CHAPTER 1

VECTOR SPACES, SIGNALS, AND IMAGES

1.1 OVERVIEW

In this chapter we introduce the mathematical framework of vector spaces, matrices, and inner products. We motivate the mathematics by using it to model signals and images, both outside the computer (the *analog* signal as it exists in the “real world”) and inside the computer (the *digitized* signal, suitable for computer storage and processing). In either case the signal or image may be viewed as an element of a vector space, so we define and develop some essential concepts concerning these spaces. In particular, to analyze signals and images, we will decompose them into linear combinations of basic sinusoidal or complex exponential waveforms. This is the essence of the discrete Fourier and cosine transforms.

The process of sampling the analog signal and converting it to digital form causes an essential loss of information, called “aliasing” and “quantization error.” We examine these errors rather closely. Analysis of these errors motivates the development of methods to quantify the distortion introduced by an image compression technique, which leads naturally to the concept of the “energy” of a signal and a deeper analysis of inner products and orthogonality.

1.2 SOME COMMON IMAGE PROCESSING PROBLEMS

To open this chapter, we discuss a few common challenges in image processing, to try to give the reader some perspective on the subject and why mathematics is such

Discrete Fourier Analysis and Wavelets: Applications to Signal and Image Processing
By S. Allen Broughton and Kurt Bryan

an essential tool. In particular, we take a very short look at the following:

- Image compression
- Image restoration and denoising
- Edge and defect detection

We also briefly discuss the "transform" paradigm that forms the basis of so much signal and image processing, and indeed, much of mathematics.

1.2.1 Applications

Compression Digitized images are everywhere. The Internet provides obvious examples, but we also work with digitized images when we take pictures, scan documents into computers, send faxes, photocopy documents, and read books on CD. Digitized images underlie video games, and soon television will go (almost) entirely digital. In each case the memory requirements for storing digitized images are an important issue. For example, in a digital camera we want to pack as many pictures as we can onto the memory card, and we've all spent too much time waiting for Web pages to load large images. Minimizing the memory requirements for digitized images is thus important, and this task is what motivates much of the mathematics in this text.

Without going into too much detail, let's calculate the memory requirement for a typical photograph taken with a digital camera. Assume that we have 24-bit color, so that one byte of memory is required for each of the red, green, and blue components of each pixel. With a 2048×1536 pixel image there will be $2048 \times 1536 \times 3 = 9,431,040$ bytes or 9 megabytes of memory required, if no compression is used. On a camera with a 64-megabyte memory card we can store seven large, gorgeous pictures. This is unacceptably few. We need to do something more sophisticated, to reduce memory requirements by a substantial factor.

However, the compression algorithm we devise cannot sacrifice significant image quality. Even casual users of digital cameras frequently enlarge and print portions of their photographs, so any degradation of the original image will rapidly become apparent. Besides, more than aesthetics may be at stake: medical images (e.g., X rays) may be compressed and stored digitally, and any corruption of the images could have disastrous consequences. The FBI has also digitized and compressed its database of fingerprints, where similar considerations apply; see [7].

At this point the reader might find it motivational to do Exercise 1.1.

Restoration Images can be of poor quality for a variety of reasons: low-quality image capture (e.g., security video cameras), blurring when the picture is taken, physical damage to an actual photo or negative, or noise contamination during the image capture process. Restoration seeks to return the image to its original quality or even "better." Some of this technology is embedded into image capture devices such as scanners. A very interesting and mathematically sophisticated area of research involves *inpainting*, in which one tries to recover missing portions of an image, perhaps because a film negative was scratched, or a photograph written on.

Edge Detection Sometimes the features of essential interest in an image are the edges, areas of sharp transition that indicate the end of one object and the start of another. Situations such as this may arise in industrial processing, for automatic detection of defects, or in automated vision and robotic manipulation.

1.2.2 Transform-Based Methods

The use of transforms is ubiquitous in mathematics. The general idea is to take a problem posed in one setting, transform to a new domain where the problem is more easily solved, then inverse transform the solution back to the original setting. For example, if you've taken a course in differential equations you may have encountered the Laplace transform, which turns linear differential equations into algebra problems that are more easily solved.

Many imaging processing procedures begin with some type of transform T that is applied to the original image. The transform T takes the image data from its original format in the "image domain" to an altered format in the "frequency domain." Operations like compression, denoising, or other restoration are sometimes more easily performed in this frequency domain. The modified frequency domain version of the image can then be converted back to the original format in the image domain by applying the inverse of T.

The transform operator T is almost always linear, and for finite-sized signals and images such linear operators are implemented with matrix algebra. The matrix approach thus constitutes a good portion of the mathematical development of the text. Other processes, such as quantization (discussed later), are nonlinear. These are usually the lossy parts of the computation; that is, they cause irreversible but (we hope!) acceptable loss of data.

1.3 SIGNALS AND IMAGES

Before beginning a general discussion of vector spaces, it will be helpful to look at a few specific examples that provide physical realizations of the mathematical objects of interest. We'll begin with one-dimensional signals, then move on to two-dimensional images.

1.3.1 Signals

A signal may be modeled as a real-valued function of a real independent variable t, which is usually time. More specifically, consider a physical process that is dependent on time. Suppose that at each time t within some interval $a \leq t \leq b$ we perform a measurement on some aspect of this process, and this measurement yields a real number that may assume any value in a given range. In this case our measurements are naturally represented by a real-valued function $x(t)$ with domain $a \leq t \leq b$. We will refer to $x(t)$ as an *analog signal*. The function $x(t)$ might represent the intensity of sound at a given location (an audio signal), the current through a wire, the speed of an object, and so on.

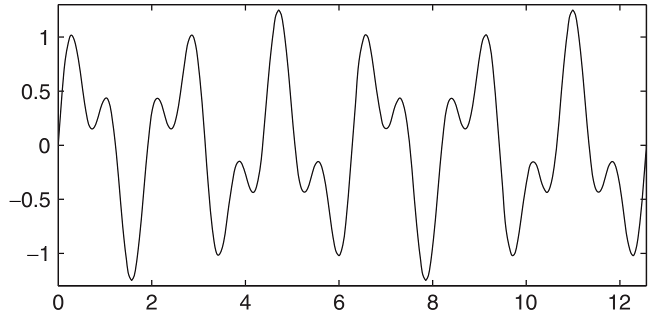

FIGURE 1.1 Analog or continuous model $x(t) = 0.75\ \sin(3t) + 0.5\ \sin(7t)$.

For example, a signal might be given by the function

$$x(t) = 0.75\ \sin(3t) + 0.5\ \sin(7t)$$

over the range $0 \leq t \leq 4\pi$. The graph of this function is shown in Figure 1.1. This signal is somewhat unrealistic, however, for it is a linear combination or superposition of a small number of simple sinusoidal functions with no noise. In general, in signal processing we can depend on being vexed by a few persistent annoyances:

- We almost never have an explicit formula for $x(t)$.
- Most signals are very complex.
- Most signals have noise.

Despite the difficulty of writing out an analog description in any specific instance, many physical processes are naturally modeled by analog signals. Analog models also have the advantage of being amenable to analysis using methods from calculus and differential equations. However, most modern signal processing takes place in computers where the computations can be done quickly and flexibly. Unfortunately, analog signals generally cannot be stored in a computer in any meaningful way.

1.3.2 Sampling, Quantization Error, and Noise

To store a signal in a computer we must first *digitize* the signal. The first step in digitization consists of measuring the signal's instantaneous value at specific times over a finite interval of interest. This process is called *sampling*. For the moment let us assume that these measurements can be carried out with "infinite precision." The process of sampling the signal converts it from an analog form to a finite list of real numbers, and is usually carried out by a piece of hardware known as an *analog-to-digital* ("A-to-D") converter.

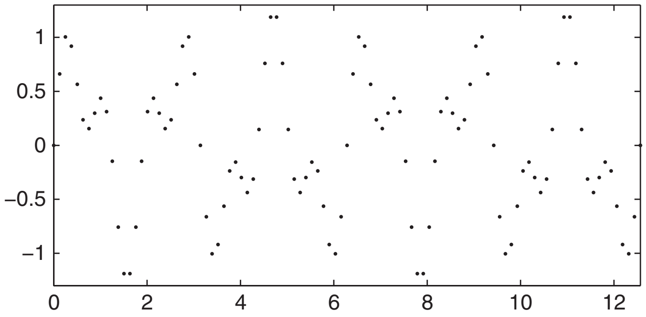

FIGURE 1.2 Discrete or sampled model, $x(t) = 0.75\ \sin(3t) + 0.5\ \sin(7t)$.

More explicitly, suppose that the signal $x(t)$ is defined on the time interval $a \leq t \leq b$. Choose an integer $N \geq 1$ and define the *sampling interval* $\Delta t = (b - a)/N$. We then measure $x(t)$ at times $t = a, a + \Delta t, a + 2\Delta t, \ldots$, to obtain samples

$$x_n = x(a + n\Delta t), \qquad n = 0, 1, \ldots, N.$$

Define

$$\mathbf{x} = (x_0, x_1, \ldots, x_N) \in \mathbb{R}^{N+1}.$$

With the given indexing $x_0 = x(a)$ and $x_N = x(b)$. The vector $\mathbf{x}$ is the sampled version of the signal $x(t)$. The quantity $1/\Delta t$ is the number of samples taken during each time period, so it is called the *sampling rate*.

In Figure 1.2 we have a graphical representation of the sampled signal from Figure 1.1. It should be intuitively clear that sampling causes a loss of information. That is, if we know only the sampled signal, then we have no idea what the underlying analog signal did between the samples. The nature of this information loss can be more carefully quantified, and this gives rise to the concept of *aliasing*, which we examine later.

The sampling of the signal in the independent variable t isn't the only source of error in our A-to-D conversion. In reality, we cannot measure the analog signal's value at any given time with infinite precision, for the computer has only a finite amount of memory. Consider, for example, an analog voltage signal that ranges from 0 to 1 volt. An A-to-D converter might divide up this one volt range into $2^8 = 256$ equally sized intervals, say with the kth interval given by $k\Delta x \leq x < (k+1)\Delta x$ where $\Delta x = 1/256$ and $0 \leq k \leq 255$. If a measurement of the analog signal at an instant in time falls within the kth interval, then the A-to-D converter might simply record the voltage at this time as $k\Delta x$. This is the *quantization* step, in which a continuously varying quantity is truncated or rounded to the nearest of a finite set of values. An A-to-D converter as above would be said to be "8-bit," because each analog measurement is converted into an 8-bit quantity. The error so introduced is called the *quantization error*.

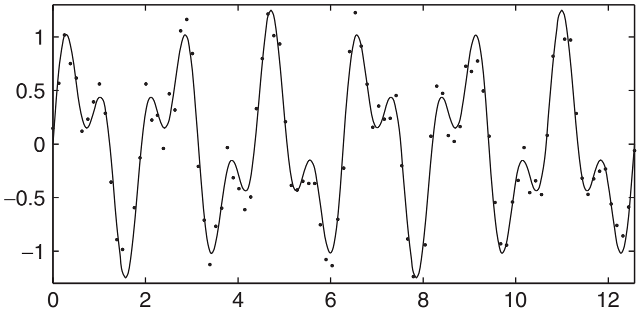

FIGURE 1.3 Analog model and discrete model with noise, $x(t) = 0.75\ \sin(3t) + 0.5\ \sin(7t)$.

Unfortunately, quantization is a nonlinear process that corrupts the algebraic structure afforded by the vector space model; see Exercise 1.6. In addition quantization introduces irreversible, though usually acceptable, loss of information. This issue is explored further in Section 1.9.

The combination of sampling and quantization allows us to *digitize* a signal or image, and thereby convert it into a form suitable for computer storage and processing.

One last source of error is random noise in the sampled signal. If the noiseless samples are given by x_n as above, the noisy sample values y_n might be modeled as

$$y_n = x_n + \epsilon_n, \tag{1.1}$$

where ϵ_n represents the noise in the nth measurement. The errors ϵ_n are usually assumed to be distributed according to some probability distribution, known or unknown. The noise model in equation (1.1) is additive; that is, the noise is merely added onto the sampled signal. Other models are possible and appropriate in some situations.

In Figure 1.3 we show a discrete signal with noise added. The analog signal and the corrupted discrete signal are graphed together so that the errors introduced by noise may be easily seen.

1.3.3 Grayscale Images

For simplicity we first consider monochrome or *grayscale* images. An analog grayscale image is modeled as a real-valued function $f(x, y)$ defined on a two-dimensional region Ω. Usually Ω is a rectangle, defined in xy coordinates by $a \le x \le b,\ c \le y \le d$. The value $f(x, y)$ represents the "intensity" of the image at the point (x, y) in Ω. Grayscale images are typically displayed visually so that smaller values of f correspond to darker shades of gray (down to black) and higher values to lighter shades (up to white).

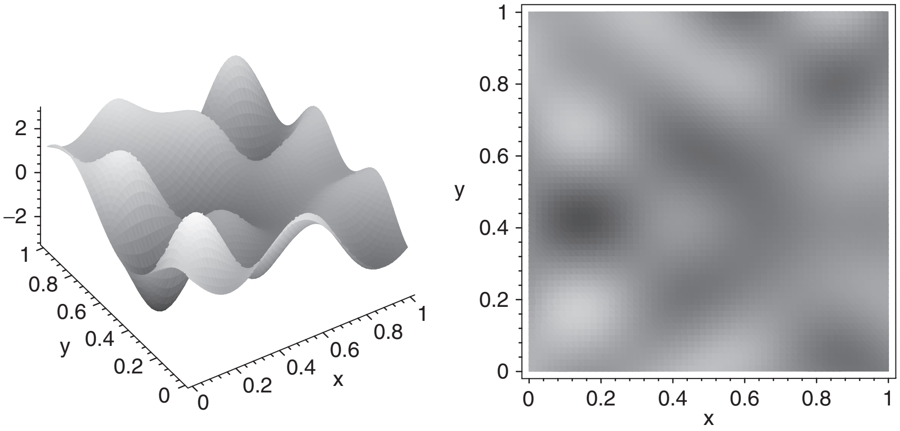

FIGURE 1.4 Grayscale image from two perspectives.

For natural images $f(x, y)$ would never be a simple function. Nonetheless, to illustrate let us consider the image defined by the function

$$\begin{aligned} f(x, y) = {} & 1.5\ \cos(2x)\cos(7y) + 0.75\ \cos(5x)\sin(3x) \\ & -1.3\ \sin(9x)\cos(15y) + 1.1\ \sin(13x)\sin(11y) \end{aligned}$$

on the domain $\Omega = \{(x, y); 0 \le x \le 1, 0 \le y \le 1\}$. The situation is illustrated in Figure 1.4. The plot on the left is a conventional $z = f(x, y)$ plot with the surface "gray-coded" according to height, where $f = -5$ corresponds to black and $f = 5$ to white. The plot on the right is the same surface but viewed from directly above, looking down the z axis. This is the actual grayscale image encoded by f according to the scheme above.

1.3.4 Sampling Images

As in the one-dimensional case an analog image must be sampled prior to storage or processing in a computer. The simplest model to adopt is the discrete model obtained by sampling the intensity function $f(x, y)$ on a regular grid of points (x_s, y_r) in the plane. For each point (x_s, y_r) the value of $f(x_s, y_r)$ is the "graylevel" or intensity at that location. The values $f(x_s, y_r)$ are collected into a $m \times n$ matrix $\mathbf{A}$ with entries a_{rs} given by

$$a_{rs} = f(x_s, y_r). \tag{1.2}$$

If you're wondering why it's $f(x_s, y_r)$ instead of $f(x_r, y_s)$, see Remark 1.1 below.

The sampling points (x_s, y_r) can be chosen in many ways. One approach is as follows: subdivide the rectangle Ω into mn identical subrectangles, with m equal vertical (y) subdivisions and n equal horizontal (x) subdivisions, of length $\Delta x = (b - a)/n$ and height $\Delta y = (d - c)/m$. We may take the points (x_s, y_r) as the centers of these subrectangles so that

$$(x_s, y_r) = \left(a + \left(s - \tfrac{1}{2}\right)\Delta x, d - \left(r - \tfrac{1}{2}\right)\Delta y\right), \tag{1.3}$$

or alternatively as the upper left corners of these rectangles,

$$(x_s, y_r) = (a + s\Delta x, d - r\Delta y), \tag{1.4}$$

where in either case $1 \le r \le m$, $1 \le s \le n$.

Note that in either case $x_1 < x_2 < \cdots < x_n$ and $y_1 > y_2 > \cdots > y_m$. Using the centers seems more natural, although the method (1.4) is a bit cleaner mathematically. For images of reasonable size, however, it will make little difference.

Remark 1.1 On a computer screen the pixel coordinates conventionally start in the upper left-hand corner and increase as we move to the right or down, which is precisely how the rows and columns of matrices are indexed. This yields a natural correspondence between the pixels on the screen and the indexes for the matrix $\mathbf{A}$ for either (1.3) or (1.4). However, this is different from the way that coordinates are usually assigned to the plane: with the matrix $\mathbf{A}$ as defined by either (1.3) or (1.4), increasing column index (the index s in a_{rs}) corresponds to the increasing x direction, but increasing row index (the index r in a_{rs}) corresponds to the *decreasing* y direction. Indeed, on those rare occasions when we actually try to identify any (x, y) point with a given pixel or matrix index, we'll take the orientation of the y axis to be reversed, with increasing y as downward.

Remark 1.2 There are other ways to model the sampling of an analog image. For example, we may take a_{rs} as some kind of integral or weighted average of f near the point (x_s, y_r). These approaches can more accurately model the physical process of sampling an analog image, but the function evaluation model in equation (1.2) has reasonable accuracy and is a simple conceptual model. For almost all of our work in this text we will assume that the sampling has been done and the input image matrix or signal is already in hand.

Remark 1.3 The values of m and n are often decided by the application in mind, or perhaps storage restrictions. It is useful and commonplace to have both m and n to be divisible by some high power of 2.

1.3.5 Color

There are a variety of approaches to modeling color images. One of the simplest is the "RGB" (red, green, blue) model in which a color image is described using three functions $r(x, y)$, $g(x, y)$, and $b(x, y)$, appropriately scaled, that correspond to the intensities of these three additive primary colors at the point (x, y) in the image domain. For example, if the color components are each scaled in the range 0 to 1, then $r = g = b = 1$ (equal amounts of all colors, full intensity) at a given point in the image would correspond to pure white, while $r = g = b = 0$ is black. The choice $r = 1$, $g = b = 0$ would be pure red, and so on. See [19] for a general discussion of the theory of color perception and other models of color such as HSI (hue, saturation, intensity) and CMY (cyan, magenta, yellow) models.

