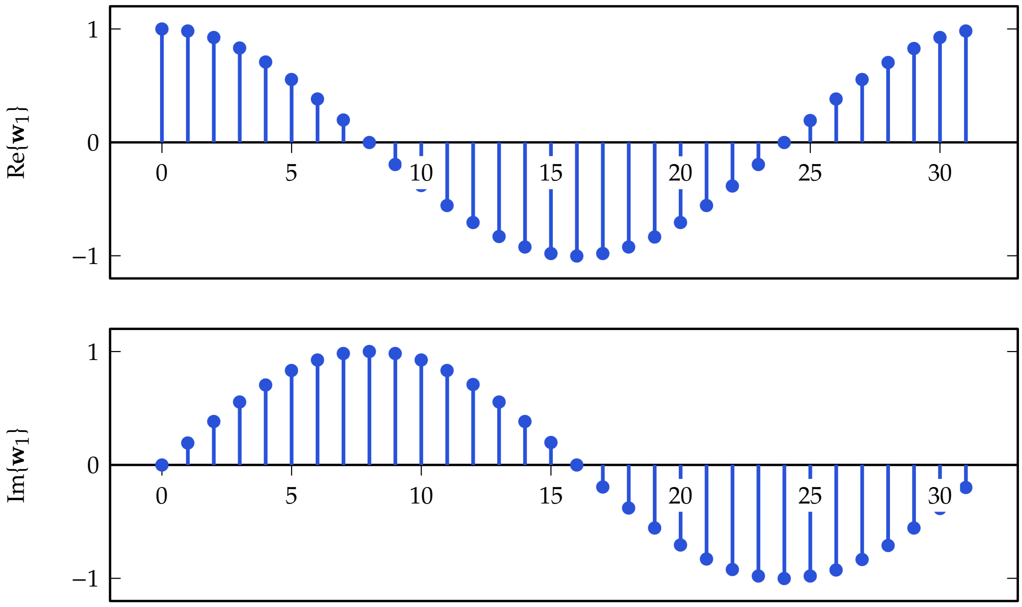

Figure 3.7: *Fourier basis vector* $\mathbf{w}_1$.

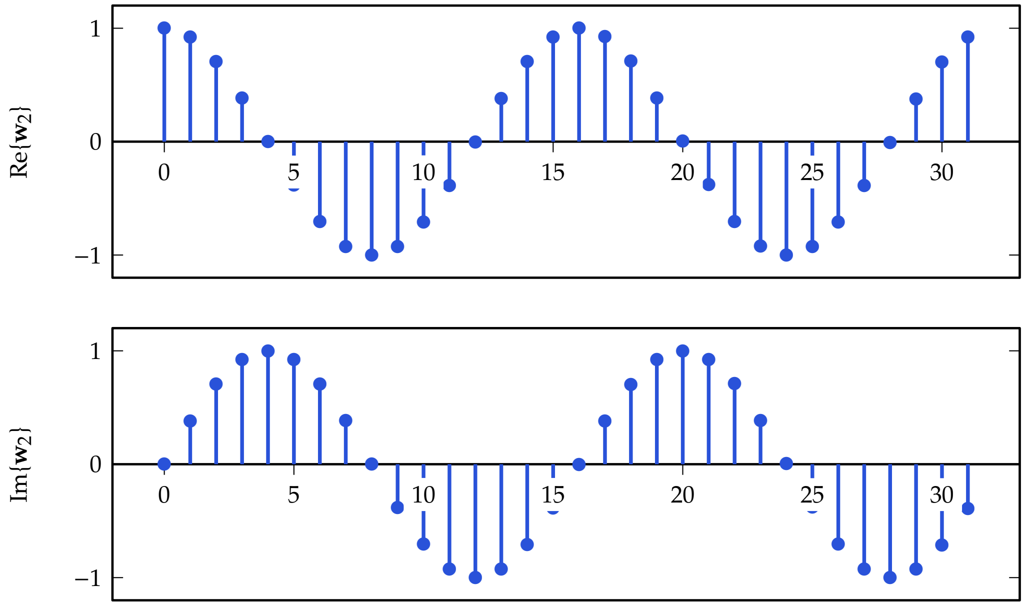

Figure 3.8: *Fourier basis vector* $\mathbf{w}_2$.

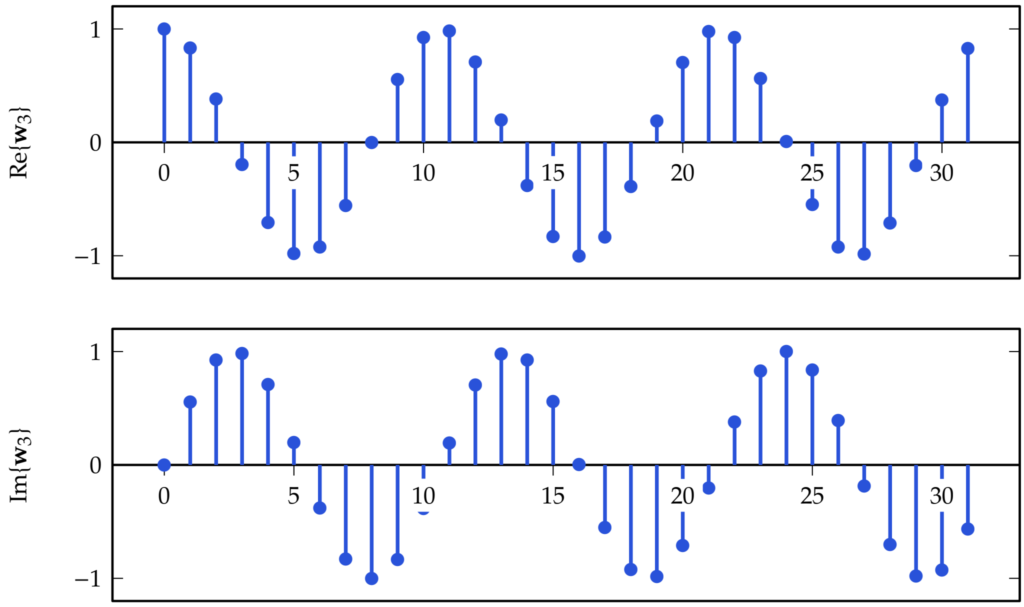

Figure 3.9: *Fourier basis vector* $\mathbf{w}_3$.

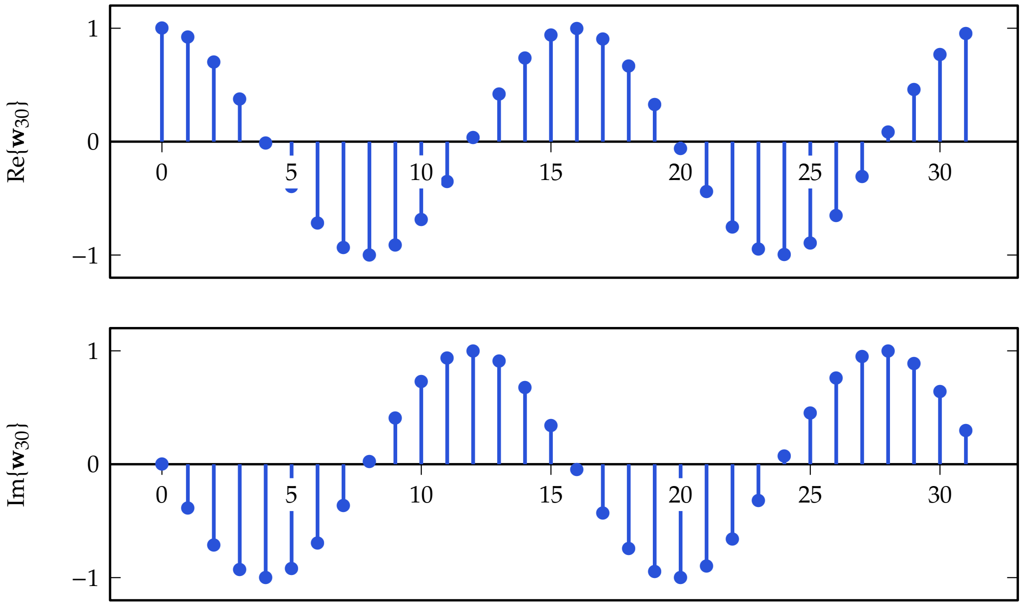

Figure 3.10: *Fourier basis vector* $\mathbf{w}_{30}$.