For simplicity's sake we are only going to consider grayscale and RGB models, given that computer screens are based on RGB. In fact we will be working almost exclusively with grayscale images in order to keep the discussion simple and focused on the mathematical essentials. An example where the CMY model needs to be considered is in color laser printers that use cyan, magenta, and yellow toner. The printer software automatically makes the translation from RGB to CMY. It is worth noting that the actual JPEG compression standard specifies color images with a slightly different scheme, the luminance-chrominance or "YCbCr" scheme. Images can easily be converted back and forth from this scheme to RGB.

When we consider RGB images, we will assume the sampling has already been done at points (x_s, y_r) as described above for grayscale images. In the sampled image at a given pixel location on the display device the three colors are mixed according to the intensities $r(x_s, y_r)$, $g(x_s, y_r)$, and $b(x_s, y_r)$ to produce the desired color. Thus a sampled m by n pixel image consists of three m by n arrays, one array for each color component.

1.3.6 Quantization and Noise for Images

Just as for one-dimensional signals, quantization error is introduced when an image is digitized. In general, we will structure our grayscale images so that each pixel is assigned an integer value from 0 to 255 (2^8 values) and displayed with 0 as black, 255 as white, and intermediate values as shades of gray. The range is thus quantized with 8-bit precision. Similarly each color component in an RGB image will be assigned value in the 0 to 255 range, so each pixel needs three bytes to determine its color. Some applications require more than 8-bit quantization. For example, medical images are often 12-bit grayscale, offering 4096 shades of gray.

Like one-dimensional signals, images may have noise. For example, let $\mathbf{T}$ be the matrix with entries t_{sr} representing the image on the left in Figure 1.5 and let $\mathbf{A}$ be the matrix with entries a_{sr} representing the noisy image, shown on the right. Analogous to audio signals we can posit an additive noise model

$$a_{sr} = t_{sr} + \epsilon_{sr}, \tag{1.5}$$

where $\mathbf{E}$ has entries ϵ_{sr} and represents the noise. The visual effect is to give the image a kind of "grainy" appearance.

1.4 VECTOR SPACE MODELS FOR SIGNALS AND IMAGES

We now develop a natural mathematical framework for signal and image analysis. At the core of this framework lies the concept of a *vector space*.

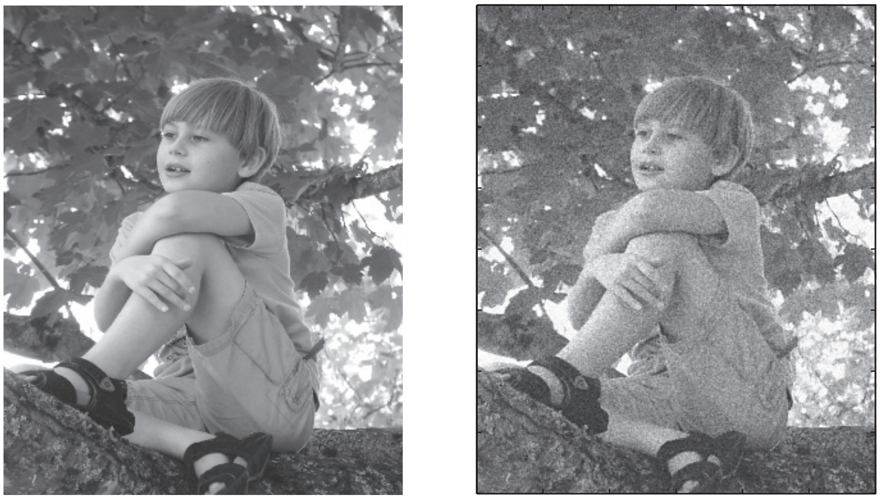

FIGURE 1.5 Image without and with additive noise.

Definition 1.4.1 *A vector space over the real numbers* $\mathbb{R}$ *is a set* V *with two operations, vector addition and scalar multiplication, with the properties that*

1. *for all vectors* $\mathbf{u}, \mathbf{v} \in V$ *the vector sum* $\mathbf{u} + \mathbf{v}$ *is defined and lies in* V *(closure under addition);*
2. *for all* $\mathbf{u} \in V$ *and scalars* $a \in \mathbb{R}$ *the scalar multiple* $a\mathbf{u}$ *is defined and lies in* V *(closure under scalar multiplication);*
3. *the "familiar" rules of arithmetic apply, specifically, for all scalars* a, b *and* $\mathbf{u}, \mathbf{v}, \mathbf{w} \in V$*:*
   a. $\mathbf{u} + \mathbf{v} = \mathbf{v} + \mathbf{u}$, *(addition is commutative),*
   b. $(\mathbf{u} + \mathbf{v}) + \mathbf{w} = \mathbf{u} + (\mathbf{v} + \mathbf{w})$ *(addition is associative),*
   c. there is a "zero vector" $\mathbf{0}$ *such that* $\mathbf{u} + \mathbf{0} = \mathbf{0} + \mathbf{u} = \mathbf{u}$ *(additive identity),*
   d. for each $\mathbf{u} \in V$ *there is an additive inverse vector* $\mathbf{w}$ *such that* $\mathbf{u} + \mathbf{w} = \mathbf{0}$*, we conventionally write* $-\mathbf{u}$ *for the additive inverse of* $\mathbf{u}$*,*
   e. $(ab)\mathbf{u} = a(b\mathbf{u})$,
   f. $(a + b)\mathbf{u} = a\mathbf{u} + b\mathbf{u}$,
   g. $a(\mathbf{u} + \mathbf{v}) = a\mathbf{u} + a\mathbf{v}$,
   h. $1\mathbf{u} = \mathbf{u}$.

If we replace $\mathbb{R}$ above by the field of complex numbers $\mathbb{C}$, then we obtain the definition of a *vector space over the complex numbers*.

We'll also make frequent use of *subspaces*:

Definition 1.4.2 *A nonempty subset W of a vector space V is called a "subspace" of V if W is itself closed under addition and scalar multiplication (as defined for V).*

Let's look at a few examples of vector spaces and subspaces, especially those useful in signal and image processing.

1.4.1 Examples—Discrete Spaces

We'll first consider examples appropriate for sampled signals or images.

■ **EXAMPLE 1.1**

The vector space $\mathbb{R}^N$ consists of vectors $\mathbf{x}$ of the form

$$\mathbf{x} = (x_1, x_2, \ldots, x_N), \tag{1.6}$$

where the x_k are all real numbers. Vector addition and scalar multiplication are defined component by component as

$$\mathbf{x} + \mathbf{y} = (x_1 + y_1, x_2 + y_2, \ldots, x_N + y_N), \quad c\mathbf{x} = (cx_1, cx_2, \ldots, cx_N),$$

where $\mathbf{y} = (y_1, y_2, \ldots, y_N)$ and $c \in \mathbb{R}$. The space $\mathbb{R}^N$ is appropriate when we work with sampled audio or other one-dimensional signals. If we allow the x_k in (1.6) and scalar c to be complex numbers, then we obtain the vector space $\mathbb{C}^N$. That $\mathbb{R}^N$ or $\mathbb{C}^N$ satisfy the properties of a vector space (with addition and scalar multiplication as defined) follows easily, with zero vector $\mathbf{0} = (0, 0, \ldots, 0)$ and additive inverse $(-x_1, -x_2, \ldots, -x_n)$ for any vector $\mathbf{x}$.

As we'll see, use of the space $\mathbb{C}^N$ can simplify much analysis, even when the signals we work with are real-valued.

Remark 1.4 *Warning*: In later work we will almost always find it convenient to index the components of vectors in $\mathbb{R}^N$ or $\mathbb{C}^N$ starting with index 0, that is, as $\mathbf{x} = (x_0, x_1, \ldots, x_{N-1})$, rather than the more traditional range 1 to N.

■ **EXAMPLE 1.2**

The sets $M_{m,n}(\mathbb{R})$ or $M_{m,n}(\mathbb{C})$, $m \times n$ matrices with real or complex entries respectively, form vector spaces. Addition is defined in the usual way for matrices, entry by entry, as is multiplication by scalars. The vector $\mathbf{0}$ is just the matrix with all zero entries, and the additive inverse for a matrix $\mathbf{M}$ with entries m_{jk} is the matrix with entries $-m_{jk}$. Any multiplicative properties of matrices are irrelevant in this context. On closer examination it should be clear that these vector spaces are nothing more than $\mathbb{R}^{mn}$ or $\mathbb{C}^{mn}$, but spaces where we choose to display the "vectors" as m rows of n components rather than a single row or column with mn entries.

The vector space $M_{m,n}(\mathbb{R})$ is an appropriate model for the discretization of images on a rectangle. As in the one-dimensional case, analysis of images is often facilitated by viewing them as members of space $M_{m,n}(\mathbb{C})$.

■ EXAMPLE 1.3

On occasion it is useful to think of an analog signal $f(t)$ as beginning at some time $t = a$ and continuing "indefinitely." If we sample such a signal at intervals of Δt starting at time $t = a$ without stopping, we obtain a vector

$$\mathbf{x} = (x_0, x_1, x_2, \ldots) \tag{1.7}$$

with real components $x_k = f(a + k\Delta t)$, $k \geq 0$. Given another vector $\mathbf{y} = (y_0, y_1, y_2, \ldots)$, we define vector addition and scalar multiplication as

$$c\mathbf{x} = (cx_0, cx_1, cx_2, \ldots), \quad \mathbf{x} + \mathbf{y} = (x_0 + y_0, x_1 + y_1, \ldots).$$

Let V denote the resulting set with these operations. It's an easy algebra problem to verify that V is a vector space over the real numbers with zero vector $\mathbf{0} = (0, 0, 0, \ldots)$; the additive inverse of $\mathbf{x}$ above is $(-x_0, -x_1, -x_2, \ldots)$. And though it may seem painfully obvious, to say that "$\mathbf{x} = \mathbf{y}$" in V means precisely that $x_k = y_k$ for each $k \geq 0$. We will later encounter vector spaces where we have to be quite careful about what is meant by "$\mathbf{x} = \mathbf{y}$."

A simple variant of this vector space is the bi-infinite space of vectors

$$\mathbf{x} = (\ldots, x_{-2}, x_{-1}, x_0, x_1, x_2, \ldots) \tag{1.8}$$

with the analogous vector space structure. A space like this would be appropriate for modeling a physical process with a past and future of indefinite duration.

■ EXAMPLE 1.4

As defined, the set V in the previous example lacks sufficient structure for the kinds of analysis we usually want to do, so we typically impose additional conditions on the components of $\mathbf{x}$. For example, let us impose the additional condition that for each $\mathbf{x}$ as defined by equation (1.7) there is some number M (which may depend on $\mathbf{x}$) such that $|x_k| \leq M$ for all $k \geq 0$. In this case the resulting set (with addition and scalar multiplication as defined above for V) is a vector space called $L^\infty(\mathbb{N})$ (here $\mathbb{N} = \{0, 1, 2, \ldots\}$ denotes the set of natural numbers), or often just ℓ^∞. This would be an appropriate space for analyzing the class of sampled signals in which the magnitude of any particular signal remains bounded for all $t \geq 0$.

The verification that $L^\infty(\mathbb{N})$ is a vector space over $\mathbb{R}$ is fairly straightforward. The algebraic properties of item 3 in Definition 1.4.1 are verified exactly as for V in the previous example, where again the zero vector is $(0, 0, 0, \ldots)$ and the additive inverse of $\mathbf{x}$ is $(-x_0, -x_1, -x_2, \ldots)$. To show closure under vector addition, consider vectors $\mathbf{x}$ and $\mathbf{y}$ with $|x_k| \leq M_x$ and $|y_k| \leq M_y$ for all $k \geq 0$. From the triangle inequality for real numbers

$$|x_k + y_k| \leq |x_k| + |y_k| \leq M_x + M_y,$$

so the components of $\mathbf{x} + \mathbf{y}$ are bounded in magnitude by $M_x + M_y$. Thus $\mathbf{x} + \mathbf{y} \in L^\infty(\mathbb{N})$, and the set is closed under addition. Similarly for any k the kth component cx_k of $c\mathbf{x}$ is bounded by $|c|M_x$, and the set is closed under scalar multiplication. This makes $L^\infty(\mathbb{N})$ a subspace of the vector space V from the previous example.

If we consider bi-infinite vectors as defined by equation (1.8) with the condition that for each $\mathbf{x}$ there is some number M such that $|x_k| \leq M$ for all $k \in \mathbf{Z}$, then we obtain the vector space $L^\infty(\mathbb{Z})$.

■ EXAMPLE 1.5

We may impose the condition that for each sequence of real numbers $\mathbf{x}$ of the form in (1.7) we have

$$\sum_{k=0}^{\infty} |x_k|^2 < \infty, \tag{1.9}$$

in which case the resulting set is called $L^2(\mathbb{N})$, or often just ℓ^2. This is even more stringent than the condition for $L^\infty(\mathbb{N})$; verification of this assertion and that $L^2(\mathbb{N})$ is a vector space is left for Exercise 1.11. We may also let the components x_k be complex numbers, and the result is still a vector space.

Conditions like (1.9) that bound the “squared value” of some object are common in applied mathematics and usually correspond to finite energy in an underlying physical process.

A very common variant of $L^2(\mathbb{N})$ is the space $L^2(\mathbb{Z})$, consisting of vectors of the form in equation (1.8) that satisfy

$$\sum_{k=-\infty}^{\infty} |x_k|^2 < \infty.$$

The space $L^2(\mathbb{Z})$ will play an important role throughout Chapters 6 and 7.

Variations on the spaces above are possible, and common. Which vector space we work in depends on our model of the underlying physical process and the analysis we hope to carry out.

1.4.2 Examples—Function Spaces

In the examples above the spaces all consist of vectors that are lists or arrays, finite or infinite, of real or complex numbers. Functions can also be interpreted as elements of vector spaces, and this is the appropriate setting when dealing with analog signals or images. The mathematics in this case can be more complicated, especially when dealing with issues concerning approximation, limits, and convergence (about which we've said little so far). We'll have limited need to work in this setting, at least until Chapter 7. Here are some relevant examples.

■ EXAMPLE 1.6

Consider the set of all real-valued functions f that are defined and continuous at every point in a closed interval $[a, b]$ of the real line. This means that for any $t_0 \in [a, b]$,

$$\lim_{t \to t_0} f(t) = f(t_0),$$

where t approaches from the right only in the case that $t_0 = a$ and from the left only in the case that $t_0 = b$. The sum $f + g$ of two functions f and g is the function defined by $(f + g)(t) = f(t) + g(t)$, and the scalar multiple cf is defined via $(cf)(t) = cf(t)$. With these operations this is a vector space over $\mathbb{R}$, for it is closed under addition since the sum of two continuous functions is continuous. It is also closed under scalar multiplication, since a scalar multiple of a continuous function is continuous. The algebraic properties of item 3 in Definition 1.4.1 are easily verified with the "zero function" as the additive identity and $-f$ as the additive inverse of f. The resulting space is denoted $C[a, b]$.

The closed interval $[a, b]$ can be replaced by the open interval (a, b) to obtain the vector space $C(a, b)$. The spaces $C[a, b]$ and $C(a, b)$ do not coincide, for example, $f(t) = 1/t$ lies in $C(0, 1)$ but not $C[0, 1]$. In this case $1/t$ isn't defined at $t = 0$, and moreover this function can't even be extended to $t = 0$ in a continuous manner.

■ EXAMPLE 1.7

Consider the set of all real-valued functions f that are piecewise continuous on the interval $[a, b]$; that is, f is defined and continuous at all but finitely many points in $[a, b]$. With addition and scalar multiplication as defined in the last example this is a vector space over $\mathbb{R}$. The requisite algebraic properties are verified in precisely the same manner. To show closure under addition, just note that any point of discontinuity for $f + g$ must be a point of discontinuity for f or g; hence $f + g$ can have only finitely many points of discontinuity. The discontinuities for cf are precisely those for f.

Both $C(a, b)$ and $C[a, b]$ are subspaces of this vector space (which doesn't have any standard name).

■ EXAMPLE 1.8

Let V denote those functions f in $C(a, b)$ for which

$$\int_a^b f^2(t)\,dt < \infty. \tag{1.10}$$

A function f that is continuous on (a, b) can have no vertical asymptotes in the open interval, but may be unbounded as t approaches the endpoint $t = a$ or $t = b$. Thus the integral above (and all integrals in this example) should be interpreted as improper integrals, that is,

$$\int_a^b f^2(t)\,dt = \lim_{p\to a^+} \int_p^r f^2(t)\,dt + \lim_{q\to b^-} \int_r^b f^2(t)\,dt,$$

where r is any point in (a, b).

To show that V is closed under scalar multiplication, note that

$$\int_a^b (cf)^2(t)\,dt = c^2 \int_a^b f^2(t)\,dt < \infty,$$

since f satisfies the inequality (1.10). To show closure under vector addition, first note that for any real numbers p and q, we have $(p + q)^2 \leq 2p^2 + 2q^2$ (this follows easily from $0 \leq (p - q)^2$). As a consequence, for any two functions f and g in V and any $t \in (a, b)$, we have

$$(f(t) + g(t))^2 \leq 2f^2(t) + 2g^2(t).$$

Integrate both sides above from $t = a$ to $t = b$ (as improper integrals) to obtain

$$\int_a^b (f(t) + g(t))^2\,dt \leq 2\int_a^b f^2(t)\,dt + 2\int_a^b g^2(t)\,dt < \infty,$$

so $f + g$ is in V. The algebraic properties in Definition 1.4.1 follow as before, so that V is a vector space over $\mathbb{R}$.

The space V as defined above doesn't have any standard name, but it is "almost" the vector space commonly termed $L^2(a, b)$, also called "the space of square integrable functions on (a, b)." More precisely, the space defined above is the intersection $C(a, b) \cap L^2(a, b)$. Nonetheless, we will generally refer to it as "$L^2(a, b)$," and say more about it in Section 1.10.4. This space will make more appearances in the text, especially in Chapter 7.

Similar to the inequality (1.9), the condition (1.10) comes up fairly often in applied mathematics and usually corresponds to signals of finite energy.

■ EXAMPLE 1.9

Consider the set of functions $f(x, y)$ defined on some rectangular region $\Omega = \{(x, y); a \leq x \leq b, c \leq y \leq d\}$. We make no particular assumptions about the continuity or other nature of the functions. Addition and scalar multiplication are defined in the usual way, as $(f + g)(x, y) = f(x, y) + g(x, y)$ and $(cf)(x, y) = cf(x, y)$. This is a vector space over $\mathbb{R}$. The proof is in fact the same as in the case of functions of a single variable. This space would be useful for image analysis, with the functions representing graylevel intensities and Ω the image domain.

Of course, we can narrow the class of functions, for example, by considering only those that are continuous on Ω; this space is denoted $C(\Omega)$. Or we can impose the further restriction that

$$\int_a^b \int_c^d f^2(x, y)\, dy\, dx < \infty,$$

which, in analogy to the one-dimensional case, we denote by $L^2(\Omega)$. There are many other important and potentially useful vector spaces of functions.

We could also choose the domain Ω to be infinite, for example, a half-plane or the whole plane. The region is selected to give a good tractable vector space model and to be relevant to the physical situation of interest, though unbounded domains are not generally necessary in image processing.

In addition to the eight basic arithmetic properties listed in Definition 1.4.1, certain other arithmetic properties of vector spaces are worth noting.

Proposition 1.4.1 *If V is a vector space over $\mathbb{R}$ or $\mathbb{C}$, then*

1. *the vector $\mathbf{0}$ is unique;*
2. $0\mathbf{u} = \mathbf{0}$ *for any vector* $\mathbf{u}$*;*
3. *the additive inverse of any vector* $\mathbf{u}$ *is unique, and is given by* $(-1)\mathbf{u}$.

These properties look rather obvious and are usually easy to verify in any specific vector space as in Examples 1.1 to 1.9. They also hold in any vector space, and can be shown directly from the eight arithmetic properties for a vector space. The careful proofs can be a bit tricky though! See Exercise 1.12.

We have already started to use the additive vector space structure when we modeled noise in signals and images with equations (1.1) and (1.5). The vector space structure will be indispensable when we discuss the decomposition of signals and images into linear combinations of more "basic" components.

Tables 1.1 and 1.2 give a brief summary of some of the important spaces of interest, as well as when each space might be used. As mentioned, the analog models are used when we consider the actual physical processes that underly signals and images, but for computation we always consider the discrete version.

TABLE 1.1 Discrete Signal Models and Uses

Notation	Vector Space Description
$\mathbb{R}^{\mathbb{N}}$	$\{\mathbf{x} = (x_1, \dots, x_N) : x_i \in \mathbb{R}\}$, finite sampled signals
$\mathbb{C}^{\mathbb{N}}$	$\{\mathbf{x} = (x_1, \dots, x_N) : x_i \in \mathbb{C}\}$, analysis of sampled signals
$L^\infty(\mathbb{N})$ or ℓ^∞	$\{\mathbf{x} = (x_0, x_1, \dots) : x_i \in \mathbb{R},\ \ \lvert x_i\rvert \le M \ \text{ for all } \ i \ge 0\}$ bounded, sampled signals, infinite time
$L^2(\mathbb{N})$ or ℓ^2	$\{\mathbf{x} = (x_0, x_1, \dots) : x_i \in \mathbb{R}$ or $x_i \in \mathbb{C}$, $\sum_k \lvert x_k\rvert^2 < \infty\}$ sampled signals, finite energy, infinite time
$L^2(\mathbb{Z})$	$\{\mathbf{x} = (\dots, x_{-1}, x_0, x_1, \dots) : x_i \in \mathbb{R}$ or $x_i \in \mathbb{C}$, $\sum_k \lvert x_k\rvert^2 < \infty\}$ sampled signals, finite energy, bi-infinite time
$M_{m,n}(\mathbb{R})$	Real $m \times n$ matrices, sampled rectangular images
$M_{m,n}(\mathbb{C})$	Complex $m \times n$ matrices, analysis of images

1.5 BASIC WAVEFORMS—THE ANALOG CASE

1.5.1 The One-dimensional Waveforms

To analyze signals and images, it can be extremely useful to decompose them into a sum of more elementary pieces or patterns, and then operate on the decomposed version, piece by piece. We will call these simpler pieces the *basic waveforms*. They serve as the essential building blocks for signals and images. In the context of Fourier analysis for analog signals these basic waveforms are simply sines and cosines, or equivalently, complex exponentials. Specifically, the two basic waveforms of interest are $\cos(\omega t)$ and $\sin(\omega t)$, or their complex exponential equivalent $e^{i\omega t}$, where ω acts as a frequency parameter.

The complex exponential basic waveforms will be our preferred approach. Recall Euler's identity,

$$e^{i\theta} = \cos(\theta) + i \sin(\theta). \tag{1.11}$$

TABLE 1.2 Analog Signal Models and Uses

Notation	Vector Space Description
$C(a, b)$ or $C[a, b]$	Continuous functions on (a, b) or $[a, b]$, continuous analog signal
$L^2(a, b)$	f Riemann integrable and $\int_a^b f^2(x)\, dx < \infty$, analog signal with finite energy
$L^2(\Omega)$ $(\Omega = [a, b] \times [c, d])$	f Riemann integrable and $\int_a^b \int_c^d f^2(x, y)\, dy\, dx < \infty$, analog image

From this we have (with $\theta = \omega t$ and $\theta = -\omega t$)

$$\begin{aligned} e^{i\omega t} &= \cos(\omega t) + i\ \sin(\omega t), \\ e^{-i\omega t} &= \cos(\omega t) - i\ \sin(\omega t), \end{aligned} \tag{1.12}$$

which can be solved for $\cos(\omega t)$ and $\sin(\omega t)$ as

$$\begin{aligned} \cos(\omega t) &= \frac{e^{i\omega t} + e^{-i\omega t}}{2}, \\ \sin(\omega t) &= \frac{e^{i\omega t} - e^{-i\omega t}}{2i}. \end{aligned} \tag{1.13}$$

If we can decompose a given signal $x(t)$ into a linear combination of waveforms $\cos(\omega t)$ and $\sin(\omega t)$, then equations (1.13) make it clear that we can also decompose $x(t)$ into a linear combination of appropriate complex exponentials. Similarly equations (1.12) can be used to convert any complex exponential decomposition into sines and cosines. We thus also consider the complex exponential functions $e^{i\omega t}$ as basic waveforms.

Remark 1.5 In the real-valued sine/cosine case we only need to work with $\omega \geq 0$, since $\cos(-\omega t) = \cos(\omega t)$ and $\sin(-\omega t) = -\sin(\omega t)$. Any function that can be constructed as a sum using negative values of ω has an equivalent expression with positive ω.

■ **EXAMPLE 1.10**

Consider the signal $x(t) = \sin(t) + 3\ \sin(-2t) - 2\ \cos(-5t)$. From Remark 1.5 we can express $x(t)$ as $x(t) = \sin(t) - 3\ \sin(2t) - 2\ \cos(5t)$, using only positive values of ω in the expressions $\sin(\omega t)$ and $\cos(\omega t)$. Equations (1.13) also yield

$$x(t) = \frac{1}{2i}e^{it} - \frac{1}{2i}e^{-it} - \frac{3}{2i}e^{2it} + \frac{3}{2i}e^{-2it} - e^{5it} - e^{-5it},$$

a sum of basic complex exponential waveforms. Whether we work in trigonometric functions or complex exponentials matters little from the mathematical perspective. The trigonometric functions, because they're real-valued and familiar, have a natural appeal, but the complex exponentials often yield much cleaner mathematical formulas. As such, we will usually prefer to work with the complex exponential waveforms.

We can visualize $e^{i\omega t}$ by simultaneously graphing the real and imaginary parts as functions of t, as in Figure 1.6, with real parts solid and imaginary parts dashed. Note that

$$\begin{aligned} \cos(\omega t) &= \mathrm{Re}(e^{i\omega t}), \\ \sin(\omega t) &= \mathrm{Im}(e^{i\omega t}). \end{aligned} \tag{1.14}$$

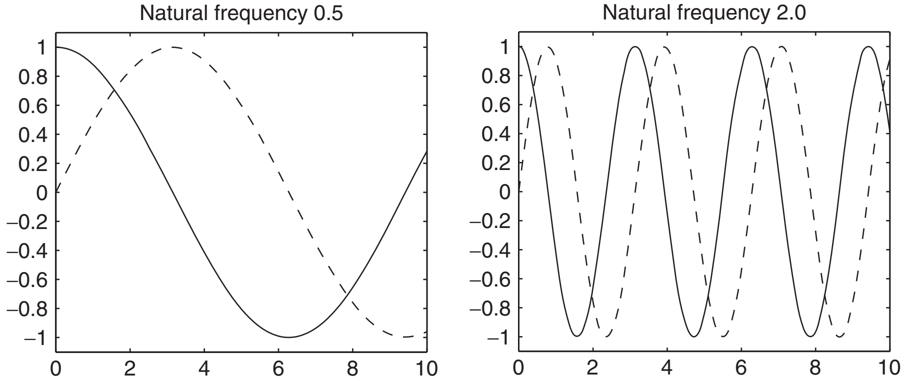

FIGURE 1.6 Real (*solid*) and imaginary (*dashed*) parts of complex exponentials.

Of course the real and imaginary parts, and $e^{i\omega t}$ itself, are periodic and ω controls the frequency of oscillation. The parameter ω is called the *natural frequency* of the waveform.

The period of $e^{i\omega t}$ can be found by considering those values of λ for which $e^{i\omega(t+\lambda)t} = e^{i\omega t}$ for all t, which yields

$$e^{i\omega(t+\lambda)t} = e^{i\omega t}e^{i\omega\lambda} = e^{i\omega t}(\cos(\omega\lambda) + i\,\sin(\omega\lambda)), \tag{1.15}$$

so that $e^{i\omega(t+\lambda)} = e^{i\omega t}$ forces $\cos(\omega\lambda) + i\,\sin(\omega\lambda) = 1$. The smallest positive value of λ for which this holds satisfies $\lambda|\omega| = 2\pi$. Thus $\lambda = 2\pi/|\omega|$, which is the *period* of $e^{i\omega t}$ (or the *wavelength* if t is a spatial variable).

The quantity $q = l/\lambda = \omega/2\pi$ (so $\omega = 2\pi q$) is the number of oscillations made by the waveform in a unit time interval and is called the *frequency* of the waveform. If t denotes time in seconds, then q has units of *Hertz*, or cycles per second. It is often useful to write the basic waveform $e^{i\omega t}$ as $e^{2\pi iqt}$, to explicitly note the frequency q of oscillation. More precisely, the frequency (in Hertz) of the waveform $e^{2\pi iqt}$ is $|q|$ Hertz, since frequency is by convention nonnegative.

In real-valued terms, we can use $\cos(2\pi qt)$ and $\sin(2\pi qt)$ with $q \geq 0$ in place of $\cos(\omega t)$ and $\sin(\omega t)$ with $\omega \geq 0$.

As we will see later, any "reasonable" (e.g., bounded and piecewise continuous) function $x(t)$ defined on an interval $[-T, T]$ can be written as an infinite sum of basic waveforms $e^{i\omega t}$, as

$$x(t) = \sum_{k=-\infty}^{\infty} c_k e^{\pi ikt/T} \tag{1.16}$$

for an appropriate choice of the constants c_k. The natural frequency parameter ω assumes the values $\pi k/T$ for $k \in \mathbb{Z}$, or equivalently the frequency q assumes values $k/2T$. An expansion analogous to (1.16) also exists using the sine/cosine waveforms.

1.5.2 2D Basic Waveforms

The 2D basic waveforms are governed by a pair of frequency parameters α and β. Let (x, y) denote coordinates in the plane. The basic waveforms are products of complex exponentials and can be written in either additive form (left side below) or a product form (right side),

$$e^{i(\alpha x+\beta y)} = e^{i\alpha x}e^{i\beta y}. \tag{1.17}$$

As in the one-dimensional case we can convert to a trigonometric form,

$$e^{i(\alpha x+\beta y)} = \cos(\alpha x + \beta y) + i\sin(\alpha x + \beta y),$$

and conversely,

$$\begin{aligned}\cos(\alpha x + \beta y) &= \frac{e^{i(\alpha x+\beta y)} + e^{-i(\alpha x+\beta y)}}{2},\\ \sin(\alpha x + \beta y) &= \frac{e^{i(\alpha x+\beta y)} - e^{-i(\alpha x+\beta y)}}{2i}.\end{aligned}$$

Thus the family of functions

$$\{\cos(\alpha x + \beta y), \sin(\alpha x + \beta y)\} \tag{1.18}$$

is an alternate set of basic waveforms.

Sometimes a third class of basic waveforms is useful. An application of Euler's formula to both exponentials on the right in equation (1.17) shows that

$$\begin{aligned}e^{i(\alpha x+\beta y)} &= \cos(\alpha x)\cos(\beta y) - \sin(\alpha x)\sin(\beta y)\\ &\quad + i(\cos(\alpha x)\sin(\beta y) + \sin(\alpha x)\cos(\beta y)),\end{aligned}$$

so these complex exponential basic waveforms can be expanded into linear combinations of functions from the family

$$\{\cos(\alpha x)\cos(\beta y), \sin(\alpha x)\sin(\beta y), \cos(\alpha x)\sin(\beta y), \sin(\alpha x)\cos(\beta y)\}. \tag{1.19}$$

Conversely, each of these functions can be written in terms of complex exponentials (see Exercise 1.15). The functions in (1.19) also form a basic set of waveforms.

Just as in the one-dimensional case, for the real-valued basic waveforms (1.18) or (1.19) we can limit our attention to the cases $\alpha, \beta \geq 0$.

We almost always use the complex exponential waveforms in our analysis, however, except when graphing. As in the one-dimensional case these exponential

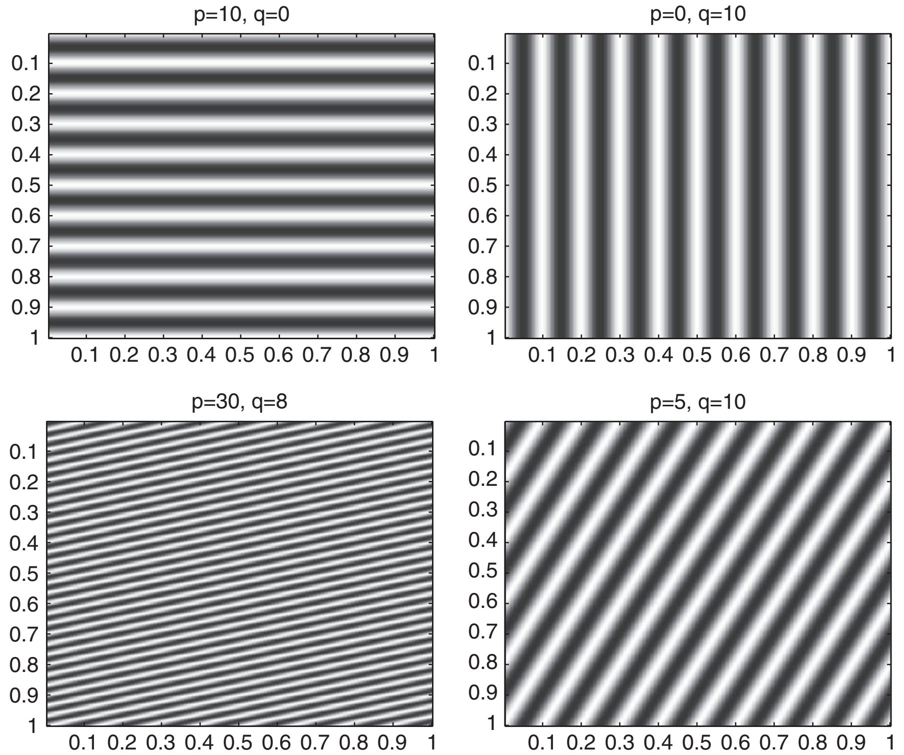

FIGURE 1.7 Grayscale image of $\cos(2\pi(px + qy)) = \mathrm{Re}(e^{2\pi i(px+qy)})$ for various p and q.

waveforms can be written in a frequency format as

$$e^{i(\alpha x+\beta y)} = e^{2\pi i(px+qy)},$$

where p and q are frequencies in the x and y directions. In the plots in Figure 1.7 we show the real parts of the basic exponential waveforms for several values of p and q, as grayscale images on the unit square $0 \leq x, y \leq 1$, with y downward as per Remark 1.1 on page 8. The waves seem to have a direction and wavelength; see Exercise 1.18.

1.6 SAMPLING AND ALIASING

1.6.1 Introduction

As remarked prior to equation (1.16), an analog signal or function on an interval $[-T, T]$ can be decomposed into a linear combination of basic waveforms $e^{i\omega t}$, or the corresponding sines and cosines. For computational purposes, however, we sample the signal and work with the corresponding discrete quantity, a vector. The

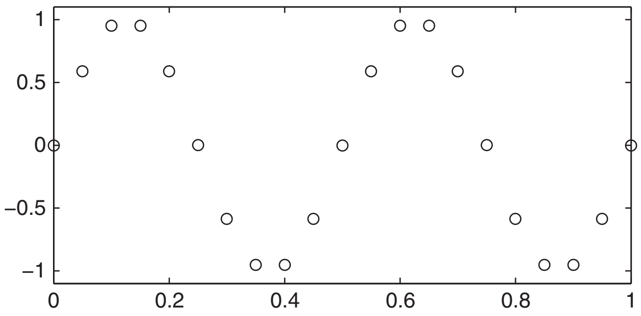

FIGURE 1.8 Sampled signal, $\Delta T = 0.05$.

analog waveforms $e^{i\omega t}$ must then be replaced with "equivalent" discrete waveforms. What should these waveforms be?

The obvious answer is to use the sampled analog waveforms, but an interesting phenomenon called *aliasing* shows up. It should be intuitively clear that sampling destroys information about the analog signal. It is in the case where the sampling is done on basic waveforms that this loss of information is especially easy to quantify. We will thus take a short detour to discuss aliasing, and then proceed to the discrete model waveforms in the next section.

To illustrate aliasing, consider the sampled signal graphed in Figure 1.8, obtained by sampling the basic waveform $\sin(\omega t)$ for some "unknown" ω on the interval $0 \leq t \leq 1$ at intervals of $\Delta T = 0.05$. The sampled waveform appears to make exactly two full cycles in the time interval [0, 1], corresponding to a frequency of two Hertz and the basic waveform $\sin(4\pi t)$. However, Figure 1.9 shows a plot of the "true" analog signal, superimposed on the sampled signal!