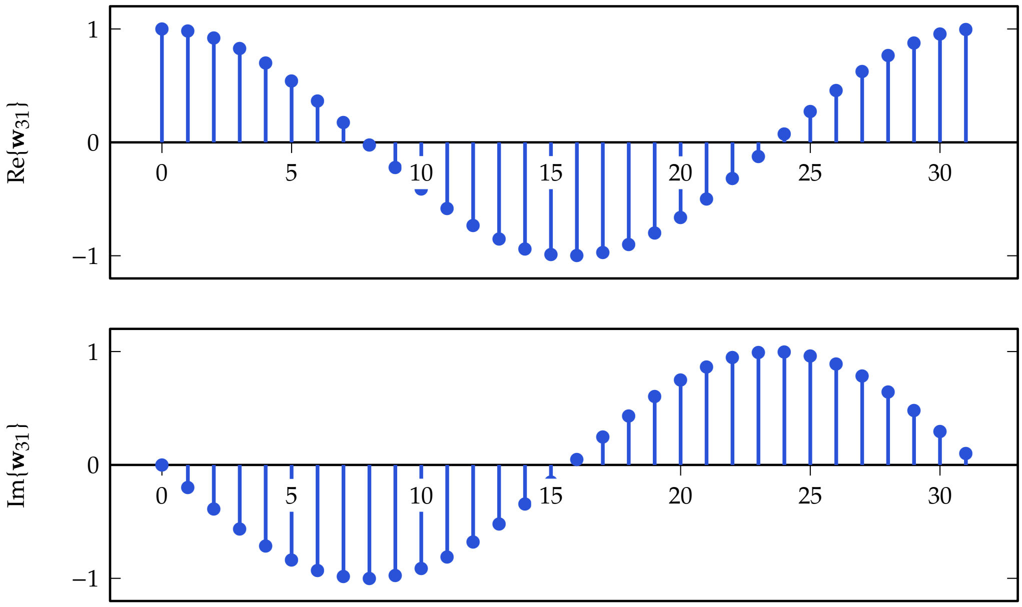

Figure 3.11: *Fourier basis vector* $\mathbf{w}_{31}$.

we remember that $(e^{j\theta})^* = e^{-j\theta}$, and we use the geometric sum formula to obtain

$$\begin{aligned}
\langle \mathbf{w}_h, \mathbf{w}_k \rangle &= \sum_{n=0}^{N-1} \left(e^{j\frac{2\pi}{N}hn}\right)^* e^{j\frac{2\pi}{N}kn} \\
&= \sum_{n=0}^{N-1} e^{j\frac{2\pi}{N}(k-h)n} \\
&= \begin{cases} N & \text{for } h = k \\ \dfrac{1 - e^{j\frac{2\pi}{N}(k-h)N}}{1 - e^{j\frac{2\pi}{N}(k-h)}} = 0 & \text{for } k \neq h \end{cases}
\end{aligned} \tag{3.6}$$

since $(k-h)$ is an integer and therefore, when $k - h \neq 0$, $e^{j2(k-h)\pi} = e^{j2\pi} = 1$. Because of orthogonality the vectors form a basis for $\mathbb{C}^N$, called the *Fourier basis* for the space of finite-length signals. In compact form, we can express the orthogonality of the Fourier vectors with the notation

$$\langle \mathbf{w}_h, \mathbf{w}_k \rangle = N\,\delta[h-k]. \tag{3.7}$$

Clearly the basis is not orthonormal; while it could be normalized by multiplying each vector by $1/\sqrt{N}$, in signal processing practice it is customary to keep the normalization factor explicit in the change of basis formulas. We too will follow this convention, that exists primarily for computational reasons.

3.3.2 The DFT as a Change of Basis

In the previous chapter we have illustrated how we can efficiently perform an orthonormal change of basis in $\mathbb{C}^N$; the Discrete Fourier Transform is such a transformation, allowing us to move from the time domain, represented by the canonical basis $\{\boldsymbol{\delta}_k\}$, to the frequency domain, spanned by the Fourier basis $\{\mathbf{w}_k\}$. Here, since the Fourier basis is orthogonal but not orthonormal, we simply need to slightly adjust the formulas to take into account the required normalization factors.

Given a vector $\mathbf{x} \in \mathbb{C}^N$ and the Fourier basis $\{\mathbf{w}_k\}$, we can always express $\mathbf{x}$ as the following linear combination of basis vectors:

$$\mathbf{x} = \frac{1}{N} \sum_{k=0}^{N-1} X[k]\,\mathbf{w}_k. \tag{3.8}$$

Using the orthogonality of the Fourier basis in (3.7), and taking the left inner product of the left- and right-hand sides of (3.8) with each $\mathbf{w}_k$, we can see that the N complex scalars $X[k]$, called the *Fourier coefficients*, can be obtained simply as

$$X[k] = \langle \mathbf{w}_k, \mathbf{x} \rangle \tag{3.9}$$

The coefficients thus capture the similarity between $\mathbf{x}$ and each of the basis vectors via an inner product; structurally, the set of N Fourier coefficients is also a vector $\mathbf{X} \in \mathbb{C}^N$ so that the DFT is an endomorphism on the space of finite-length sequences.

The DFT in algorithmic form. The analysis and synthesis formulas can be written out explicitly in terms of the elements in the original data vector and in the vector of Fourier coefficients. This formulation highlights the computable algorithmic nature of the DFT and provides a straightforward way to implement the transform numerically.

The DFT coefficients can be computed using the following formula:

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{N}nk}, \qquad k = 0, \ldots, N-1 \tag{3.10}$$

while the inverse DFT is computed from the Fourier coefficients as

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{j\frac{2\pi}{N}nk}, \qquad n = 0, \ldots, N-1. \tag{3.11}$$

The explicit formulas allows us to appreciate the highly structured form of the summations, which leads to extremely efficient implementations as we will discuss briefly in Item **??** at the end of this chapter.

The DFT in matrix form. From our discussion in the previous chapter we know that a change of basis in $\mathbb{C}^N$ can be expressed as a matrix-vector multiplication:

$$\mathbf{X} = \mathbf{W}\mathbf{x};$$

the matrix $\mathbf{W}$ is built by stacking the Hermitian transposes of the basis vectors as

$$\mathbf{W} = \begin{bmatrix} \mathbf{w}_0^H \\ \hline \mathbf{w}_1^H \\ \hline \vdots \\ \hline \mathbf{w}_{N-1}^H \end{bmatrix} = \begin{bmatrix} W_N^0 & W_N^0 & W_N^0 & \ldots & W_N^0 \\ W_N^0 & W_N^1 & W_N^2 & \ldots & W_N^{N-1} \\ W_N^0 & W_N^2 & W_N^4 & \ldots & W_N^{2(N-1)} \\ & & & \ldots & \\ W_N^0 & W_N^{N-1} & W_N^{2(N-1)} & \ldots & W_N^{(N-1)^2} \end{bmatrix} \tag{3.12}$$

where, for convenience, we have introduced the scalar $W_N = e^{-j\frac{2\pi}{N}}$. Again using (3.7), it is immediate to show that $\mathbf{W}^H\mathbf{W} = \mathbf{I}\, N$, where $\mathbf{I}$ is the identity matrix. Since the change of basis is invertible, both $\mathbf{x}$ and $\mathbf{X}$ represent the same information, albeit from two different "points of view": $\mathbf{x}$ lives in the time domain, while $\mathbf{X}$ lives in the frequency domain. In order to "go back" we can use the synthesis formula in matrix form, taking into account the explicit normalization factor:

$$\mathbf{x} = \frac{1}{N}\mathbf{W}^H\mathbf{X}. \tag{3.13}$$

Some examples of Fourier matrices for low-dimensional spaces are:

$$\mathbf{W}_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{3.14}$$

$$\mathbf{W}_3 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & W_3 & W_3^2 \\ 1 & W_3^2 & W_3^4 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & W_3 & W_3^2 \\ 1 & W_3^2 & W_3 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 1 & 1 \\ 1 & \frac{-1-j\sqrt{3}}{2} & \frac{-1+j\sqrt{3}}{2} \\ 1 & \frac{-1+j\sqrt{3}}{2} & \frac{-1-j\sqrt{3}}{2} \end{bmatrix} \tag{3.15}$$

$$\mathbf{W}_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & W_4 & W_4^2 & W_4^3 \\ 1 & W_4^2 & W_4^4 & W_4^6 \\ 1 & W_4^3 & W_4^6 & W_4^9 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & W_4 & W_4^2 & W_4^3 \\ 1 & W_4^2 & 1 & W_4^2 \\ 1 & W_4^3 & W_4^2 & W_4 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \tag{3.16}$$

Please note that:

- the elements in the first row and the first columns are all equal to one since $W_N^0 = 1$ for all N;
- powers of W_N can be computed modulo N because of the "aliasing" property of complex exponentials discussed in Section 3.2.2:

  $$W_N^n = W_N^{n \bmod N};$$

- the matrices for a DFT of size two and four involve no multiplications (multiplication by $\pm j$ simply requires swapping real and imaginary parts, and a change of sign).

3.4 Examples

3.4.1 Plotting the DFT

The DFT of a N-point signal is in general a complex-valued vector of size N (even if the input is real-valued). The best way to look at the DFT coefficients is to examine their role in the synthesis formula (3.8), in which the time-domain signal is exactly reconstructed by a weighed sum of oscillatory components. Each DFT coefficient affects the corresponding oscillation in two ways:

- the magnitude sets the peak-to-peak amplitude of the oscillation;

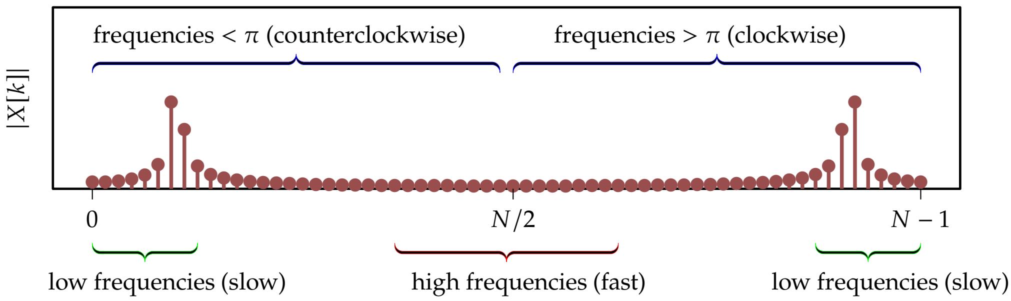

Figure 3.12: *Reading a DFT plot.*

- the phase sets the phase offset, that is, the initial delay of the oscillation.

Since these two actions are clear and distinct, it is customary to represent the DFT in terms of magnitude and phase rather than showing its real and imaginary parts. Graphically, magnitude and phase are plotted as a sequence of discrete values indexed by an integer "frequency" value $0 \leq k < N$; the index implicitly identifies the actual frequency $\omega_k = (2\pi/N)k$ of the associated oscillation. As we move along the horizontal axis from left to right, the corresponding frequencies describe a counterclockwise rotation with increasing speeds, until $k = \lfloor N/2 \rfloor$; beyond this midpoint the rotation will switch to clockwise and the speed will start to decrease, as shown in Figure 3.12.

DFT magnitude. Since the basis vectors are orthogonal, each basis vector carries an independent piece of information and the overall energy of the signal will be equivalent the sum of the energy of all components. This is in fact a consequence of Parseval's theorem which, taking the normalization factor into account, states that the signals's energy is preserved across a change of basis:

$$\sum_{n=0}^{N-1} |x[n]|^2 = \frac{1}{N} \sum_{k=0}^{N-1} |X[k]|^2.$$

Since the energy of a DFT basis vector is equal to N, the square magnitude of the k-th DFT coefficients represents the amount energy in the signal that is present at a frequency $\omega_k = (2\pi/N)k$; as a consequence, the DFT magnitude is often plotted squared. If the time-domain signal is real-valued, the magnitude of its DFT is symmetric:

$$|X[n]| = |X[N-n]|;$$

in these cases, it is often customary to show only the first half of the coefficients in a magnitude plot.

DFT phase. For a pure sinusoid, a phase offset corresponds to a shift in time; in the reconstruction formula, therefore, the different oscillatory components will be added together with a relative alignment that depends on the phase of the DFT, and this will determine the shape of the signal in the time domain, as exemplified in Figure 3.13. The phase, due to its inherent 2π periodicity, is usually plotted over a range of width 2π (normally the $[-\pi, \pi]$ interval) and values outside of this range are "wrapped" (i.e., integer multiples of 2π are added to the original value until the result is within range).

3.4.2 Elementary DFT pairs

We will now compute the DFT of some elementary signals in $\mathbb{C}^N$; for the illustrations, plotting the coefficients in magnitude and phase, the value $N = 64$ is used as an example.

This section will also introduce several conventions that are in use to notate the relationship between a time-domain signal and its frequency-domain counterpart; be aware that none is perfect and that each has its own advantages and drawbacks. In vector notation we can elegantly write things like

$$\mathbf{x} \stackrel{\text{DFT}}{\longleftrightarrow} \mathbf{X}$$

or, equivalently,

$$\text{DFT}\{\mathbf{x}\} = \mathbf{X};$$

this notation is compact and precise, since it does not make use of explicit index variables for the time and frequency data vectors. But it is also impractical to use when applying the DFT to expressions that explicitly invoke the time or frequency index; in these cases we will resort to a less precise approach that uses placeholder variables (generally n for the time domain and k for the frequency domain):

$$x[n] \stackrel{\text{DFT}}{\longleftrightarrow} X[k].$$

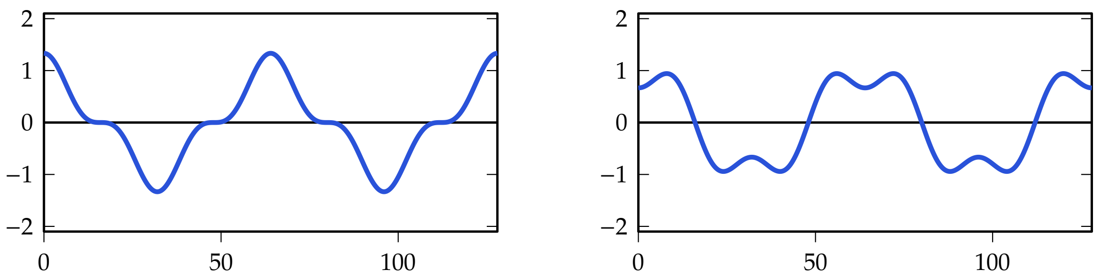

Figure 3.13: *Effects of phase alignment on the shape of a signal in the time domain. Both plots show* $x[n] = \cos(\omega_0 n) + (1/3)\cos(3\omega_0 n + \theta)$ *but* $\theta = 0$ *in the left panel and* $\theta = \pi$ *on the right; both signals have exactly the same magnitude DFT.*