The actual analog waveform is $x(t) = \sin(44\pi t)$, corresponding to a frequency of 22 Hertz ($\omega = 44\pi$). The plot of the sampled signal is quite deceptive and

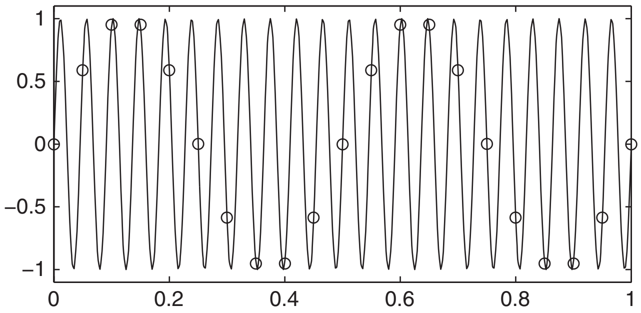

FIGURE 1.9 Analog and sampled signal, $\Delta T = 0.05$.

illustrates *aliasing*, in which sampling destroys our ability to distinguish between basic waveforms with certain relative frequencies.

1.6.2 Aliasing for Complex Exponential Waveforms

To quantify this phenomenon, let's first look at the situation for complex waveforms $e^{i\omega t}$. For simplicity we write these in frequency form $e^{2\pi iqt}$ with $q = \omega/2\pi$, so q is in Hertz; note q is not required to be an integer. Suppose that we sample such a waveform N times per second, at times $t = k/N$ for $k = 0, 1, 2, \ldots$. The sampled waveform yields values $e^{2\pi iqk/N}$. Under what circumstances will another analog waveform $e^{2\pi i\tilde{q}t}$ at frequency $\tilde{q}$ Hertz yield the same sampled values at times $t = k/N$? Stated quantitatively, this means that

$$e^{2\pi iqk/N} = e^{2\pi i\tilde{q}k/N}$$

for all integers $k \geq 0$. Divide both sides above by $e^{2\pi iqk/N}$ to obtain $1 = e^{2\pi i(\tilde{q}-q)k/N}$, or equivalently,

$$1 = \left(e^{2\pi i(\tilde{q}-q)/N}\right)^k \tag{1.20}$$

for all $k \geq 0$. Now if a complex number z satisfies $z^k = 1$ for all integers k then $z = 1$ (consider the case $k = 1$). From equation (1.20) we conclude that $e^{2\pi i(\tilde{q}-q)/N} = 1$. Since $e^x = 1$ only when $x = 2\pi im$ where $m \in \mathbb{Z}$, it follows that $2\pi i(\tilde{q} - q)/N = 2\pi im$, or $\tilde{q} - q = mN$. Thus the waveforms $e^{2\pi iqt}$ and $e^{2\pi i\tilde{q}t}$ sampled at times $t = k/N$ yield identical values exactly when

$$\tilde{q} - q = mN \tag{1.21}$$

for some integer m.

Equation (1.21) quantifies the phenomenon of aliasing: When sampled with sampling interval $\Delta T = 1/N$ (frequency N Hertz) the two waveforms $e^{2\pi iqt}$ and $e^{2\pi i\tilde{q}t}$ will be *aliased* (yield the same sampled values) whenever $\tilde{q}$ and q differ by any multiple of the sampling rate N. Equivalently, $e^{i\omega t}$ and $e^{i\tilde{\omega}t}$ yield exactly the same sampled values when $\tilde{\omega} - \omega = 2\pi mN$.

Aliasing has two implications, one "physical" and one "mathematical." The physical implication is that if an analog signal consists of a superposition of basic waveforms $e^{2\pi iqt}$ and is sampled at N samples per second, then for any particular frequency q_0 the waveforms

$$\ldots, e^{2\pi i(q_0-2N)t}, e^{2\pi i(q_0-N)t}, e^{2\pi iq_0t}, e^{2\pi i(q_0+N)t}, e^{2\pi i(q_0+2N)t} \ldots$$

are all aliased. Any information concerning their individual characteristics (amplitudes and phases) is lost. The only exception is if we know a priori that the signal consists only of waveforms in a specific and sufficiently small frequency range. For example, if we know that the signal consists only of waveforms $e^{2\pi iqt}$ with $-N/2 < q \leq N/2$ (i.e., frequencies $|q|$ between 0 and $N/2$), then no aliasing will occur because $q \pm N$, $q \pm 2N$, and so on, do not lie in this range. This might be the case if the signal has been *low-pass filtered* prior to being sampled, to remove (by analog means) all frequencies greater than $N/2$. In this case sampling at frequency N would produce no aliasing.

The mathematical implication of aliasing is this: when analyzing a signal sampled at frequency N, we need only use the sampled waveforms $e^{2\pi iqk/N}$ with $-N/2 < q \leq N/2$. Any discrete basic waveform with frequency outside this range is aliased with, and hence identical to, a basic waveform within this range.

1.6.3 Aliasing for Sines and Cosines

Similar considerations apply when using the sine/cosine waveforms. From equations (1.13) it's easy to see that when sampled at frequency N the functions $\sin(2\pi qt)$ or $\cos(2\pi qt)$ will be aliased with waveforms $\sin(2\pi\tilde{q}t)$ or $\cos(2\pi\tilde{q}t)$ if $\tilde{q} - q = mN$ for any integer m. Indeed, one can see directly that if $\tilde{q} = q + mN$, then

$$\sin(2\pi\tilde{q}k/N) = \sin(2\pi(q + mN)k/N) = \sin(2\pi qk/N + 2\pi km) = \sin(2\pi qk/N),$$

since $\sin(t + 2\pi km) = \sin(t)$ for any t, where k and m are integers. A similar computation holds for the cosine. Thus as in the complex exponential case we may restrict our attention to a frequency interval in q of length N, for example, $-N/2 < q \leq N/2$.

However, in the case of the sine/cosine waveforms the range for q (or ω) can be narrowed a bit further. In light of Remark 1.5 on page 18 we need not consider $q < 0$ if our main interest is the decomposition of a signal into a superposition of sine or cosine waveforms, for $\cos(-2\pi qt) = \cos(2\pi qt)$ and $\sin(-2\pi qt) = -\sin(2\pi qt)$. In the case of the sine/cosine waveforms we need only consider the range $0 \leq q \leq N/2$. This is actually identical to the restriction for the complex exponential case, where the frequency $|q|$ of $e^{2\pi iqt}$ is restricted to $0 \leq |q| \leq N/2$.

1.6.4 The Nyquist Sampling Rate

For both complex exponential waveforms $e^{2\pi iqt}$ and the basic trigonometric waveforms $\sin(2\pi qt)$, $\cos(2\pi qt)$, sampling at N samples per second results in frequencies greater than $N/2$ being aliased with frequencies between 0 and $N/2$. Thus, if an analog signal is known to contain only frequencies of magnitude F and lower, sampling at a frequency $N \geq 2F$ (so $F \leq N/2$) results in no aliasing. This is one form of what is called the *Nyquist sampling rate* or *Nyquist sampling criterion*: to avoid aliasing, we must sample at twice the highest frequency present in an analog signal. This means at least two samples per cycle for the highest frequency. One typically samples at a slightly greater rate to ensure greater fidelity. Thus commercial CD's use a sample rate of 44.1 kHz, which is slightly greater than the generally accepted maximum audible frequency of 20 kHz. A CD for dogs would require a higher sampling rate!

A closely related result, the Shannon sampling theorem, states that if an analog signal $x(t)$ contains only frequencies in the range 0 to $N/2$ and is sampled at sampling rate N, then $x(t)$ can be perfectly recovered for all t; see [17, p. 87].

1.6.5 Aliasing in Images

Aliasing also occurs when images are sampled. Consider the simple grayscale image embodied by the function

$$f(x, y) = 256\sin(2\pi(50x + 70y))$$

on the domain $0 \leq x, y \leq 1$. In Figure 1.10 are images based on sampling f on n by n grids for $n = 60, 100, 300$, and 1000, and displayed with 0 as black, 255 as white (rounded down). To avoid aliasing, we expect to need a sampling frequency of at least 100 samples per unit distance in the x direction, 140 in the y. The 1000 by 1000 image comes closest to the "true" analog image, while the $n = 60$ and $n = 100$ images completely misrepresent the nature of the underlying analog signal. When $n = 60$, the stripes actually go the wrong way.

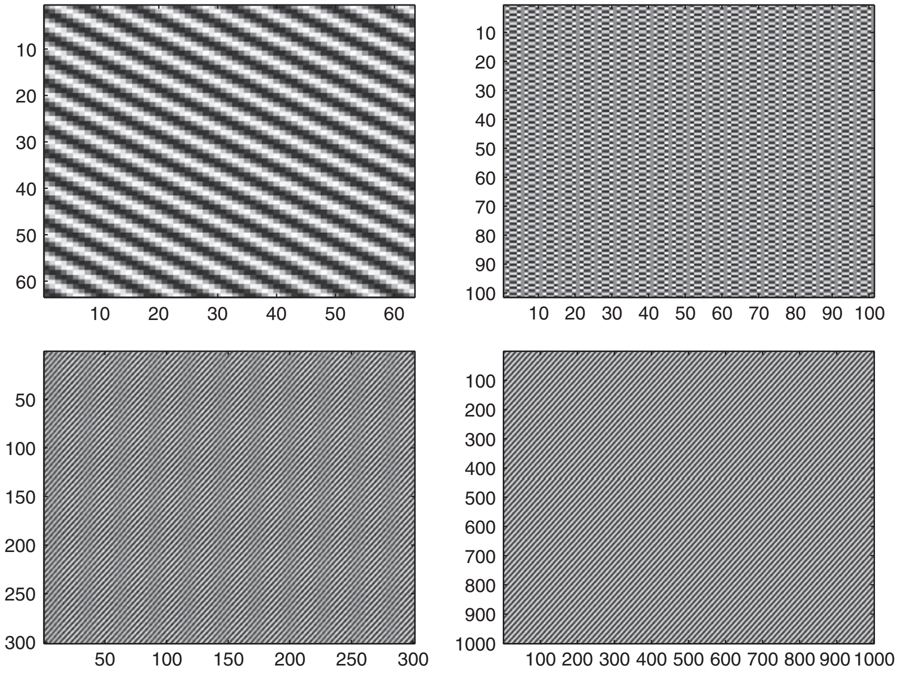

FIGURE 1.10 Aliasing in a 2D region, $n = 60$ (*top left*), $n = 100$ (*top right*), $n = 300$ (*bottom left*), and $n = 1000$ (*bottom right*).

In general, however, aliasing in images is difficult to convey consistently via the printed page, or even on a computer screen, because the effect is highly dependent on printer and screen resolution. See Section 1.11 at the end of this chapter, in which you can construct your own aliasing examples in Matlab, as well as audio examples.

1.7 BASIC WAVEFORMS—THE DISCRETE CASE

1.7.1 Discrete Basic Waveforms for Finite Signals

Consider a continuous signal $x(t)$ defined on a time interval $[0, T]$, sampled at the N times $t = nT/N$ for $n = 0, 1, 2, \ldots, N-1$; note we don't sample $x(t)$ at $t = T$. This yields discretized signal $\mathbf{x} = (x_0, x_1, \ldots, x_{N-1})$, where $x_n = nT/N$, a vector in $\mathbb{R}^N$. In all that follows we will index vectors in $\mathbb{R}^N$ from index 0 to index $N-1$, as per Remark 1.4 on page 11.

As we show in a later section, the analog signal $x(t)$ can be decomposed into an infinite linear combination of basic analog waveforms, in this case of the form $e^{2\pi ikt/T}$ for $k \in \mathbb{Z}$. As discussed in the previous section, the appropriate basic waveforms are then the discretized versions of the waveforms $e^{2\pi ikt/T}$, obtained by sampling at times $t = nT/N$. This yields a sequence of basic waveform vectors which we denote by $\mathbf{E}_{N,k}$, indexed by k, of the form

$$\mathbf{E}_{N,k} = \begin{bmatrix} e^{2\pi ik0/N} \\ e^{2\pi ik1/N} \\ \vdots \\ e^{2\pi ik(N-1)/N} \end{bmatrix}, \tag{1.22}$$

a discrete version of $e^{2\pi ikt/T}$. Note though that the waveform vectors don't depend on T. The mth component $\mathbf{E}_{N,k}(m)$ of $\mathbf{E}_{N,k}$ is given by

$$\mathbf{E}_{N,k}(m) = e^{2\pi ikm/N}. \tag{1.23}$$

For any fixed N we can construct the basic waveform vector $\mathbf{E}_{N,k}$ for any $k \in \mathbb{Z}$, but as shown when we discussed aliasing, $\mathbf{E}_{N,k} = \mathbf{E}_{N,k+mN}$ for any integer m. As a consequence we need only consider the $\mathbf{E}_{N,k}$ for a range in k of length N, say of the form $k_0 + 1 \leq k \leq k_0 + N$ for some k_0. A "natural" choice is $k_0 = -N/2$ (if N is even) corresponding to $-N/2 < k \leq N/2$ as in the previous aliasing discussion, but the range $0 \leq k \leq N-1$ is usually more convenient for indexing and matrix algebra. However, no matter which range in k we use to index, we'll always be using the same set of N vectors since $\mathbf{E}_{N,k} = \mathbf{E}_{N,k-N}$.

When there is no potential confusion, we will omit the N index and write simply $\mathbf{E}_k$, rather than $\mathbf{E}_{N,k}$.

■ EXAMPLE 1.11

As an illustration, here are the vectors $\mathbf{E}_{4,k}$ for $k = -2$ to $k = 3$:

$$\mathbf{E}_{4,-2} = \begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \end{bmatrix}, \quad \mathbf{E}_{4,-1} = \begin{bmatrix} 1 \\ -i \\ -1 \\ i \end{bmatrix}, \quad \mathbf{E}_{4,0} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix},$$

$$\mathbf{E}_{4,1} = \begin{bmatrix} 1 \\ i \\ -1 \\ -i \end{bmatrix}, \quad \mathbf{E}_{4,2} = \begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \end{bmatrix}, \quad \mathbf{E}_{4,3} = \begin{bmatrix} 1 \\ -i \\ -1 \\ i \end{bmatrix}.$$

Note the aliasing relations $\mathbf{E}_{4,-2} = \mathbf{E}_{4,2}$ and $\mathbf{E}_{4,-1} = \mathbf{E}_{4,3}$. In particular, the sets $\{\mathbf{E}_{4,-1}, \mathbf{E}_{4,0}, \mathbf{E}_{4,1}, \mathbf{E}_{4,2}\}$ and $\{\mathbf{E}_{4,0}, \mathbf{E}_{4,1}, \mathbf{E}_{4,2}, \mathbf{E}_{4,3}\}$ (corresponding to $\mathbf{E}_{N,k}$ on the range $-N/2 < k \leq N/2$ or $0 \leq k \leq N-1$) are identical.

It's also worth noting the relation

$$\overline{\mathbf{E}_{N,k}} = \mathbf{E}_{N,N-k}, \tag{1.24}$$

where the overline denotes complex conjugation. Equation (1.24) is sometimes called "conjugate aliasing." See Exercise 1.20.

Remark 1.6 There are a lot of periodic functions in the discussion above. For example, the basic waveform $e^{2\pi ikt/T}$ is periodic in t with period T/k. The quantity $\mathbf{E}_{N,k}(m)$ in equation (1.23) is defined for all k and m and periodic in both. As a consequence $\mathbf{E}_{N,k}$ is defined for all k, and periodic with period N. The one entity that is not manifestly periodic is the analog time signal $x(t)$, or its sampled version $\mathbf{x} = (x_0, x_1, \ldots, x_{N-1})$. At times it will be useful, at least conceptually, to extend either periodically. The sampled signal $\mathbf{x} = (x_0, \ldots, x_{N-1})$ can be extended periodically with period N in its index by defining

$$x_m = x_{m \bmod N}$$

for all m outside the range $0 \leq m \leq N-1$. We can also extend the analog signal $x(t)$ periodically to all real t by setting $x(t) = x(t \bmod P)$, where P denotes the period of $x(t)$.

1.7.2 Discrete Basic Waveforms for Images

As with the one-dimensional waveforms, the appropriate discrete waveforms in the two-dimensional case are the sampled basic waveforms. These discrete waveforms are naturally rectangular arrays or matrices.

To be more precise, consider a rectangular domain or image defined by $0 \leq x \leq S$, $0 \leq y \leq R$, but recall Remark 1.1 on page 8; here increasing y is downward. The sampling will take place on an m (in the y direction) by n (x direction) rectangular grid. The basic two-dimensional waveforms were given in equation (1.17). As we will see later, the parameters α and β are most conveniently taken to be of the form $\alpha = 2\pi l/S$ and $\beta = 2\pi k/R$ for integers k and l. Thus the analog basic waveforms to be sampled are the functions $e^{2\pi i(lx/S+ky/R)}$. Each such waveform is sampled at points of the form $x_s = sS/n$, $y_r = rR/m$, with $0 \leq s \leq n-1, 0 \leq r \leq m-1$. The result is an $m \times n$ matrix $\mathcal{E}_{m,n,k,l}$ with row r, column s entry

$$\mathcal{E}_{m,n,k,l}(r, s) = e^{2\pi i(kr/m+ls/n)}. \tag{1.25}$$

Note that $\mathcal{E}_{m,n,k,l}$ does not depend on the image dimensions R or S. The indexes may seem a bit confusing, but recall that m and n are fixed by the discretization size (an m by n pixel image); l and k denote the frequency of the underlying analog waveform in the x and y directions, respectively. The parameters s and r correspond to the x and y coordinates of the sample point. These $m \times n$ matrices $\mathcal{E}_{m,n,k,l}$ constitute the basic waveforms in the discrete setting.

In Exercise 1.17 you are asked to show that $\mathcal{E}_{m,n,k,l}$ can be factored into a product

$$\mathcal{E}_{m,n,k,l} = \mathbf{E}_{m,k}\mathbf{E}_{n,l}^T, \tag{1.26}$$

where the superscript T denotes the matrix transpose operation and where the vectors $\mathbf{E}_{m,k}$ and $\mathbf{E}_{n,l}$ are the discrete basic waveforms in one-dimension, as defined in equation (1.22) (as column vectors). For example,

$$\mathcal{E}_{4,4,1,2} = E_{4,1}E_{4,2}^T = \begin{bmatrix} 1 \\ i \\ -1 \\ -i \end{bmatrix} \begin{bmatrix} 1 & -1 & 1 & -1 \end{bmatrix} = \begin{bmatrix} 1 & -1 & 1 & -1 \\ i & -i & i & -i \\ -1 & 1 & -1 & 1 \\ -i & i & -i & i \end{bmatrix}$$

As in the one-dimensional case we will write $\mathcal{E}_{k,l}$ instead of $\mathcal{E}_{m,n,k,l}$ when there is no possibility for confusion.