Impulse. The DFT of the discrete-time delta is the constant signal equal to one since

$$\sum_{n=0}^{N-1} \delta[n] e^{-j\frac{2\pi}{N}nk} = e^{-j\frac{2\pi}{N}nk}\Big|_{n=0} = 1 \quad \forall k.$$

In vector notation we can write

$$\boldsymbol{\delta} \stackrel{\text{DFT}}{\longleftrightarrow} \mathbf{1} \tag{3.17}$$

or, equivalently,

$$\text{DFT}\{\boldsymbol{\delta}\} = \mathbf{1} \tag{3.18}$$

where $\mathbf{1}$ is a vector of all ones, and whose length is equal to the dimensionality of the underlying space.

For the shifted delta $\boldsymbol{\delta}_m = \mathcal{S}^{-m}\{\boldsymbol{\delta}\}$, the analysis formula yields

$$\sum_{n=0}^{N-1} \delta[n-m] e^{-j\frac{2\pi}{N}nk} = e^{-j\frac{2\pi}{N}mk}$$

so that, compactly, we can write

$$\boldsymbol{\delta}_m \stackrel{\text{DFT}}{\longleftrightarrow} \mathbf{w}_m^* \tag{3.19}$$

or, less formally,

$$\delta[n-m] \stackrel{\text{DFT}}{\longleftrightarrow} e^{-j\frac{2\pi}{N}mk}. \tag{3.20}$$

The DFT of a canonical basis vector in time is thus the conjugate of the corresponding vector in the Fourier basis; since all the elements of a Fourier basis vector have unit magnitude, we have that the most "concentrated" signals in time have nonzero content at every frequency, that is, they have the most "spread-out" spectrum. This inverse relationship between time and frequency supports is a general property of Fourier pairs and will reappear frequently in the rest of the course.

Rectangular signal. Consider the step signal $\mathbf{x}$ defined by

$$x[n] = \begin{cases} 1 & \text{for } 0 \leq n < M \\ 0 & \text{for } M \leq n < N, \end{cases} \tag{3.21}$$

shown in Figure 3.14 for $M = 5$ and $N = 64$. We can express the signal as

$$\mathbf{x} = \sum_{m=0}^{M-1} \boldsymbol{\delta}_m$$

and, exploiting (3.19) and the obvious linearity of the DFT, obtain

$$\text{DFT}\{\mathbf{x}\} = \mathbf{X} = \sum_{m=0}^{M-1} \mathbf{w}_m^*.$$

The coefficients can be computed explicitly as:

$$X[k] = \sum_{n=0}^{M-1} e^{-j\frac{2\pi}{N}nk} = \frac{1 - e^{-j\frac{2\pi}{N}Mk}}{1 - e^{-j\frac{2\pi}{N}k}} \tag{3.22}$$

$$= \frac{e^{-j\frac{\pi}{N}Mk}\left(e^{j\frac{\pi}{N}Mk} - e^{-j\frac{\pi}{N}Mk}\right)}{e^{-j\frac{\pi}{N}k}\left(e^{j\frac{\pi}{N}k} - e^{-j\frac{\pi}{N}k}\right)}$$

$$= \frac{\sin(\pi Mk/N)}{\sin(\pi k/N)}\, e^{j\frac{\pi}{N}(M-1)k}. \tag{3.23}$$

In the derivation above, we have manipulated the expression for $X[k]$ into a product of a real-valued term (which captures the magnitude) and a pure phase term; this allows us to easily plot the DFT as in Figure 3.14. Note that, while nominally the phase grows linearly with k, the plot shows a "wrapped" phase as explained in Section 3.4.1.

Constant signal. By setting $M = N$ in (3.22) we obtain the DFT of the constant signal $\mathbf{1}$; the sum, as in (3.6), uses once again the orthogonality of the roots of unity and yields

$$X[k] = \sum_{n=0}^{M-1} e^{-j\frac{2\pi}{N}nk} = \begin{cases} N & \text{for } k = 0 \\ 0 & \text{otherwise.} \end{cases} \tag{3.24}$$

Compactly,

$$\mathbf{1} \stackrel{\text{DFT}}{\longleftrightarrow} N\boldsymbol{\delta} \tag{3.25}$$

which, up to a normalization factor, is the dual of (3.17); in this case, the time-domain signal with the largest support leads to a DFT with a single nonzero coefficient.

Harmonic sinusoids. The fundamental frequency for $\mathbb{C}^N$ is $2\pi/N$ and a complex exponential at a multiple of this frequency will coincide with a Fourier basis vector. Because of the orthogonality relation

$$\langle \mathbf{w}_k, \mathbf{w}_m \rangle = N\delta[k-m]$$

the DFT of a harmonic complex exponential is

$$\text{DFT}\{\mathbf{w}_m\} = N\boldsymbol{\delta}_m; \tag{3.26}$$

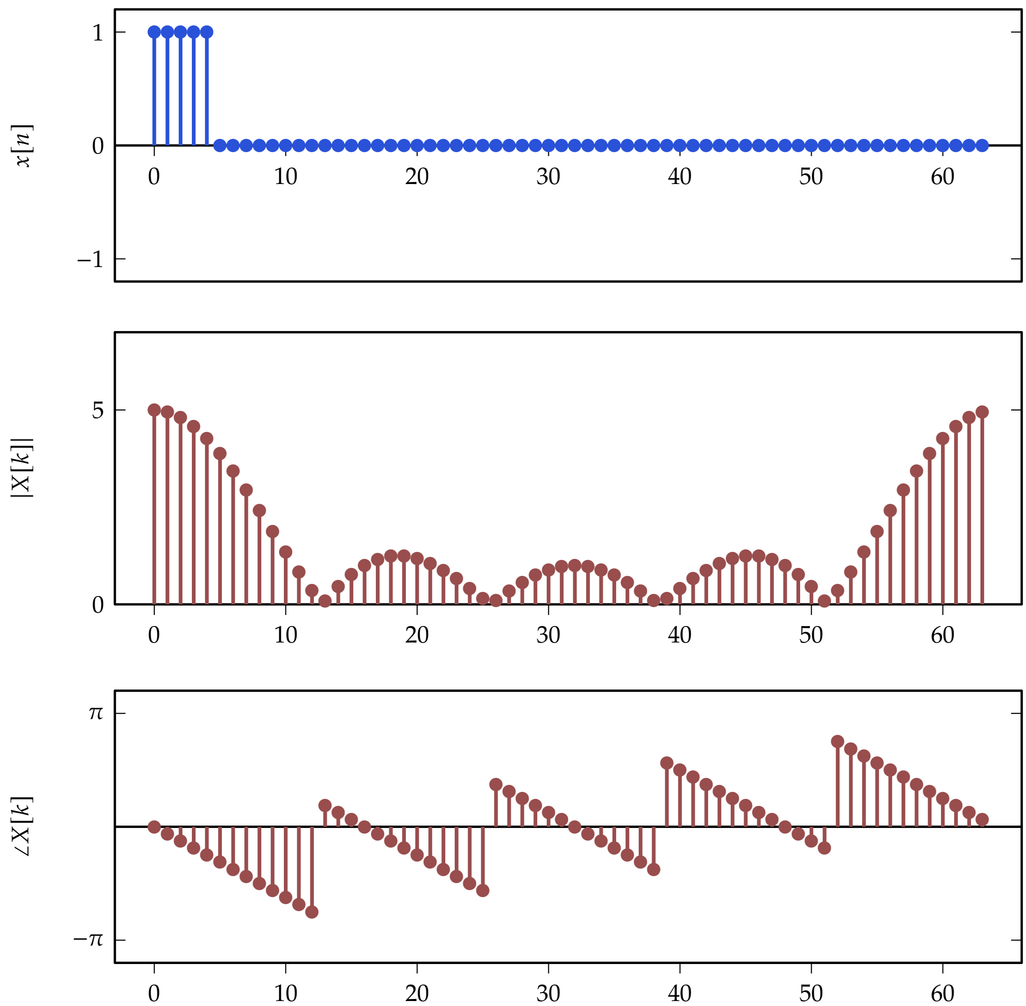

Figure 3.14: *DFT of a step signal.*

up to a normalization factor, this is the dual of (3.19). Note that the result in the previous section, for the constant signal $\mathbf{x} = \mathbf{1}$, is just a particular case of the above relationship when $m = 0$.

With this we can also easily compute the DFT of standard trigonometric functions whose frequency is a multiple of the fundamental frequency for the space. Consider for instance the signal $\mathbf{x}$ defined by

$$x[n] = \cos\left(\frac{\pi}{8}n\right), \qquad n = 0, 1, 2, \ldots, 63.$$

With a simple manipulation we can write:

$$\begin{aligned}\cos\left(4\frac{2\pi}{64}n\right) &= \frac{1}{2}\left(e^{j4\frac{2\pi}{64}n} + e^{-j4\frac{2\pi}{64}n}\right)\\ &= \frac{1}{2}\left(e^{j4\frac{2\pi}{64}n} + e^{j60\frac{2\pi}{64}n}\right)\end{aligned}$$

where we have used the fact that we can take all frequency indexes modulo 64 because of the aliasing property of complex exponentials; we can therefore express $\mathbf{x}$ as

$$\mathbf{x} = \frac{1}{2}\mathbf{w}_4 + \frac{1}{2}\mathbf{w}_{60}.$$

and, by linearity,

$$\text{DFT}\{\mathbf{x}\} = (N/2)(\boldsymbol{\delta}_4 + \boldsymbol{\delta}_{60}). \tag{3.27}$$

Explicitly, the DFT coefficients are

$$X[k] = \begin{cases} 32 & k = 4 \\ 32 & k = 60 \\ 0 & \text{otherwise} \end{cases}$$

The DFT of the signal is plotted in Figure 3.15; the spectrum shows how the entire frequency content of the signal is concentrated over two single frequencies. Since the original signal is real-valued, the DFT component at $k = 60$ ensures that the imaginary parts in the reconstruction formula cancel out; this symmetry is a general property of the Fourier transform that we will formalize in Section 3.6.

Consider now a slight variation of the previous signal obtained by introducing a phase offset:

$$x[n] = \cos\left(\frac{\pi}{8}n + \frac{2\pi}{3}\right), \qquad n = 0, 1, 2, \ldots, 63.$$

Again, we can easily manipulate the signal to obtain

$$\mathbf{x} = \frac{e^{j2\pi/3}}{2}\mathbf{w}_4 + \frac{e^{-j2\pi/3}}{2}\mathbf{w}_{60}$$

so that the resulting DFT coefficients are all zero except for

$$X[4] = 32\, e^{j2\pi/3}$$
$$X[60] = 32\, e^{-j2\pi/3}.$$

The resulting DFT is plotted in Figure 3.16; the magnitude does not change but the phase offset is reflected by the nonzero phase values at $k = 4, 60$.

Non-harmonic sinusoids Consider now a sinusoid whose frequency is *not* a multiple of the fundamental frequency for the space, such as

$$x[n] = \cos\left(\frac{\pi}{5}n\right), \qquad n = 0, 1, 2, \ldots, 63.$$

In this case we cannot decompose the signal into a sum of basis vectors and we must therefore explicitly compute all the DFT coefficients. We could do this algebraically and work out the resulting geometric sums as we did for the step signal. More conveniently, however, since the DFT is an *algorithm*, we can just use a standard numerical package (Numpy, Matlab, Octave) and use the built-in `fft()` function. The resulting DFT is shown in Figure 3.17 and the important observation is that in this case *all* the DFT coefficients are nonzero. While the magnitude is larger for frequencies close to that of the time-domain sinusoid ($6\pi/64 < \pi/5 < 7\pi/64$), in order to reconstruct $\mathbf{x}$ exactly we need a nonzero contribution from each one of the basis vectors.

3.5 DFT, periodicity, and DFS

Let's return to the DFT reconstruction formula (3.11), which is formally defined only for $0 \leq n < N$. If we let the index n take values outside of this interval, however, we can always write[1] $n = mN + i$ with $m \in \mathbb{Z}$ and $i = n \mod N$ so that

$$x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k] e^{j\frac{2\pi}{N}ik}\, e^{j2\pi mk} = x[n \mod N]. \tag{3.28}$$

Because of the aliasing property for complex exponentials, the inverse DFT formula is in fact valid for all $n \in \mathbb{Z}$ and it generates an N-*periodic* sequence; this should not come as a surprise, given the N-periodic nature of the Fourier basis vectors for $\mathbb{C}^N$. Similarly, the DFT analysis formula remains valid if the frequency index k is allowed to take values outside the $[0, N-1]$ interval and the resulting sequence of DFT coefficients is N-periodic as well.

Because of its inherent periodicity, the DFT is the natural Fourier analysis tool in the case of periodic signals as well and the whole time-domain signal can be reconstructed via an inverse transform of the DFT coefficients for one period of the original signal;