A variety of aliasing relations for the waveforms $\mathcal{E}_{m,n,k,l}$ follow from equation (1.26) and those for the $E_{m,k}$; see Exercise 1.21. Also the $\mathcal{E}_{m,n,k,l}$ are periodic in k with period m, and periodic in l with period n. If we confine our attention to the ranges $0 \leq k < m,\ 0 \leq l < n$, there are exactly mn distinct $\mathcal{E}_{m,n,k,l}$ waveforms. For any index pair (l, k) outside this range the corresponding waveform is identical to one of the mn basic waveforms in this range.

The 2D images of the discrete 2D waveforms look pretty much the same as the analog ones do for low values of k and l. For larger values of k and l the effects of aliasing begin to take over and the waveforms are difficult to accurately graph.

Remark 1.7 As in the one-dimensional case it may occasionally be convenient to extend an image matrix with entries $a_{r,s}$, $0 \leq r \leq m-1$ and $0 \leq s \leq n-1$, periodically to the whole plane. We can do this as

$$a_{r,s} = a_{r \bmod m, s \bmod n}.$$

1.8 INNER PRODUCT SPACES AND ORTHOGONALITY

1.8.1 Inner Products and Norms

Vector spaces provide a convenient framework for analyzing signals and images. However, it is helpful to have a bit more mathematical structure to carry out the analysis, specifically some ideas from geometry. Most of the vector spaces we'll be concerned with can be endowed with geometric notions such as "length" and "angle." Of special importance is the idea of "orthogonality." All of these notions can be quantified by adopting an *inner product* on the vector space of interest.

Inner Products The inner product is just a generalization of the familiar dot product from basic multivariable calculus. In the definition below a "function on $V \times V$" means a function whose domain consists of ordered pairs of vectors from V.

Definition 1.8.1 *Let V be a vector space over $\mathbb{C}$ (resp., $\mathbb{R}$). An inner product (or scalar product) on V is a function from $V \times V$ to $\mathbb{C}$ (resp., $\mathbb{R}$). We use $(\mathbf{v}, \mathbf{w})$ to denote the inner product of vectors $\mathbf{v}$ and $\mathbf{w}$ and require that for all vectors $\mathbf{u}, \mathbf{v}, \mathbf{w} \in V$ and scalars a, b in $\mathbb{C}$ (resp., $\mathbb{R}$).*

1. $(\mathbf{v}, \mathbf{w}) = \overline{(\mathbf{w}, \mathbf{v})}$*, (conjugate symmetry)*
2. $(a\mathbf{u} + b\mathbf{v}, \mathbf{w}) = a(\mathbf{u}, \mathbf{w}) + b(\mathbf{v}, \mathbf{w})$*, (linearity in the first argument)*
3. $(\mathbf{v}, \mathbf{v}) \geq 0$*, and* $(\mathbf{v}, \mathbf{v}) = 0$ *if and only if* $\mathbf{v} = \mathbf{0}$

In the case where V is a vector space over $\mathbb{R}$, condition 1 is simply $(\mathbf{v}, \mathbf{w}) = (\mathbf{w}, \mathbf{v})$. If V is over $\mathbb{C}$, then condition 1 also immediately implies that $(\mathbf{v}, \mathbf{v}) = \overline{(\mathbf{v}, \mathbf{v})}$ so that $(\mathbf{v}, \mathbf{v})$ is always real-valued, and hence condition 3 makes sense. The linearity with respect to the first variable in property 2 easily extends to any finite linear combination.

One additional fact worth noting is that the inner product is conjugate-linear in the second argument, that is,

$$\begin{aligned} (\mathbf{w}, a\mathbf{u} + b\mathbf{v}) &= \overline{(a\mathbf{u} + b\mathbf{v}, \mathbf{w})}, && \text{by property 1,} \\ &= \overline{a}\, \overline{(\mathbf{u}, \mathbf{w})} + \overline{b}\, \overline{(\mathbf{v}, \mathbf{w})}, && \text{by property 2,} \\ &= \overline{a}\, (\mathbf{w}, \mathbf{u}) + \overline{b}\, (\mathbf{w}, \mathbf{v}), && \text{by property 1.} \end{aligned} \tag{1.27}$$

A vector space equipped with an inner product is called an *inner product space*.

Norms Another useful geometric notion on a vector space V is that of a *norm*, a way of quantifying the size or length of vectors in V. This also allows us to quantify the distance between elements of V.

Definition 1.8.2 *A norm on a vector space V (over $\mathbb{R}$ or $\mathbb{C}$) is a function $\|\mathbf{v}\|$ from V to $\mathbb{R}$ with the properties that*

1. $\|\mathbf{v}\| \geq 0$*, and* $\|\mathbf{v}\| = 0$ *if and only if* $\mathbf{v} = \mathbf{0}$
2. $\|a\mathbf{v}\| = |a|\|\mathbf{v}\|$
3. $\|\mathbf{v} + \mathbf{w}\| \leq \|\mathbf{v}\| + \|\mathbf{w}\|$ *(the triangle inequality)*

for all vectors $\mathbf{v}, \mathbf{w} \in V$*, and scalars* a.

A vector space equipped with a norm is called (not surprisingly) a *normed vector space*, or sometimes a *normed linear space*.

An inner product $(\mathbf{v}, \mathbf{w})$ on a vector space V always induces a corresponding norm via the relation

$$\|\mathbf{v}\| = \sqrt{(\mathbf{v}, \mathbf{v})}. \tag{1.28}$$

(See Exercise 1.27.) Thus every inner product space is a normed vector space, but the converse is not true; see Example 1.16 and Exercise 1.28. We'll make frequent use of equation (1.28) in the form $\|\mathbf{v}\|^2 = (\mathbf{v}, \mathbf{v})$.

Remark 1.8 It's also useful to define the distance between two vectors $\mathbf{v}$ and $\mathbf{w}$ in a normed vector space as $\|\mathbf{v} - \mathbf{w}\|$. For example, in $\mathbb{R}^n$,

$$\|\mathbf{v} - \mathbf{w}\| = ((v_1 - w_1)^2 + \cdots + (v_n - w_n)^2)^{1/2},$$

which is the usual distance formula.

Generally, the function $\|\mathbf{v} - \mathbf{w}\|$ on $V \times V$ defines a *metric* on V, a way to measure the distance between the elements of the space, and turns V into a *metric space*. The study of metrics and metric spaces is a large area of mathematics, but in this text we won't need this much generality.

Remark 1.9 If the normed vector space V in question has any kind of physical interpretation, then the quantity $\|\mathbf{v}\|^2$ frequently turns out to be proportional to some natural of measure of "energy." The expression for the energy of most physical systems is quadratic in nature with respect to the variables that characterize the state of the system. For example, the kinetic energy of a particle with mass m and speed v is $\frac{1}{2}mv^2$, quadratic in v. The energy dissipated by a resistor is V^2/R, quadratic in V, where R is the resistance and V the potential drop across the resistor. The concept of energy is also important in signal and image analysis, and the quantification of the energy in these settings is quadratic in nature. We'll say more on this later.

1.8.2 Examples

Here are some specific, useful inner product spaces, and the corresponding norms.

■ **EXAMPLE 1.12**

$\mathbb{R}^n$ (real Euclidian space): The most common inner product on $\mathbb{R}^n$ is the dot product, defined by

$$(\mathbf{x}, \mathbf{y}) = x_1 y_1 + x_2 y_2 + \cdots + x_n y_n$$

for vectors $\mathbf{x} = (x_1, \ldots, x_n)$, $\mathbf{y} = (y_1, \ldots, y_n)$ in $\mathbb{R}^n$. Properties 1 to 3 for inner products are easily verified.

The corresponding norm from equation (1.28) is

$$\|\mathbf{x}\| = \left(x_1^2 + x_2^2 + \cdots + x_n^2\right)^{1/2},$$

the usual Euclidean norm.

■ **EXAMPLE 1.13**

$\mathbb{C}^n$ (complex Euclidean space): On $\mathbb{C}^n$ the usual inner product is

$$(\mathbf{x}, \mathbf{y}) = x_1 \overline{y_1} + x_2 \overline{y_2} + \cdots + x_n \overline{y_n}$$

for vectors $\mathbf{x} = (x_1, \ldots, x_n)$, $\mathbf{y} = (y_1, \ldots, y_n)$ in $\mathbb{C}^n$. Conjugation of the second vector's components is important! The corresponding norm from equation (1.28) is

$$\|\mathbf{x}\| = (|x_1|^2 + |x_2|^2 + \cdots + |x_n|^2)^{1/2},$$

where we have made use of the fact that $z\overline{z} = |z|^2$ for any complex number z.

■ **EXAMPLE 1.14**

$M_{m,n}(\mathbb{C})$ ($m \times n$ matrices with complex entries): On $M_{m,n}(\mathbb{C})$ an inner product is given by

$$(\mathbf{A}, \mathbf{B}) = \sum_{j=1}^{m} \sum_{k=1}^{n} a_{j,k} \overline{b_{j,k}}$$

for matrices $\mathbf{A}$ and $\mathbf{B}$ with entries $a_{j,k}$ and $b_{j,k}$, respectively. The corresponding norm from equation (1.28) is

$$\|\mathbf{A}\| = \left(\sum_{j=1}^{m} \sum_{k=1}^{n} |a_{j,k}|^2 \right)^{1/2},$$

called the *Frobenius* norm. As in Example 1.2, as an inner product space $M_{m,n}(\mathbb{C})$ is really "identical" to $\mathbb{C}^{mn}$.

■ EXAMPLE 1.15

$C[a, b]$ (continuous functions on $[a, b]$): Consider the vector space $C[a, b]$, and suppose that the functions can assume complex values. An inner product on this space is given by

$$(f, g) = \int_a^b f(t)\overline{g(t)}\,dt.$$

It's not hard to see that the integral is well-defined, for f and g are continuous functions on a closed interval, hence bounded, so that the product fg is continuous and bounded. The integral therefore converges. Of course, if the functions are real-valued, the conjugation is unnecessary.

Properties 1 and 2 for the inner product follow easily from properties of the Riemann integral. Only property 3 for inner products needs comment. First,

$$(f, f) = \int_a^b |f(t)|^2\,dt \geq 0, \tag{1.29}$$

since the integrand is nonnegative and the integral of a nonnegative function is nonnegative. However, the second assertion in property 3 needs some thought—if the integral in (1.29) actually equals zero, must f be the zero function?

We can prove that this is so by contradiction: suppose that f is not identically zero, say $f(t_0) \neq 0$ for some $t_0 \in [a, b]$. Then $|f(t_0)|^2 > 0$. Moreover, since f is continuous, so is $|f(t)|^2$. We can thus find some small interval $(t_0 - \delta, t_0 + \delta)$ with $\delta > 0$ on which $|f(t)|^2 \geq |f(t_0)|^2/2$. Then

$$(f, f) = \int_a^{t_0-\delta} |f(t)|^2\,dt + \int_{t_0-\delta}^{t_0+\delta} |f(t)|^2\,dt + \int_{t_0+\delta}^{b} |f(t)|^2\,dt.$$

Since $|f(t)|^2 \geq |f(t_0)|^2/2$ for $t \in (t_0 - \delta, t_0 + \delta)$ the middle integral on the right above is positive and greater than $\delta|f(t_0)|^2$ (the area of a 2δ width by $|f(t_0)|^2/2$ tall rectangle under the graph of $|f(t)|^2$). The other two integrals are at least nonnegative. We conclude that if $f \in C[a, b]$ is not the zero function, then $(f, f) > 0$. Equivalently $(f, f) = 0$ only if $f \equiv 0$.

The corresponding norm for this inner product is

$$\|f\| = \left(\int_a^b |f(t)|^2\,dt\right)^{1/2}.$$

In light of the discussion above $\|f\| = 0$ if and only if $f \equiv 0$.

■ EXAMPLE 1.16

Another commonly used norm on the space $C[a, b]$ is the *supremum* norm, defined by

$$\|f\|_\infty = \sup_{x\in[a,b]} |f(x)|.$$

Recall that the supremum of a set $A \subset \mathbb{R}$ is the smallest real number M such $a \leq M$ for every $a \in A$, meaning M is the "least upper bound" for the elements of A. If f is continuous, then we can replace "sup" in the definition of $\|f\|_\infty$ with "max," since a continuous function on a closed bounded interval $[a, b]$ must attain its supremum.

The supremum norm does not come from any inner product in equation (1.28); see Exercise 1.28.

■ EXAMPLE 1.17

$C(\Omega)$ (the set of continuous complex-valued functions on a closed rectangle $\Omega = \{(x, y); a \leq x \leq b, c \leq y \leq d\}$): An inner product on this space is given by

$$(f, g) = \int_a^b\int_c^d f(x, y)\overline{g(x, y)}\,dy\,dx.$$

As in the one-dimensional case, the integral is well-defined since f and g must be bounded; hence the product fg is continuous and bounded. The integral therefore converges. An argument similar to that of the previous example shows that property 3 for inner products holds.

The corresponding norm is

$$\|f\| = \left(\int_a^b \int_c^d |f(x, y)|^2\,dy\,dx\right)^{1/2}.$$

This space can be considerably enlarged, to include many discontinuous functions that satisfy $\|f\| < \infty$.

When we work in a function space like $C[a, b]$ we'll sometimes use the notation $\|f\|_2$ (rather than just $\|f\|$) to indicate the Euclidean norm that stems from the inner product, and so avoid confusion with the supremum norm (or other norms).

1.8.3 Orthogonality

Recall from elementary vector calculus that the dot product $(\mathbf{v}, \mathbf{w})$ of two vectors $\mathbf{v}$ and $\mathbf{w}$ in $\mathbb{R}^2$ or $\mathbb{R}^3$ satisfies the relation

$$(\mathbf{v}, \mathbf{w}) = \|\mathbf{v}\|\|\mathbf{w}\| \cos(\theta), \tag{1.30}$$

where $\|\mathbf{v}\|$ is the length of $\mathbf{v}$, $\|\mathbf{w}\|$ the length of $\mathbf{w}$, and θ is the angle between $\mathbf{v}$ and $\mathbf{w}$. In particular, it's easy to see that $(\mathbf{v}, \mathbf{w}) = 0$ exactly when $\theta = \pi/2$ radians, so $\mathbf{v}$ and $\mathbf{w}$ are orthogonal to each other. The notion of orthogonality can be an incredibly powerful tool. This motivates the following general definition:

Definition 1.8.3 *Two vectors* $\mathbf{v}$ *and* $\mathbf{w}$ *in an inner product space* V *are "orthogonal" if* $(\mathbf{v}, \mathbf{w}) = 0$.

The notion of orthogonality depends on not only the vectors but also the inner product we are using (there may be more than one!) We will say that a subset S (finite or infinite) of vectors in an inner product space V is *pairwise orthogonal*, or more commonly just *orthogonal*, if $(\mathbf{v}, \mathbf{w}) = 0$ for any pair of distinct $(\mathbf{v} \neq \mathbf{w})$ vectors $\mathbf{v}, \mathbf{w} \in S$.

■ EXAMPLE 1.18

Let $S = \{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_n\}$ denote the standard basis vectors in $\mathbb{R}^n$ ($\mathbf{e}_k$ has a "1" in the kth position, zeros elsewhere), with the usual Euclidean inner product (and indexing from 1 to n). The set S is orthogonal since $(\mathbf{e}_j, \mathbf{e}_k) = 0$ when $j \neq k$.

■ EXAMPLE 1.19

Let S denote the set of functions $e^{\pi ikt/T}$, $k \in \mathbb{Z}$, in the vector space $C[-T, T]$, with the inner product as defined in Example 1.15. The set S is orthogonal, for if $k \neq m$, then

$$\begin{aligned}\left(e^{\pi ikt/T}, e^{\pi imt/T}\right) &= \int_{-T}^{T} e^{\pi ikt/T}\overline{e^{\pi imt/T}}\,dt \\ &= \int_{-T}^{T} e^{\pi ikt/T}e^{-\pi imt/T}\,dt \\ &= \int_{-T}^{T} e^{\pi i(k-m)t/T} \\ &= \frac{T(e^{\pi i(k-m)} - e^{-\pi i(k-m)})}{\pi i(k-m)} \\ &= 0,\end{aligned}$$

since $e^{\pi in} - e^{-\pi in} = 0$ for any integer n.

■ EXAMPLE 1.20

Let S denote the set of basic discrete waveforms $\mathbf{E}_{N,k} \in \mathbb{C}^N$, $-N/2 < k \leq N/2$, as defined in equation (1.22). With the inner product on $\mathbb{C}^N$ as defined in

Example 1.13, but on the index range 0 to $N-1$ instead of 1 to N, the set S is orthogonal. The proof is based on the surprisingly versatile algebraic identity

$$1+z+z^2+\cdots+z^{N-1}=\frac{1-z^N}{1-z} \qquad \text{if } z\neq 1. \tag{1.31}$$

If $k\neq l$, then

$$\begin{aligned}(\mathbf{E}_k,\mathbf{E}_l)&=\sum_{r=0}^{N-1}e^{2\pi ikr/N}\overline{e^{2\pi ilr/N}}\\&=\sum_{r=0}^{N-1}e^{2\pi ikr/N}e^{-2\pi ilr/N}\\&=\sum_{r=0}^{N-1}e^{2\pi i(k-l)r/N}\\&=\sum_{r=0}^{N-1}\left(e^{2\pi i(k-l)/N}\right)^r.\end{aligned} \tag{1.32}$$

Let $z=e^{2\pi i(k-l)/N}$ in (1.31) and equation (1.32) becomes

$$\begin{aligned}(\mathbf{E}_k,\mathbf{E}_l)&=\frac{1-\left(e^{2\pi i(k-l)/N}\right)^N}{1-e^{2\pi i(k-l)/N}}\\&=\frac{1-e^{2\pi i(k-l)}}{1-e^{2\pi i(k-l)/N}}\\&=0,\end{aligned}$$

since $e^{2\pi i(k-l)}=1$. Moreover the denominator above cannot equal zero for $-N/2<k,l\leq N/2$ if $k\neq l$. Note the similarity of this computation to the computation in Example 1.19.

Remark 1.10 A very similar computation to that of Example 1.20 shows that the waveforms or matrices $\mathcal{E}_{m,n,k,l}$ in $M_{m,n}(\mathbb{C})$ (with the inner product from Example 1.14 but indexing from 0) are also orthogonal.

1.8.4 The Cauchy–Schwarz Inequality

The following inequality will be extremely useful. It has wide-ranging application and is one of the most famous inequalities in mathematics.

Theorem 1.8.1 (Cauchy–Schwarz) *For any vectors* $\mathbf{v}$ *and* $\mathbf{w}$ *in an inner product space* V *over* $\mathbb{C}$ *or* $\mathbb{R}$*,*

$$|(\mathbf{v}, \mathbf{w})| \leq \|\mathbf{v}\|\|\mathbf{w}\|,$$

where $\|\cdot\|$ *is the norm induced by the inner product via equation (1.28).*

Proof Note that $0 \leq (\mathbf{v} - c\mathbf{w}, \mathbf{v} - c\mathbf{w})$ for any scalar c, from property 3 for inner products. If we expand this out by using the properties of the inner product (including equation (1.27)), we find that

$$\begin{aligned} 0 &\leq (\mathbf{v} - c\mathbf{w}, \mathbf{v} - c\mathbf{w}) \\ &= (\mathbf{v}, \mathbf{v}) - c(\mathbf{w}, \mathbf{v}) - \overline{c}(\mathbf{v}, \mathbf{w}) + (c\mathbf{w}, c\mathbf{w}) \\ &= \|\mathbf{v}\|^2 - c(\mathbf{w}, \mathbf{v}) - \overline{c}(\mathbf{v}, \mathbf{w}) + |c|^2\|\mathbf{w}\|^2. \end{aligned}$$

Let us suppose that $\mathbf{w} \neq \mathbf{0}$, for otherwise, Cauchy–Schwarz is obvious. Choose $c = (\mathbf{v}, \mathbf{w})/(\mathbf{w}, \mathbf{w})$ so that $\overline{c} = (\mathbf{w}, \mathbf{v})/(\mathbf{w}, \mathbf{w})$; note that $(\mathbf{w}, \mathbf{w})$ is real and positive. Then $c(\mathbf{w}, \mathbf{v}) = \overline{c}(\mathbf{v}, \mathbf{w}) = |(\mathbf{v}, \mathbf{w})|^2/\|\mathbf{w}\|^2$ and

$$0 \leq \|\mathbf{v}\|^2 - 2\frac{|(\mathbf{v}, \mathbf{w})|^2}{\|\mathbf{w}\|^2} + \frac{|(\mathbf{v}, \mathbf{w})|^2}{\|\mathbf{w}\|^2} = \|\mathbf{v}\|^2 - \frac{|(\mathbf{v}, \mathbf{w})|^2}{\|\mathbf{w}\|^2}$$

from which the Cauchy-Schwarz inequality follows. ■

1.8.5 Bases and Orthogonal Decomposition

Bases Recall that the set of vectors $S = \{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_n\}$ in $\mathbb{R}^n$ is called the *standard basis*. The reason is that any vector $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ in $\mathbb{R}^n$ can be written as a linear combination

$$\mathbf{x} = x_1\mathbf{e}_1 + x_2\mathbf{e}_2 + \cdots + x_n\mathbf{e}_n$$

of elements from S, in one and only one way. The $\mathbf{e}_k$ thus form a convenient set of building blocks for $\mathbb{R}^n$. The set is "minimal" in the sense that any vector $\mathbf{x}$ can be constructed from the $\mathbf{e}_k$ in only one way.

The following concepts may be familiar from elementary linear algebra in $\mathbb{R}^N$, but they are useful in any vector space.

Definition 1.8.4 *A set* S *(finite or infinite) in a vector space* V *over* $\mathbb{C}$ *(resp.,* $\mathbb{R}$*) is said to "span* V*" if every vector* $\mathbf{v} \in V$ *can be constructed as a finite linear combination of elements of* S*,*

$$\mathbf{v} = \alpha_1\mathbf{v}_1 + \alpha_2\mathbf{v}_2 + \cdots + \alpha_n\mathbf{v}_n,$$

for suitable scalars α_k *in* $\mathbb{C}$ *(resp.,* $\mathbb{R}$*) and vectors* $\mathbf{v}_k \in S$*.*

Definition 1.8.5 *A set S (finite or infinite) in a vector space V over $\mathbb{C}$ (resp., $\mathbb{R}$) is said to be "linearly independent" if, for any finite set of vectors $\mathbf{v}_1, \ldots, \mathbf{v}_n$ in S, the only solution to*

$$\alpha_1 \mathbf{v}_1 + \alpha_2 \mathbf{v}_2 + \cdots + \alpha_n \mathbf{v}_n = \mathbf{0}$$

is $\alpha_k = 0$ for all $1 \leq k \leq n$.

A set S that spans V is thus sufficient to build any vector in V by superposition. Linear independence ensures that no vector can be built in more than one way. A set S that both spans V and is linearly independent is especially useful, for each vector in V can be built from elements of S in a unique way.

Definition 1.8.6 *A linearly independent set S that spans a vector space V is called a basis for V.*

Be careful: a basis S for V may have infinitely many elements, but according to the definition above we must be able to construct any *specific* vector in V using only a *finite* linear combination of vectors in S. The word "basis" has a variety of meanings in mathematics, and the more accurate term for the type of basis defined above, in which only finite combinations are allowed, is a "Hamel basis." If infinite linear combinations of basis vectors are allowed (as in Section 1.10), then issues concerning limits and convergence arise. In either case, however, we'll continue to use the term "basis," and no confusion should arise.

It's worth noting that no linearly independent set and hence no basis can contain the zero vector.

The standard basis in $\mathbb{R}^n$ or $\mathbb{C}^n$, of course, provides an example of a basis. Here are a couple slightly more interesting examples.

■ EXAMPLE 1.21

Consider the space $M_{m,n}(\mathbb{C})$ of $m \times n$ complex matrices, and define mn distinct elements $\mathbf{A}_{p,q} \in M_{m,n}(\mathbb{C})$ as follows: let the row p, column q entry of $\mathbf{A}_{p,q}$ equal 1, and set all other entries of $\mathbf{A}_{p,q}$ equal to zero (quite analogous to the standard basis of $\mathbb{R}^n$ or $\mathbb{C}^n$). The set $S = \{\mathbf{A}_{p,q}; 1 \leq p \leq m, 1 \leq q \leq n\}$ forms a basis for $M_{m,n}(\mathbb{C})$.

■ EXAMPLE 1.22

Let P denote the vector space consisting of all polynomials in the variable x,

$$p(x) = a_0 + a_1 x + \cdots + a_n x^n$$

with real coefficients a_k and no condition on the degree n. You should convince yourself that this is indeed a vector space over $\mathbb{R}$, with the obvious operations. And note that a polynomial has a highest degree term; we're not allowing expressions like $1 + x + x^2 + \cdots$, that is, power series. One basis for P is given by the infinite set

$$S = \{1, x, x^2, x^3, \ldots\}$$

It's not hard to see that any polynomial can be expressed as a finite linear combination of elements of S, and in only one way.

A vector space typically has many different bases. In fact each vector space that will interest us in this text has infinitely many different bases. Which basis we use depends on what we're trying to do.

If a vector space V has a finite basis $S = \{\mathbf{v}_1, \ldots, \mathbf{v}_n\}$, then V is said to be *finite-dimensional*. It is a fact from elementary linear algebra that any other basis for V must also contain exactly n vectors. In this case V is called an *n-dimensional* vector space. In light of the remarks above we can see that $\mathbb{R}^n$ really is an n-dimensional vector space over $\mathbb{R}$ (surprise!), while $\mathbb{C}^n$ is n-dimensional over $\mathbb{C}$. Based on Example 1.21 the spaces $M_{m,n}(\mathbb{R})$ and $M_{m,n}(\mathbb{C})$ are both mn-dimensional vector spaces over $\mathbb{R}$ or $\mathbb{C}$, respectively. The space P in Example 1.22 is infinite-dimensional.

Orthogonal and Orthonormal Bases A lot of vector algebra becomes ridiculously easy when the vectors involved are orthogonal. In particular, finding an orthogonal set of basis vectors for a vector space V can greatly aid analysis and facilitate certain computations.

One very useful observation is the following theorem.

Theorem 1.8.2 *If a set $S \subset V$ of non-zero vectors is orthogonal then S is linearly independent.*

Proof Consider the equation

$$\alpha_1 \mathbf{v}_1 + \alpha_2 \mathbf{v}_2 + \cdots + \alpha_n \mathbf{v}_n = \mathbf{0},$$

where the $\mathbf{v}_k$ are elements of S. Form the inner product of both sides above with any one of the vectors $\mathbf{v}_m$, $1 \leq m \leq n$, and use the linearity of the inner product in the first variable to obtain

$$\sum_{k=1}^{n} \alpha_k (\mathbf{v}_k, \mathbf{v}_m) = (\mathbf{0}, \mathbf{v}_m).$$

The right side above is, of course, 0. Since S is orthogonal, $(\mathbf{v}_k, \mathbf{v}_m) = 0$ unless $k = m$, so the equation above degenerates to $\alpha_m (\mathbf{v}_m, \mathbf{v}_m) = 0$. Because $(\mathbf{v}_m, \mathbf{v}_m) > 0$ (each $\mathbf{v}_m$ is nonzero by hypothesis), we obtain $\alpha_m = 0$ for $1 \leq m \leq n$. ■

In the remainder of this section we assume that V is a finite-dimensional vector space over either $\mathbb{R}$ or $\mathbb{C}$. Of special interest are bases for V that are orthogonal so that $(\mathbf{v}_k, \mathbf{v}_m) = 0$ for any two distinct basis vectors. In this case it's easy to explicitly write any $\mathbf{v} \in V$ as a linear combination of basis vectors.

Theorem 1.8.3 *Let $S = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n\}$ be an orthogonal basis for a vector space V. Then any $\mathbf{v} \in V$ can be expressed as*

$$\mathbf{v} = \sum_{k=1}^{n} \alpha_k \mathbf{v}_k, \tag{1.33}$$

where $\alpha_k = (\mathbf{v}, \mathbf{v}_k)/(\mathbf{v}_k, \mathbf{v}_k)$.

Proof This proof is very similar to that of Theorem 1.8.2. First, since S is a basis, there is some set of α_k that work in equation (1.33). Form the inner product of both sides of equation (1.33) with any $\mathbf{v}_m$, $1 \le m \le n$, and use the linearity of the inner product in the first variable to obtain

$$(\mathbf{v}, \mathbf{v}_m) = \sum_{k=1}^{n} \alpha_k (\mathbf{v}_k, \mathbf{v}_m).$$

Since S is orthogonal $(\mathbf{v}_k, \mathbf{v}_m) = 0$ unless $k = m$, in which case the equation above becomes $(\mathbf{v}, \mathbf{v}_m) = \alpha_m(\mathbf{v}_m, \mathbf{v}_m)$. Thus $\alpha_m = (\mathbf{v}, \mathbf{v}_m)/(\mathbf{v}_m, \mathbf{v}_m)$ is uniquely determined. The denominator $(\mathbf{v}_m, \mathbf{v}_m)$ cannot be zero, since $\mathbf{v}_m$ cannot be the zero vector (because $\mathbf{v}_m$ is part of a linearly independent set). ■

Definition 1.8.7 *An orthogonal set S is orthonormal if $\|\mathbf{v}\| = 1$ for each $\mathbf{v} \in S$.*

In the case where a basis S for V forms an orthonormal set, the expansion in Theorem 1.8.3 becomes a bit simpler since $(\mathbf{v}_k, \mathbf{v}_k) = \|\mathbf{v}_k\|^2 = 1$, so we can take $\alpha_k = (\mathbf{v}, \mathbf{v}_k)$ in equation (1.33).

Remark 1.11 Any orthogonal basis S can be replaced by a "re-scaled" basis that is orthonormal. Specifically, if S is an orthogonal basis for a vector space V, let S' denote the set obtained by replacing each vector $\mathbf{x} \in S$ by the re-scaled vector $\mathbf{x}' = \mathbf{x}/\,\|\mathbf{x}\|$ of length 1. If a vector $\mathbf{v} \in V$ can be expanded according to Theorem 1.8.3, then $\mathbf{v}$ can also be written as a superposition of elements of S', as

$$\mathbf{v} = \sum_{k=1}^{n} (\alpha_k \|\mathbf{v}_k\|)\mathbf{v}'_k,$$

where $\mathbf{v}'{}_k = \mathbf{v}_k/\,\|\mathbf{v}_k\|$ has norm one.

■ EXAMPLE 1.23

The standard basis for $\mathbb{C}^N$ certainly has its place, but for many types of analysis the basic waveforms $\mathbf{E}_{N,k}$ are often more useful. In fact they also form an orthogonal basis for $\mathbb{C}^N$. We've already shown this to be true when in Example 1.20 we showed that the N vectors $\mathbf{E}_{N,k}$ for $0 \leq k \leq N-1$ (or $-N/2 < k \leq N/2$) are mutually orthogonal. By Theorem 1.8.2, the vectors are necessarily linearly independent, and since a set of N linearly independent vectors in $\mathbb{C}^N$ must span $\mathbb{C}^N$, the $\mathbf{E}_{N,k}$ form a basis for $\mathbb{C}^N$.

From equation 1.33 we then obtain a simple decomposition formula

$$\begin{aligned}\mathbf{x} &= \sum_{k=0}^{N-1} \frac{(\mathbf{x}, \mathbf{E}_{N,k})}{(\mathbf{E}_{N,k}, \mathbf{E}_{N,k})}\mathbf{E}_{N,k} \\ &= \frac{1}{N}\sum_{k=0}^{N-1}(\mathbf{x}, \mathbf{E}_{N,k})\mathbf{E}_{N,k} \qquad (1.34)\end{aligned}$$

for any vector $\mathbf{x} \in \mathbb{C}^N$, where we've made use of $(\mathbf{E}_{N,k}, \mathbf{E}_{N,k}) = N$ for each k (see Exercise 1.30). Equation (1.34) will be of paramount importance later; indeed most of Chapter 3 is devoted to the study of equation (1.34)!

■ EXAMPLE 1.24

An entirely analogous argument shows that the matrices $\mathcal{E}_{m,n,k,l}$ form a basis for $M_{m,n}(\mathbb{C})$, and for any matrix $\mathbf{A} \in M_{m,n}(\mathbb{C})$

$$\begin{aligned}\mathbf{A} &= \sum_{k=0}^{m-1}\sum_{l=0}^{n-1} \frac{(\mathbf{A}, \mathcal{E}_{m,n,k,l})}{(\mathcal{E}_{m,n,k,l}, \mathcal{E}_{m,n,k,l})}\mathcal{E}_{m,n,k,l} \\ &= \frac{1}{mn}\sum_{k=0}^{m-1}\sum_{l=0}^{n-1}(\mathbf{A}, \mathcal{E}_{m,n,k,l})\mathcal{E}_{m,n,k,l}, \qquad (1.35)\end{aligned}$$

where we've made use of $(\mathcal{E}_{m,n,k,l}, \mathcal{E}_{m,n,k,l}) = mn$ (see Exercise 1.31).

Parseval's Identity Suppose that S is an orthonormal basis for an n-dimensional vector space V. Let $\mathbf{v} \in V$ be expanded according to equation (1.33). Then

$$\begin{aligned}\|\mathbf{v}\|^2 &= (\mathbf{v}, \mathbf{v}) \\ &= \left(\sum_j \alpha_j \mathbf{v}_j, \sum_k \alpha_k \mathbf{v}_k\right)\end{aligned}$$

$$= \sum_{j,k=1}^{n} \alpha_j \overline{\alpha_k}(\mathbf{v}_j, \mathbf{v}_k)$$

$$= \sum_{k=1}^{n} |\alpha_k|^2, \qquad (1.36)$$

where we have used the properties of the inner product (including equation (1.27)), $\alpha_k\overline{\alpha_k} = |\alpha_k|^2$, and the fact that S is orthonormal so $(\mathbf{v}_k, \mathbf{v}_k) = 1$. Equation (1.36) is called *Parseval's identity*.

As noted in Remark 1.9 on page 30, $\|\mathbf{v}\|^2$ is often interpreted as the energy of the discretized signal. If the basis set S is orthonormal, then each vector $\mathbf{v}_k$ represents a basis signal that is scaled to have energy equal to 1, and it's easy to see that the signal $\alpha_k\mathbf{v}_k$ has energy $|\alpha_k|^2$. In this case Parseval's identity asserts that the "energy of the sum is the sum of the energies."

1.9 SIGNAL AND IMAGE DIGITIZATION

As discussed earlier in the chapter, general analog signals and images cannot be meaningfully stored in a computer but must be converted to digital form. As noted in Section 1.3.2, this introduces a quantization error. It's tempting to minimize this error by storing the underlying real numbers as high-precision floating point values, but this would be expensive in terms of storage. In Matlab a grayscale image stored as double precision floating point numbers requires eight bytes per pixel, compared to one byte per pixel for the eight-bit quantization scheme discussed Section 1.3.6. Furthermore, if very fast processing is required, it's usually better to use integer arithmetic chips than floating point hardware. Thus we must balance quantization error with the storage and computational costs associated with more accurate digitization. In order to better understand this issue, we now take a closer look at quantization and a more general scheme than that presented in Section 1.3.2.

1.9.1 Quantization and Dequantization

Let's start with a simple but representative example.

■ **EXAMPLE 1.25**

Consider an analog signal $x(t)$ that can assume "any" real value at any particular time t. The signal would, of course, be sampled to produce a string of real numbers x_k for k in some range, say $0 \leq k \leq n$. This still doesn't suffice for computer storage though, since we can't store even a single real number x_k to infinite precision. What we'll do is this: divide the real line into "bins," say the disjoint intervals $(-\infty, -5]$, $(-5, 3]$, $(3, 7]$, and $(7, \infty)$. Note that these are chosen arbitrarily here, solely for the sake of example. We thus have four *quantization* *intervals*. Every real number falls into exactly one of these intervals. We'll refer to the interval $(-\infty, -5]$ as "interval 0," $(-5, 3]$ as "interval 1," $(3, 7]$ as "interval 2," and $(7, \infty)$ as "interval 3." In this manner any real number z can be associated with an integer in the range 0 to 3, according to the quantization interval in which z lies. This defines a *quantization map* q from $\mathbb{R}$ to the set $\{0, 1, 2, 3\}$. Rather than storing z we store (with some obvious loss of information) $q(z)$. Indeed, since $q(z)$ can assume only four distinct values, it can be stored with just two bits. We can store the entire discretized signal $x(t)$ with just $2(n + 1)$ bits, for example, as "00" for a sample x_k in interval 0, "01" for interval 1, "10" for interval 2, "11" for interval 3.

To reconstruct an approximation to any given sample x_k, we proceed as follows: for each quantization interval we choose a representative value z_k for quantization interval k. For example, we can take $z_0 = -10$, $z_1 = -1$, $z_2 = 5$, and $z_3 = 10$. Here z_1 and z_2 are chosen as the midpoints of the corresponding interval, z_0 and z_3 as "representative" of their intervals. If a sample x_k falls in quantization interval 0 (i.e., was stored as the bit sequence "00"), we reconstruct approximately as $\tilde{x}_k = z_0$. A similar computation is performed for the other intervals. This yields a *dequantization map* $\tilde{q}$ from the set $\{0, 1, 2, 3\}$ back to $\mathbb{R}$.