[1] Remember our definition of the modulo operation in the first chapter

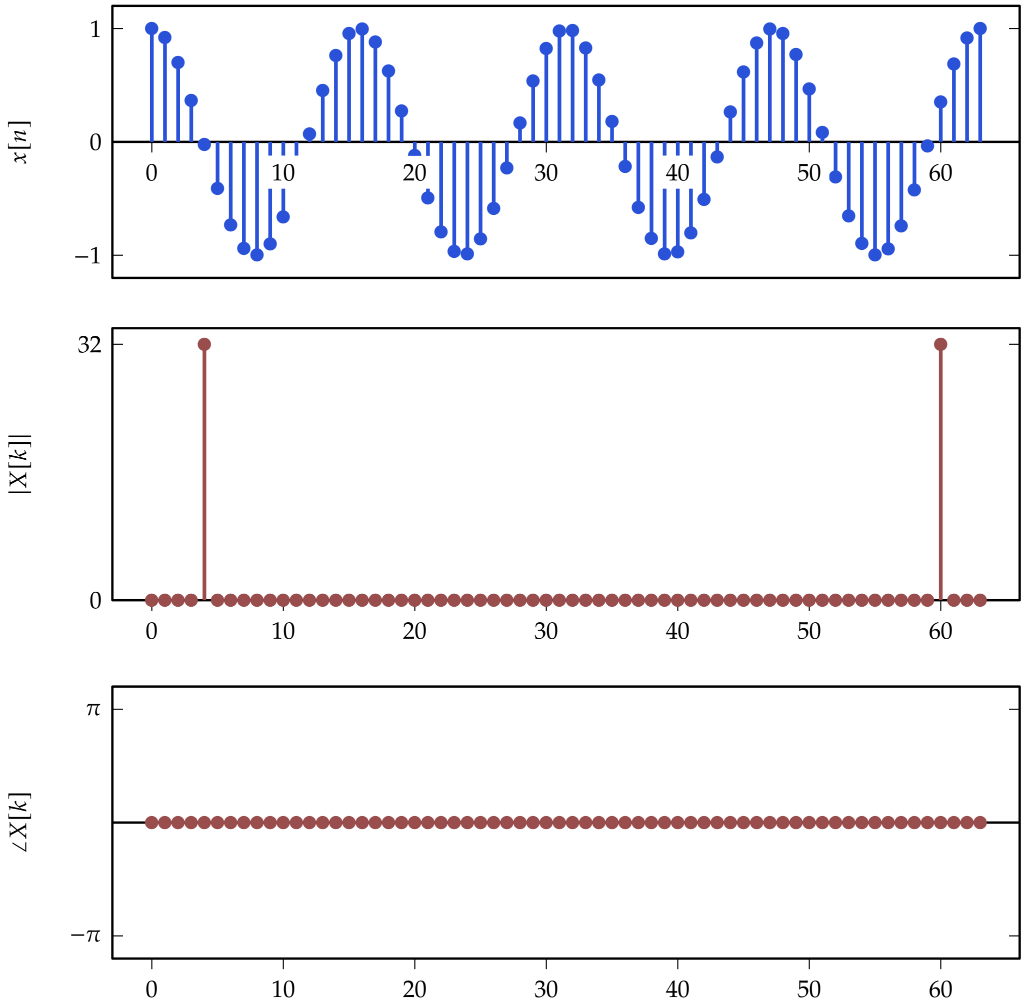

Figure 3.15: *DFT of* $x[n] = \cos\big((\pi/8)n\big)$.

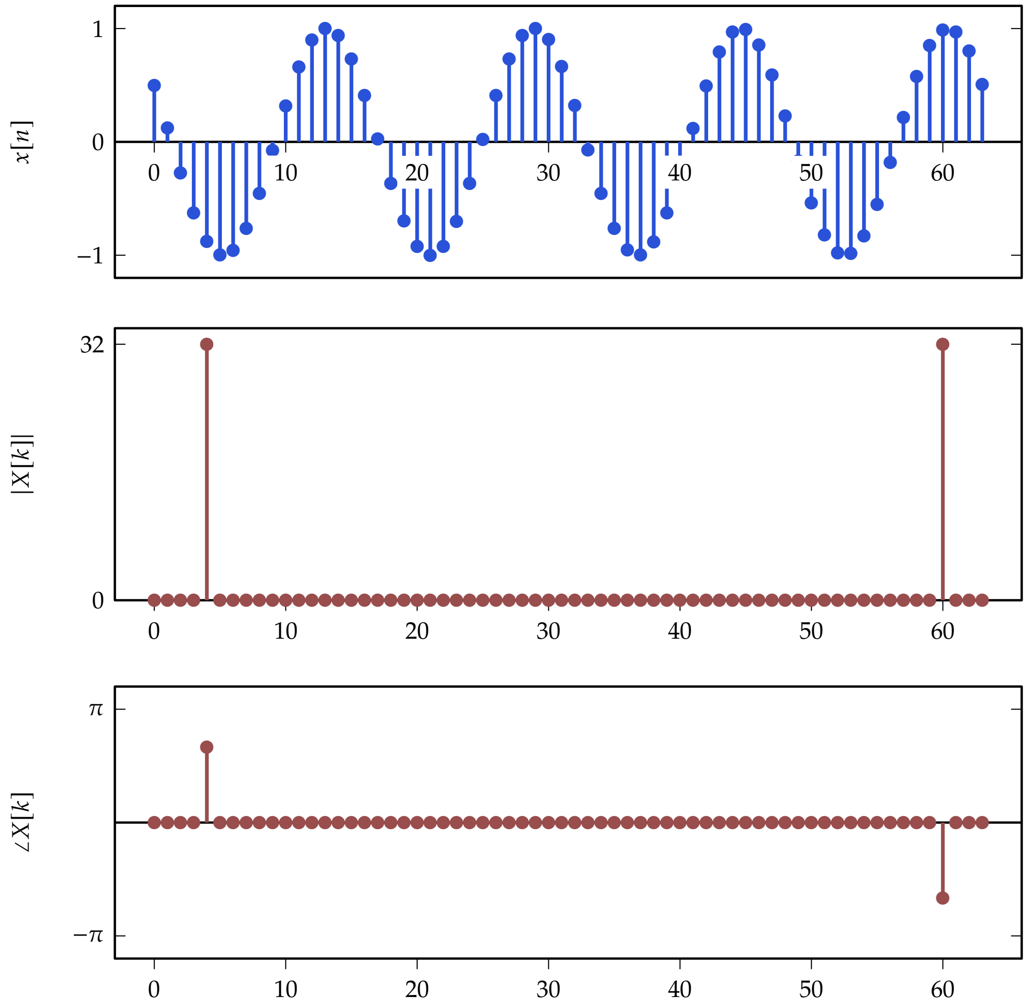

Figure 3.16: *DFT of* $x[n] = \cos\left((\pi/8)n + (2\pi/3)\right)$.