As a specific example, consider the sampled signal $\mathbf{x} \in \mathbb{R}^5$ with components $x_0 = -1.2$, $x_1 = 2.3$, $x_2 = 4.4$, $x_3 = 8.8$, and $x_4 = -2.8$. The quantization map yields $q(\mathbf{x}) = (1, 1, 2, 3, 1)$ when q is applied component-by-component to $\mathbf{x}$. The reconstructed version of $\mathbf{x}$ is $\tilde{q}(q(\mathbf{x})) = (-1, -1, 5, 10, -1)$.

The General Quantization Scheme The quantization scheme above generalizes as follows. Let r be the number of quantization levels ($r = 4$ in the previous example). Choose $r - 1$ distinct quantization "jump points" $\{y_1, \dots, y_{r-1}\}$, real numbers that satisfy $-\infty < y_1 < y_2 < \cdots < y_{r-1} < \infty$ (we had $y_1 = -5$, $y_2 = 3$, $y_3 = 7$ above). Let $y_0 = -\infty$ and $y_r = \infty$. We call the interval $(y_k, y_{k+1}]$ the *kth quantization interval*, where $0 \leq k \leq r - 1$ (we should really use an open interval (y_{r-1}, ∞) for the last interval). The leftmost and rightmost intervals are unbounded. Each real number belongs to exactly one of the r quantization intervals.

The *quantization map* $q : \mathbb{R} \to \{0, 1, \dots, r - 1\}$ assigns an integer quantization level $q(x)$ to each $x \in \mathbb{R}$ as follows: $q(x)$ is the index k such that $y_k < x \leq y_{k+1}$, meaning, x belongs to the kth quantization interval. The function q can be written more explicitly if we define the Heaviside function

$$H(x) = \begin{cases} 0, & x \leq 0, \\ 1, & x > 0, \end{cases}$$

in which case

$$q(x) = \sum_{k=1}^{r-1} H(x - y_k).$$

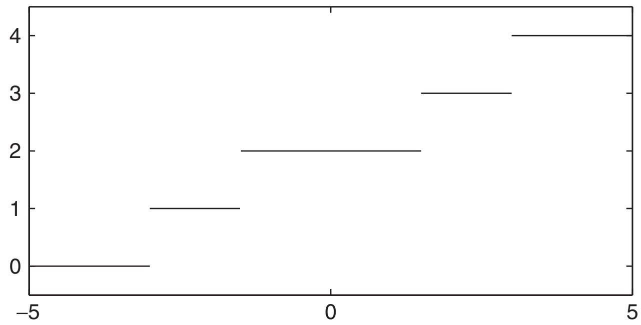

FIGURE 1.11 Quantization function $H(x + 3) + H(x + 1.5) + H(x - 1.5) + H(x - 3)$.

A sample graph of a quantization function is given in Figure 1.11, in which we have chosen $y_1 = -3$, $y_2 = -1.5$, $y_3 = 1.5$, $y_4 = 3$, and $y_0 = -\infty$, $y_5 = \infty$. Notice that the quantization interval around zero is bigger than the rest. This is not uncommon in practice.

The quantization map is used as follows: Let $\mathbf{x}$ denote a sampled signal, so each component x_j of $\mathbf{x}$ is a real number and has not yet been quantized. The quantized version of $\mathbf{x}$ is just $q(\mathbf{x})$, in which q is applied component-by-component to $\mathbf{x}$. Each x_j is thus assigned to one of the r quantization intervals. If $r = 2^b$ for some $b > 0$, then each quantized x_j can be stored using b bits, and we refer to this as "b-bit quantization." A similar procedure would be applied to images.

Dequantization Once quantized, the vector $\mathbf{x}$ cannot be exactly recovered because q is not invertible. If we need to approximately reconstruct $\mathbf{x}$ after quantization we do the following: pick real numbers $z_0, z_1, \ldots, z_{r-1}$ such that z_k lies in the kth quantization interval, that is $y_{k-1} < z_{k-1} \leq y_k$. Ideally the value of z_k should be a good approximation to the average value of the entries of $\mathbf{x}$ that fall in the kth quantization interval. A simple choice is to take z_k as the midpoint of the kth quantization interval, as in the example above. The set $\{z_k : 0 \leq k \leq r - 1\}$ is called the *codebook* and the z_k are called the *codewords*. Define the *dequantization map* $\widetilde{q} : \{0, \ldots, r - 1\} \to \mathbb{R}$ that takes k to z_k. Then define the approximate reconstruction of $\mathbf{x}$ as the vector $\widetilde{\mathbf{x}}$ with components $\widetilde{x}_j$ where

$$\widetilde{x}_j = \widetilde{q}(q(x_j)) = z_{q(x_j)}.$$

If $y_{k-1} < x_j \leq y_k$, then x_j is mapped to the kth quantization interval (i.e., $q(x_j) = k$) and $\widetilde{x}_j = \widetilde{q}(q(x_j)) = z_k$ where $y_{k-1} < z_j \leq y_k$. In other words, x_j and the quantized/dequantized quantity $\widetilde{x}_j$ both lie in the interval $y_{k-1} < x_j \leq y_k$, so at worst the discrepancy is

$$|x_j - \widetilde{x}_j| \leq y_k - y_{k-1}. \tag{1.37}$$

If the y_k are finely spaced, the error won't be too large. But, of course, more y_k means more bits are needed for storage. The same procedure can be applied to images/matrices.

Measuring Error Ideally we'd like to choose our quantization/dequantization functions to minimize the distortion or error introduced by this process. This requires a way to quantify the distortion. One simple measure of the distortion is $\|\mathbf{x} - \widetilde{\mathbf{x}}\|^2$, the squared distance between $\mathbf{x}$ and $\widetilde{\mathbf{x}}$ as vectors in $\mathbb{R}^n$ (or $\mathbb{C}^n$). Actually it is slightly more useful to quantify distortion in relative terms, as a fraction of the original signal energy, so we use $(\|\mathbf{x} - \widetilde{\mathbf{x}}\|^2)/\|\mathbf{x}\|^2$. By adjusting the values of the y_k's and the z_k's, we can, in principle, minimize distortion for any specific signal $\mathbf{x}$ or image. In the case of image compression, the quantization levels $\{z_k\}$ can be stored with the compressed file. For other applications, we may want the quantization levels to be fixed ahead of time, say in an audio application. In this case, it's sensible to minimize the distortion over an entire class of signals or images. We would also like the quantization and dequantization computations to be simple. The following example gives a scheme that works fairly well.

■ EXAMPLE 1.26

In this example we'll quantize an image, but the same principles apply to a one-dimensional signal. Let us assume that the intensity values of a class of grayscale images of interest satisfy $m \le a(x, y) \le M$ on some rectangle Ω, where $a(x, y)$ is the analog image intensity. Let $\mathbf{A}$ denote the matrix with components a_{jk} obtained by sampling (but not quantizing) the analog image $a(x, y)$. Select the y_k so that they split up the interval $[m, M]$ into r subintervals of equal length, and let z_k be the mid-point of each interval. If we define $h = (M - m)/r$, then we obtain the following formulas for the y_k's, the z_k's, q, and $\widetilde{q}$:

$$\begin{aligned} y_k &= m + kh, \qquad k = 1, \ldots, r-1, \;\; y_0 = -\infty, \;\; y_r = \infty, \\ z_k &= m + \left(k + \frac{1}{2}\right) h, \qquad k = 0, \ldots, r-1, \\ q(x) &= \operatorname{ceil}\left(r \frac{x - m}{M - m}\right) - 1 \qquad \text{for } x > m, \qquad q(m) = 0, \\ \widetilde{q}(k) &= m + \left(\frac{k+1}{2}\right) h. \end{aligned}$$

The ceiling function "ceil" from $\mathbb{R}$ to $\mathbb{Z}$ is defined by taking $\operatorname{ceil}(x)$ as the smallest integer greater than or equal to x.

To illustrate, the image at the top left in Figure 1.12 has a sample matrix $\mathbf{A}$ (stored as double precision floating point) with limits $0 \le a_{ij} \le 255$. The quantization method above at 5 bits per pixel (32 quantization intervals) or greater gives no measurable distortion. In Figure 1.12 we illustrate quantization at each of $b = 4, 2, 1$ bits per pixel (bpp) in order to see the distortion. The measure of

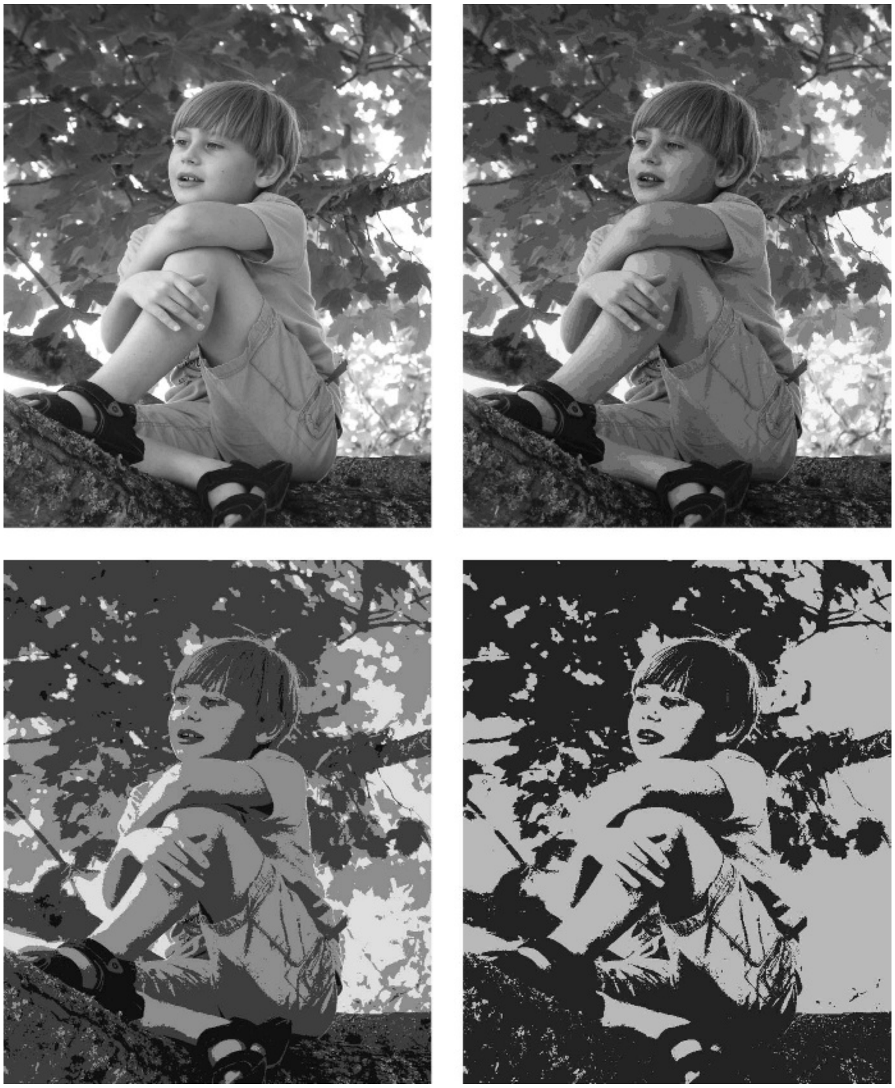

FIGURE 1.12 Original image (*top left*) and quantization at 4 bits (*top right*), 2 bits (*bottom left*) and 1 bit (*bottom right*).

the distortion *mD* is reported as a percentage of the total image energy,

$$mD = 100\frac{\|\mathbf{A} - \widetilde{\mathbf{A}}\|^2}{\|\mathbf{A}\|^2},$$

where $\|\cdot\|$ denotes the Frobenius norm of Example 1.14. The resulting errors are 0.2, 3.6, and 15.2 percent for the 4, 2, and 1 bit quantizations, respectively.

Example 1.26 illustrates uniform quantization with midpoint codewords. It also yields an improvement over the error estimate 1.37, namely

$$\left|a_{ij} - \widetilde{a}_{ij}\right| \leq \frac{h}{2}. \tag{1.38}$$

1.9.2 Quantifying Signal and Image Distortion More Generally

Suppose that we have signal that we want to compress or denoise to produce a processed approximation. The quantization discussion above provides a concrete example, but other similar situations will arise later. In general, what is a reasonable way to quantify the accuracy of the approximation?

Many approaches are possible, but we'll do essentially as we did for the image in Example 1.26. For a discretized image (or signal) $\mathbf{A}$ approximated by $\widetilde{\mathbf{A}}$, write

$$\widetilde{\mathbf{A}} = \mathbf{A} + \mathbf{E},$$

where $\mathbf{E} = \widetilde{\mathbf{A}} - \mathbf{A}$ is the error introduced by using the approximation $\widetilde{\mathbf{A}}$ for $\mathbf{A}$. We could also consider $\mathbf{E}$ as some kind of random noise. Our measure of distortion (or noise level, if appropriate) is

$$mD = \frac{\|\widetilde{\mathbf{A}} - \mathbf{A}\|^2}{\|\mathbf{A}\|^2} = \frac{\|\mathbf{E}\|^2}{\|\mathbf{A}\|^2}, \tag{1.39}$$

the relative error as is typically done in any physical application. We will often report this error as a percentage as we did with the images above, by multiplying by 100.

1.10 INFINITE-DIMENSIONAL INNER PRODUCT SPACES

Analog signals and images are naturally modeled by functions of one or more real variables, and as such the proper setting for the analysis of these objects is a function space such as $C[a, b]$. However, vector spaces of functions are infinite-dimensional, and some of the techniques developed for finite-dimensional vector spaces need a bit of adjustment. In this section we give an outline of the mathematics necessary to carry out orthogonal expansions in these function spaces, especially orthogonal expansions with regard to complex exponentials. The ideas in this section play a huge role in applied mathematics. They also provide a nice parallel to the discrete ideas.

Example: An Infinite-dimensional Space Let's focus on the vector space $C[a, b]$ for the moment. This space is not finite-dimensional. This can be shown by demonstrating the existence of m linearly independent functions in $C[a, b]$ for any integer $m > 0$. To do this, let $h = (b - a)/m$ and set $x_k = a + (k - 1)h$ for $0 \leq k \leq m$; the points x_k partition $[a, b]$ into m equal subintervals, each of length h

(with $x_0 = a, x_m = b$). Let $I_k = [x_{k-1}, x_k]$ for $1 \leq k \leq m$, and define m functions

$$\phi_k(x) = \begin{cases} 0, & \text{if } x \text{ is not in } I_k, \\ \dfrac{2}{h}(x - x_{k-1}), & x_{k-1} \leq x \leq (x_{k-1} + x_k)/2, \\ \dfrac{2}{h}(x_k - x), & (x_{k-1} + x_k)/2 < x \leq x_k, \end{cases}$$

for $1 \leq k \leq m$. Each function ϕ_k is continuous and piecewise linear, identically zero outside I_k, with $\phi_k = 1$ at the midpoint of I_k (such a function is sometimes called a "tent" function—draw a picture). It's easy to show that the functions ϕ_k are linearly independent, for if

$$\sum_{k=1}^{m} c_k \phi_k(x) = 0$$

for $a \leq x \leq b$, then evaluating the left side above at the midpoint of I_j immediately yields $c_j = 0$. The ϕ_k thus form a linearly independent set. If $C[a, b]$ were finite-dimensional, say of dimension n, then we would not be able to find a set of $m > n$ linearly independent functions. Thus $C[a, b]$ is not finite-dimensional.

A similar argument can be used to show that any of the vector spaces of functions from Section 1.4.2 are infinite-dimensional.

1.10.1 Orthogonal Bases in Inner Product Spaces

It can be shown that any vector space, even an infinite-dimensional space, has a basis in the sense of Definition 1.8.6 (a Hamel basis), in which only finite combinations of the basis vectors are allowed. However, such bases are usually difficult to exhibit explicitly and of little use for computation. As such, we're going to expand our notion of basis to allow infinite combinations of the basis vectors. Of course, the word "infinite" always means limits are involved, and if so, we need some measure of distance or length, since limits involve some quantity "getting close" to another.

In light of this criterion it's helpful to restrict our attention to vector spaces in which we have some notion of distance, such as a normed vector space. But orthogonality, especially in the infinite-dimensional case, is such a valuable asset that we're going to restrict our attention to inner product spaces. The norm will be that associated with the inner product via equation (1.28).

Let V be an inner product space, over either $\mathbb{R}$ or $\mathbb{C}$. We seek a set S of vectors in V,

$$S = \{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \ldots\}, \tag{1.40}$$

that can act as a basis for V in some reasonable sense. Note that we are assuming that the elements of S can be listed, meaning put into a one-to-one correspondence with the positive integers. Such a set is said to be *countable* (many infinite sets are not!)

Remark 1.12 In what follows it's not important that the elements of the set S be indexed as $1, 2, 3, \ldots$. Indeed the elements can be indexed from any subset of $\mathbb{Z}$, $\mathbb{Z}$ itself, or even $\mathbb{Z} \times \mathbb{Z}$. The main point is that we must be able to sum over the elements of S using traditional summation notation, $\sum$. We index the elements starting from 1 in the discussion that follows solely to fix notation.

In the finite-dimensional case the basis vectors must be linearly independent, and this was an automatic consequence of orthogonality. In the infinite-dimensional case we'll cut straight to the chase: in this text we will only consider as prospective bases those sets that are orthogonal (though nonorthogonal bases can be constructed).

In the finite-dimensional case, a basis must also span the vector space. In the present case, we are allowing infinite linear combinations of basis vectors, and want to be able to write

$$\mathbf{v} = \sum_{k=1}^{\infty} \alpha_k \mathbf{v}_k \tag{1.41}$$

for any $\mathbf{v} \in V$ by choosing the coefficients α_k suitably. But infinite linear combinations have no meaning in a general vector space. How should equation (1.41) be interpreted?

Recall that in elementary calculus the precise definition of an infinite sum

$$\sum_{k=1}^{\infty} a_k = A$$

is that $\lim_{n\to\infty}(\sum_{k=1}^{n} a_k) = A$ (the sequence of partial sums converges to A). This is equivalent to

$$\lim_{n\to\infty} \left| \sum_{k=1}^{n} a_k - A \right| = 0.$$

This motivates our interpretation of equation (1.41) and the definition of what it means for the set S in (1.40) to span V. However, in an infinite-dimensional inner product space the term "span" is replaced by "complete."