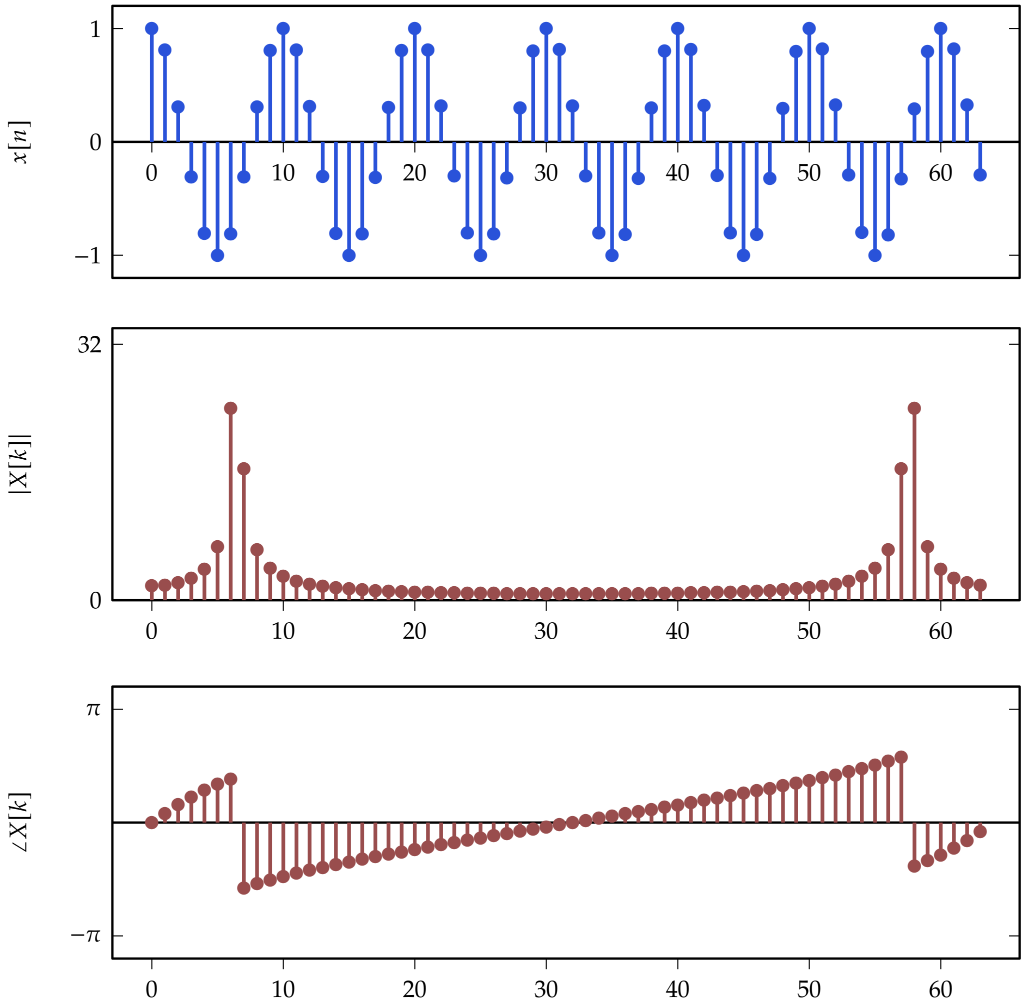

Figure 3.17: *DFT of* $x[n] = \cos\left((\pi/5)n\right)$.

this underscores the notion that, in a periodic signal, all information is contained in the samples spanning a single period. When emphasizing the periodic nature of both the signal and its Fourier coefficients, the transform is usually called the Discrete Fourier Series (DFS); this is only a change in name, however, because the DFS analysis and synthesis formulas are identical to (3.10) and (3.11), save for the range of the time and frequency indexes which now span all of $\mathbb{Z}$. The DFS of an N-periodic signal is simply the DFT of N consecutive samples (generally starting at $n = 0$:

$$\tilde{\mathbf{X}} = \mathrm{DFS}\{\tilde{\mathbf{x}}\} = \mathrm{DFT}\{\tilde{x}[0], \tilde{x}[1], \ldots, \tilde{x}[N-1]\}$$

The formal identity between DFT and DFS is a very important reminder that, in the space of finite-length signals, everything is implicitly N-periodic.

Circular shifts revisited. In the first chapter we stated that circular shifts are the natural way to interpret how the delay operator applies to finite-length signals; considering the inherent periodicity of the DFT, the reason should now be clear. Indeed, the delay operator is always well-defined for a periodic signal $\tilde{\mathbf{x}}$ and, if $\tilde{\mathbf{X}} = \mathrm{DFS}\{\tilde{\mathbf{x}}\}$, it is immediate to see that[2]

$$\tilde{x}[n - n_0] \overset{\mathrm{DFS}}{\longleftrightarrow} e^{-j\frac{2\pi}{N}n_0 k}\, \tilde{X}[k]; \tag{3.29}$$

in other words, a delay by n_0 samples in the time domain becomes a linear phase shift by $-2\pi n_0/N$ in the frequency domain. With a finite-length signal $\mathbf{x}$, for which time shifts are not well defined, we can still always compute the DFT, multiply the DFT coefficients by a linear phase shift and compute the inverse DFT. The result, by invoking the mathematical equivalence between DFT and DFS, is indeed a circular shift since

$$\frac{1}{N}\sum_{k=0}^{N-1}\left(e^{-j\frac{2\pi}{N}n_0 k}X[k]\right) e^{j\frac{2\pi}{N}nk} = x[(n-n_0) \mod N].$$

3.6 Properties of the DFT

In this section we will list, without formal proof, a series of elementary properties for the DFT. You are encouraged to verify why these properties are valid by manipulating the analysis and synthesis formulas directly. Obviously these properties apply identically to the DFS since all shifts are to be considered modulo N.

Linearity. The DFT is obviously a linear operator, since each coefficient is the result of an inner product in $\mathbb{C}^N$, and therefore

$$\mathrm{DFT}\{a\mathbf{x} + b\mathbf{y}\} = a\,\mathrm{DFT}\{\mathbf{x}\} + b\,\mathrm{DFT}\{\mathbf{y}\} \quad \forall a, b \in \mathbb{C}. \tag{3.30}$$

[2] This is one of the many cases in which a slight notational abuse actually helps. Formally we should say something like: "if $\mathbf{y} = \mathcal{S}^{-n_0}\mathbf{x}$ and $\mathbf{Y} = \mathrm{DFS}\{\mathbf{y}\}$, then $Y[k] = e^{-j\frac{2\pi}{N}n_0 k}\,\tilde{X}[k]$," which is a bit too much work.

Time shift. The circular shift of a signal in time correspond to multiplication by a linear phase factor in frequency:

$$x[(n - n_0) \mod N] \stackrel{\text{DFT}}{\longleftrightarrow} e^{-j\frac{2\pi}{N}n_0 k} X[k]. \tag{3.31}$$

Frequency shift. Multiplication in the time domain by a harmonic complex oscillation corresponds to a circular shift in frequency:

$$x[n]\, e^{j\frac{2\pi}{N}k_0 n} \stackrel{\text{DFT}}{\longleftrightarrow} X[k - k_0]. \tag{3.32}$$

As a consequence, by invoking linearity, we have

$$x[n] \cos\left(\frac{2\pi}{N}k_0 n\right) \stackrel{\text{DFT}}{\longleftrightarrow} (X[k - k_0] + X[k + k_0])/2.$$

Time and frequency reversal. Reversing a signal in time[3] reverses its transform:

$$\text{DFT}\left\{\mathcal{R}\mathbf{x}\right\} = \mathcal{R}\mathbf{X}. \tag{3.33}$$

Conjugating the time-domain signal results in both conjugation and reversal of the DFT:

$$\text{DFT}\left\{\mathbf{x}^*\right\} = \mathcal{R}\mathbf{X}^*. \tag{3.34}$$

Symmetries. A signal $\mathbf{x} \in \mathbb{C}^N$ (or a periodic sequence in $\tilde{\mathbb{C}}^N$) is called symmetric if $x[n] = x[N - n]$. If N is even, the symmetry relation leaves the values $x[0]$ and $x[N/2]$ unconstrained, whereas every other point has a "twin"; if N is odd, only $x[0]$ is unconstrained.

Similarly, an antisymmetric signal satisfies $x[n] = -x[N - n]$; for N even, this implies that $x[N/2] = 0$.

Finally, a signal is called Hermitian-symmetric if its real part is symmetric and its imaginary part is antisymmetric, that is, if $x[n] = x^*[N - n]$.

With this in mind, and using the time and frequency reversal properties, the following facts are easily verified:

- the DFT of a symmetric signal is symmetric
- the DFT of a real-valued signal is Hermitian-symmetric
- the magnitude DFT of a real-valued signal is symmetric.
- the DFT of a real-valued symmetric signal is real-valued and symmetric.

[3]Remember the way a time reversal operates on finite-length and periodic signal, as described in the first chapter

3.7 The DFT in practice

The DFT, as a change of basis, offers a fundamental *change in perspective:* the frequency domain allows us to look at the data from a different point of view and, in many cases, this new vantage point highlights features that were not immediately visible in the time domain and reveals hidden structural pattern that we can exploit to analyze and process the information contained in the signal. Most importantly, the DFT is a numerical algorithm that can be implemented very efficiently (see Example **??**) and its fundamental role as the workhorse of applied spectral analysis cannot be understated. In this section we will review some of the tools and tricks of the trade that become useful when applying the DFT to experimental data.

3.7.1 Labeling the frequency axis

We know that, in $\mathbb{C}^N$, the k-th DFT coefficient encodes the signal's content at frequency $\omega_k = 2\pi k/N$. In practical applications, however, we are more familiar with expressing frequency in units of hertz (Hz), namely, the number of *cycles per second* of a periodic phenomenon such as an oscillation. When the set of N data samples is obtained via uniformly-spaced measurements of a natural phenomenon, the associated DFT plot can be relabeled so that each DFT index corresponds to a real-world value in hertz and, for this, all we need to know is the duration of the measurement interval in physical units of time.

Assume that we observe a real-world signal for T seconds and we obtain N data samples (which is equivalent to saying that the signal was sampled at a rate of $F_s = N/T$ samples per second). Since we are now in discrete time we know that the fastest frequency is $\omega = \pi$, corresponding to the DFT index[4] $k = N/2$. This fastest frequency completes one cycle in exactly *two* samples (think about a point on the unit circle going +1, -1, +1, -1, etc.) or, equivalently, $N/2$ cycles over N samples. Since our set of samples spans a real-world time window of T seconds, the fastest frequency in the DFT plot will correspond to

$$f_N = \frac{N/2}{T} = \frac{F_s}{2} \quad \text{cycles per second (Hz).} \tag{3.35}$$

Since the frequencies in a DFT are uniformly spaced, the DFT indices between 0 and $N/2$ can be mapped to Hz linearly as

$$k \rightarrow \frac{F_s}{N} k \text{ Hz}, \qquad 0 \leq k \leq N/2.$$

If the dataset is real-valued (which is normally the case when analyzing real-world data) the DFT values for $k > N/2$ are rarely addressed explicitly because of the inherent

[4]For the sake of simplicity, let's assume here that N is even, so that $N/2$ is an integer and $e^{j\pi n}$ is among the basis vectors for $\mathbb{C}^N$. If N is odd, the set of harmonic frequencies for the space will not include $\omega = \pi$ but the relabeling argument does not change. This slight difference between and even- and odd-length DFTs crops up frequently and it is just a minor annoyance when trying to be very precise with the formulas; but it never really affects the substance of the discussion.

symmetries of the transform; when needed, however, they are best mapped to negative frequencies (that is, clockwise rotations) via

$$k \rightarrow \frac{F_s}{2} - \frac{F_s}{N} k \text{ Hz}, \qquad N/2 < k < N.$$

3.7.2 The Short-time Fourier transform

TODO (for lecture 7, week 4)

3.7.3 The FFT

When computing the numerical DFT of an N-point data vector, it is easy to see that the formula in (3.10) requires about N multiplications per DFT point, that is, a total of N^2 multiplications. A computational cost that is quadratic in the size of the input is usually bad news for practical applications but fortunately many algorithms have been developed over the years that significantly reduce this load so that, in general, a fast Fourier Transform implementation (FFT) requires on the order of $N \log N$ operations. For a data vector of, say, ten thousand samples (which represents less than a quarter of a second's worth of DVD-quality mono audio data), an FFT requires approximately forty thousand multiplications; compare this to the more than a hundred million operations required out by a naive DFT implementation: this is a cost reduction of *four orders of magnitude* and these savings are what make Fourier analysis a practical rather than a theoretical tool in today's digital devices.

The idea behind all FFT algorithms is to decompose the full DFT computation into a series of smaller DFTs; if the cost of "reassembling back the pieces" is linear in N, then the total number of operations will be less than N^2. This *divide et impera*[5] approach is in fact very general and represents the key ingredient of many other famous recursive algorithms, such as Quicksort or Mergesort. Assume we have a problem of size N, with N even, whose solution requires approximately N^2 operations; if we split the problem into two subproblems of size $N/2$, requiring $(N/2)^2$ operations each, and if merging the results requires approximately N operations, the total cost will be

$$2(N/2)^2 + N = N^2/2 + N$$

which is less than N^2 as soon as $N > 4$. Of course this subdivision can be repeated recursively $\log_2 N$ times, yielding a total cost for the solution of the problem proportional to $N \log_2 N$ operations.

Decimation in time. As a simple illustration of one of the most famous FFT implementations, consider a data size equal to a power of two: $N = 2^L$. Split the input and the output of the DFT like so:

[5]"Divide and conquer," a staple strategy of Julius Cæsar's...

- split the input into even- and odd-indexed signals, of length $N/2$ each:

$$x_e[n] = x[2n] \qquad n = 0, 1, \ldots, N/2 - 1$$
$$x_o[n] = x[2n+1] \qquad n = 0, 1, \ldots, N/2 - 1$$

- split the output into two successive halves, of length $N/2$ each:

$$X_a[k] = X[k] \qquad k = 0, 1, \ldots, N/2 - 1$$
$$X_b[k] = X[k + N/2] \qquad k = 0, 1, \ldots, N/2 - 1$$

We can now split the DFT sum into even and odd terms and write

$$\begin{aligned} X[k] &= \sum_{n=0}^{N/2-1} x[2n]e^{j\frac{2\pi}{N}2nk} + x[2n+1]e^{j\frac{2\pi}{N}(2n+1)k} \\ &= \sum_{n=0}^{N/2-1} x_e[n]e^{j\frac{2\pi}{N/2}nk} + e^{j\frac{2\pi}{N}k}x_o[n]e^{j\frac{2\pi}{N/2}nk} \\ &= X_e[k] + e^{j\frac{2\pi}{N}k}X_o[k] \end{aligned}$$

where $\mathbf{X}_e$ and $\mathbf{X}_o$ are the two $(N/2)$-point DFTs of $\mathbf{x}_e$ and $\mathbf{x}_o$ respectively. So far, there are no computational savings since each term in the summation requires 3 multiplications for a total of $N(3N/2) \approx N^2$ operations. But now consider the inherent periodicity of the DFT coefficients, as shown in 3.5; since $X_{e,o}[k + N/2] = X_{e,o}[k]$ we can compute the first and second halves of the *full* DFT as:

$$\begin{aligned} X_a[k] &= X[k] = X_e[k] + e^{j\frac{2\pi}{N}k}X_o[k] \\ X_b[k] &= X[k + N/2] = X_e[k + N/2] + e^{j\frac{2\pi}{N}(k+N/2)}X_o[k + N/2] \\ &= X_e[k] - e^{j\frac{2\pi}{N}k}X_o[k]. \end{aligned}$$

In other words, once we have computed the first $N/2$ DFT coefficients, the second half is obtained without any additional multiplication but only via a change of sign. The computational cost can be broken down like so:

1. zero multiplications for the splitting of the input sequence
2. $2(N/2)^2 = N^2/2$ multiplications for the computation of $\mathbf{X}_e$ and $\mathbf{X}_o$
3. $N/2$ multiplications by a phase terms for $\mathbf{X}_o$
4. zero multiplications for the computation of $\mathbf{X}_a$ and $\mathbf{X}_b$

for a total of $(N+1)(N/2)$ multiplication which is about half the cost of the initial problem. Of course, the splitting and merging can be performed recursively until the size of the smaller DFTs is two or four; as we have seen in Section 3.3.2, the DFT matrices of these sizes require no multiplications.