Definition 1.10.1 *An (orthogonal) set S as in (1.40) is "complete" if for each $\mathbf{v} \in V$ there are scalars α_k, $k \geq 1$, such that*

$$\lim_{n\to\infty} \left\| \sum_{k=1}^{n} \alpha_k \mathbf{v}_k - \mathbf{v} \right\| = 0. \tag{1.42}$$

The limit in (1.42) is just an ordinary limit for a sequence of real numbers. The norm on the inner product space takes the place of absolute value in $\mathbb{R}$.

We can now define

Definition 1.10.2 *A set S as in (1.40) is called an "orthogonal basis" for V if S is complete and orthogonal. If S is orthonormal, then S is called an "orthonormal basis."*

The existence of an orthogonal basis as in Definition 1.10.2 is not assured but depends on the particular inner product space. However, all of the function spaces of interest from Section 1.8.2 have such bases. In the case of spaces consisting of functions of a single real variable (e.g., $C[a, b]$), we can write out a basis explicitly, and also for function spaces defined on a rectangle in the plane. We'll do this shortly.

1.10.2 The Cauchy–Schwarz Inequality and Orthogonal Expansions

For now let's assume that an orthogonal basis $S = \{\mathbf{v}_1, \mathbf{v}_2, \ldots\}$ exists. How can we compute the α_k in the expansion (1.41)? Are they uniquely determined? It's tempting to mimic the procedure used in the finite-dimensional case: take the inner product of both sides of (1.41) with a specific basis vector $\mathbf{v}_m$ to obtain $(\mathbf{v}, \mathbf{v}_m) = (\sum_k \alpha_k \mathbf{v}_k, \mathbf{v}_m)$ then use linearity of the inner product in the first argument to obtain $(\mathbf{v}, \mathbf{v}_m) = \sum_k \alpha_k(\mathbf{v}_k, \mathbf{v}_m) = \alpha_m(\mathbf{v}_m, \mathbf{v}_m)$. This immediately yields $\alpha_m = (\mathbf{v}, \mathbf{v}_m)/(\mathbf{v}_m, \mathbf{v}_m)$, just as in the finite-dimensional case. This reasoning is a bit suspect, though, because it requires us to invoke linearity for the inner product with respect to an infinite sum. Unfortunately, the definition of the inner product makes no statements concerning infinite sums. We need to be a bit more careful (though the answer for α_m is correct!)

To demonstrate the validity of the conclusion above more carefully we'll use the Cauchy–Schwarz inequality in Theorem 1.8.1. Specifically, suppose that $S = \{\mathbf{v}_1, \mathbf{v}_2, \ldots\}$ is an orthogonal basis so that for any $\mathbf{v} \in V$ there is some choice of scalars α_k for which equation (1.42) holds. For some fixed m consider the inner product $(\mathbf{v} - \sum_{k=1}^{n} \alpha_k \mathbf{v}_k, \mathbf{v}_m)$. If we expand this inner product and suppose $n \geq m$ while taking absolute values throughout, we find that

$$\begin{aligned} \left| \left(\mathbf{v} - \sum_{k=1}^{n} \alpha_k \mathbf{v}_k, \mathbf{v}_m \right) \right| &= \left| (\mathbf{v}, \mathbf{v}_m) - \sum_{k=1}^{n} \alpha_k (\mathbf{v}_k, \mathbf{v}_m) \right| \\ &= \left| (\mathbf{v}, \mathbf{v}_m) - \alpha_m(\mathbf{v}_m, \mathbf{v}_m) \right|. \end{aligned} \tag{1.43}$$

Note that all sums above are finite. On the other hand, the Cauchy–Schwarz inequality yields

$$\left| \left(\mathbf{v} - \sum_{k=1}^{n} \alpha_k \mathbf{v}_k, \mathbf{v}_m \right) \right| \leq \|\mathbf{v} - \sum_{k=1}^{n} \alpha_k \mathbf{v}_k\| \|\mathbf{v}_m\|. \tag{1.44}$$

Combine (1.43) and (1.44) to find that for $n \geq m$

$$|(\mathbf{v}, \mathbf{v}_m) - \alpha_m(\mathbf{v}_m, \mathbf{v}_m)| \leq \left\| \mathbf{v} - \sum_{k=1}^{n} \alpha_k \mathbf{v}_k \right\| \|\mathbf{v}_m\|. \tag{1.45}$$

The completeness assumption forces $\|\mathbf{v} - \sum_{k=1}^{n} \alpha_k \mathbf{v}_k\| \to 0$ as $n \to \infty$ on the right side of (1.45) (and $\|\mathbf{v}_m\|$ remains fixed, since m is fixed). The left side of (1.45) must also approach zero, but the left side doesn't depend on n. We must conclude that $(\mathbf{v}, \mathbf{v}_m) - \alpha_m(\mathbf{v}_m, \mathbf{v}_m) = 0$, so $\alpha_m = (\mathbf{v}, \mathbf{v}_m)/(\mathbf{v}_m, \mathbf{v}_m)$. Of course, if S is orthonormal, this becomes just $\alpha_m = (\mathbf{v}, \mathbf{v}_m)$.

Note that we assumed an expansion as in (1.42) exists. What we've shown above is that IF such an expansion exists (i.e. , if S is complete), THEN the α_k are uniquely determined and given by the formula derived. Let's state this as a theorem.

Theorem 1.10.1 *If S is an orthogonal basis for an inner product space V, then for any $\mathbf{v} \in V$, equation (1.42) holds where $\alpha_k = (\mathbf{v}, \mathbf{v}_k)/(\mathbf{v}_k, \mathbf{v}_k)$.*

It's conventional to write the expansion of $\mathbf{v}$ in the shorthand form of (1.41), with the understanding that the precise meaning is the limit in (1.42).

Interestingly, Parseval's identity still holds. Note that if S is an orthonormal basis and $\alpha_k = (\mathbf{v}, \mathbf{v}_k)$, then

$$\begin{aligned}\left\|\mathbf{v} - \sum_{k=1}^{n} \alpha_k \mathbf{v}_k\right\|^2 &= \left(\mathbf{v} - \sum_{k=1}^{n} \alpha_k \mathbf{v}_k, \mathbf{v} - \sum_{k=1}^{n} \alpha_k \mathbf{v}_k\right)\\ &= (\mathbf{v}, \mathbf{v}) - \sum_{k=1}^{n} \alpha_k(\mathbf{v}_k, \mathbf{v}) - \sum_{k=1}^{n} \overline{\alpha_k}(\mathbf{v}, \mathbf{v}_k) + \sum_{k=1}^{n} |\alpha_k|^2\\ &= \|\mathbf{v}\|^2 - \sum_{k=1}^{n} |\alpha_k|^2.\end{aligned}$$

As $n \to \infty$ the left side approaches zero; hence so does the right side, and we obtain

$$\sum_{k=1}^{\infty} |\alpha_k|^2 = \|\mathbf{v}\|^2, \tag{1.46}$$

which is Parseval's identity in the infinite-dimensional case.

There are a variety of equivalent characterizations or definitions of what it means for a set S to be complete (e.g., S is complete if Parseval's identity holds for all $\mathbf{v}$), and some may make it easier or harder to verify that any given set S is complete. Proving that a given set of vectors in a specific inner product space is complete always involves some analysis ("analysis" in the sense of limits, inequalities, and estimates). We won't go into these issues in much detail in this text. We will simply exhibit, without proof, some orthogonal bases for common spaces of interest.

1.10.3 The Basic Waveforms and Fourier Series

For the moment let's focus on the space $C[-T, T]$ of continuous complex-valued functions on the closed interval $[-T, T]$, where $T > 0$. This, of course, includes the real-valued functions on the interval. The vector space $C[-T, T]$ becomes an inner product space if we define the inner product as in Example 1.15.

Complex Exponential Fourier Series Let S denote the set of basic analog waveforms $\phi_k(t) = e^{ik\pi t/T}$ for $k \in \mathbb{Z}$ introduced in Example 1.19. In that example it was shown that this set is orthogonal. It can be shown with a bit of nontrivial analysis that this set is complete in $C[-T, T]$ or $L^2(-T, T)$ (see, e.g., [21] or [10], sec. II.4.4). As a consequence S is a basis.

Note that the basis ϕ_k here is indexed in k, which ranges over all of $\mathbb{Z}$, but according to Remark 1.12 on page 48 this makes no essential difference.

Consider a typical function $f(t)$ in $C[-T, T]$. From Theorem 1.10.1 we have the expansion

$$f(t) = \sum_{k=-\infty}^{\infty} \alpha_k e^{ik\pi t/T}, \tag{1.47}$$

where

$$\alpha_k = \frac{(f, e^{ik\pi t/T})}{(e^{ik\pi t/T}, e^{\pi ikt/T})} = \frac{1}{2T}\int_{-T}^{T} f(t)e^{-ik\pi t/T}\,dt, \tag{1.48}$$

since $\int_{-T}^{T} e^{ik\pi t/T}e^{-ik\pi t/T}\,dt = 2T$. The right side of equation (1.47) is called the *Fourier series* for f, and the α_k of equation (1.48) are called the *Fourier coefficients*.

■ **EXAMPLE 1.27**

Let $T = 1$, and consider the function $f(t) = t^2$ in $C[-1, 1]$. The Fourier coefficients from equation (1.48) are given by

$$\begin{aligned}\alpha_k &= \frac{1}{2}\int_{-1}^{1} t^2 e^{-ik\pi t}\,dt\\ &= \frac{2(-1)^k}{\pi^2 k^2}\end{aligned}$$

for $k \neq 0$ (integrate by parts twice), while $\alpha_0 = \frac{1}{3}$. We can write out the Fourier series for this function as

$$f(t) = \frac{1}{3} + \frac{2}{\pi^2}\sum_{k=-\infty, k\neq 0}^{\infty} \frac{(-1)^k e^{ik\pi t}}{k^2}.$$

It should be emphasized that the series on the right "equals" $f(t)$ only in the sense defined by (1.42). As such, it is instructive to plot $f(t)$ and the right side above summed from $k = -n$ to n for a few values of n, as in Figure 1.13. In the case where $n = 0$ the approximation is just $f(t) \approx a_0 = (\int_{-T}^{T} f(t)\,dt)/2T$, the average value of f over the interval. In general, there is no guarantee that the Fourier series for any specific value $t = t_0$ converges to the value $f(t_0)$, unless one knows something more about f. However, the Fourier series will converge in the sense of equation (1.42) for any function $f \in L^2(-T, T)$.

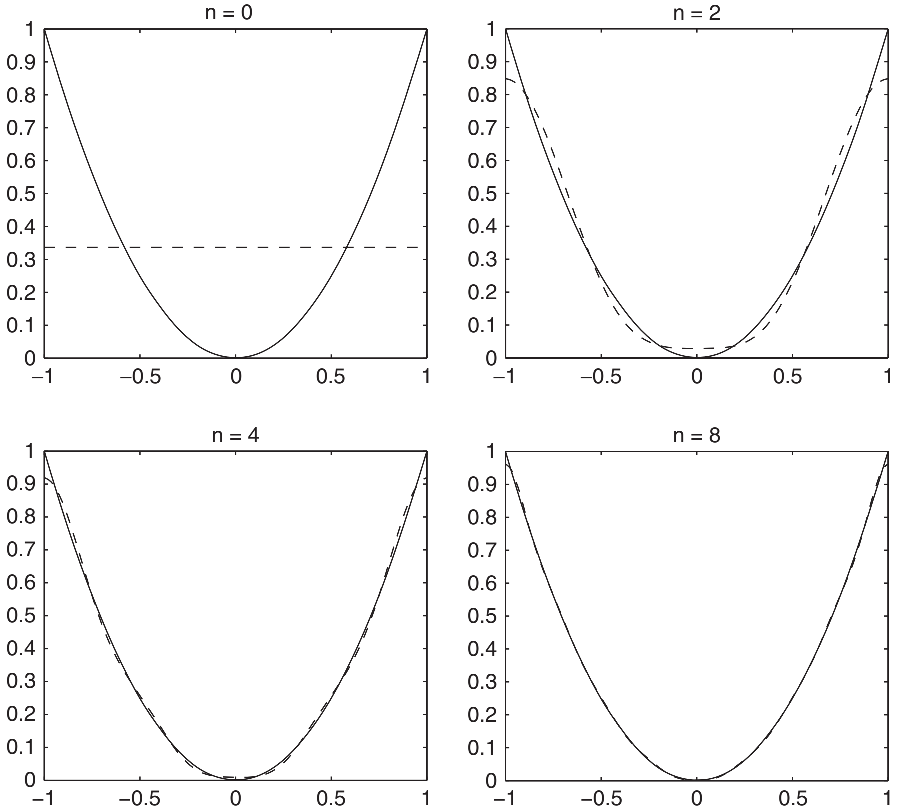

FIGURE 1.13 Function $f(t) = t^2$ (*solid*) and Fourier series approximations (*dashed*), $n = 0, 2, 4, 8$.

It's also worth noting that the Fourier series itself continues periodically outside the interval $[-T, T]$ with period $2T$, since each of the basic waveforms are periodic with period $2T$. This occurs even though f does not need to be defined outside this interval. This case is shown in Figure 1.14.

Sines and Cosines Fourier expansions can be also written using sines and cosines. Specifically, we can write the Fourier coefficients of a function $f(t)$ on $[-T, T]$ as

$$\begin{aligned}\alpha_k &= \frac{1}{2T}\int_{-T}^{T} f(t)e^{-ik\pi t/T}\,dt \\ &= \frac{1}{2T}\int_{-T}^{T} f(t)\left(\cos\left(\frac{k\pi t}{T}\right) - i\sin\left(\frac{k\pi t}{T}\right)\right)dt \\ &= a_k - ib_k \qquad (1.49)\end{aligned}$$

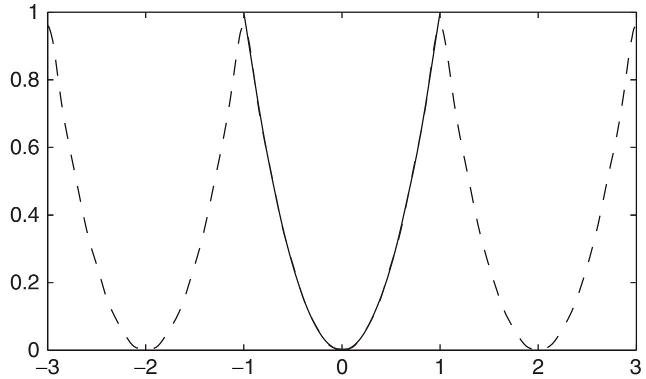

FIGURE 1.14 Function $f(t) = t^2$, $-1 \le t \le 1$, and eight-term Fourier series approximation extended to $[-3, 3]$.

where

$$a_k = \frac{1}{2T}\int_{-T}^{T} f(t)\cos\left(\frac{k\pi t}{T}\right)\,dt, \quad b_k = \frac{1}{2T}\int_{-T}^{T} f(t)\sin\left(\frac{k\pi t}{T}\right)\,dt. \tag{1.50}$$

Observe that $b_0 = 0$, while $a_0 = \alpha_0$ and $\alpha_{-k} = a_k + ib_k$. Let us rewrite the complex exponential Fourier series as

$$\begin{aligned} f(t) &= \sum_{k=-\infty}^{\infty} \alpha_k e^{ik\pi t/T} \\ &= \alpha_0 + \sum_{k=1}^{\infty} \left(\alpha_k e^{ik\pi t/T} + \alpha_{-k} e^{-ik\pi t/T}\right) \\ &= a_0 + \sum_{k=1}^{\infty} \left((a_k - ib_k)e^{ik\pi t/T} + (a_k + ib_k)e^{-ik\pi t/T}\right) \\ &= a_0 + \sum_{k=1}^{\infty} \left(a_k\left(e^{ik\pi t/T} + e^{-ik\pi t/T}\right) - ib_k\left(e^{ik\pi t/T} - e^{-ik\pi t/T}\right)\right) \\ &= a_0 + 2\sum_{k=1}^{\infty} (a_k\cos(k\pi t/T) + b_k\sin(k\pi t/T)) \end{aligned} \tag{1.51}$$

where we've made use of equation (1.13). If f is real-valued, then the Fourier series with respect to the sine/cosine basis can be computed from (1.50) and (1.51) with no reference to complex numbers.

Fourier expansions can be performed on any interval $[a, b]$, by translating and scaling the functions $e^{i\pi kt/T}$ appropriately. The linear mapping $\phi : t \to T(2t - (a + b))/(b - a)$ takes the interval $[a, b]$ to $[-T, T]$. As a consequence we can check that the functions

$$e^{i\pi k\phi(t)/T} = e^{ik\pi(2t-(a+b))/(b-a)}$$

are orthogonal on $[a, b]$ with respect to the $L^2(a, b)$ inner product, and so form a basis.

Fourier Series on Rectangles The set of functions $\phi_{k,m}(x, y)$, $k \in \mathbb{Z}$, and $m \in \mathbb{Z}$ defined by

$$\phi_{k,m}(x, y) = e^{2\pi i(kx/a+my/b)}$$

forms an orthogonal basis for the set $C(\Omega)$ where $\Omega = \{(x, y); 0 \le x \le a, 0 \le y \le b\}$. As in the one-dimensional case, we can also write out an equivalent basis of sines and cosines.

Orthogonal bases can be proved to exist for more general two-dimensional regions, or regions in higher dimensions, but they cannot usually be written out explicitly.

1.10.4 Hilbert Spaces and $L^2(a, b)$

The inner product space of continuous, square-integrable functions on an interval (a, b) was defined in Example 1.15. This space is satisfactory for some purposes, but in many cases it's helpful to enlarge the space to contain discontinuous functions. The inclusion of discontinuous functions brings with it certain technical difficulties that, in full generality and rigor, require some sophisticated analysis to resolve. This analysis is beyond the scope of this text. Nonetheless, we give below a brief sketch of the difficulties, the resolution, and a useful fact concerning orthogonal bases.

Expanding the Space of Functions In our Fourier series examples above we worked in inner product spaces that consist of continuous functions. However, the definition of the inner product

$$(f, g) = \int_a^b f(t)\overline{g(t)}\,dt$$

requires much less of the functions involved. Indeed we might want to write out Fourier expansions for functions that have discontinuities. Such expansions can be a very useful, so it makes sense to enlarge this inner product space to include more than just continuous functions.

The Cauchy–Schwarz inequality indicates the direction we should move. The inner product (f, g) of two functions is guaranteed to be finite if the L^2 norms $\|f\|$

and $\|g\|$ are both finite. This was exactly the condition imposed in Example 1.8. We will thus enlarge our inner product space to include all functions f that satisfy

$$\int_a^b |f(t)|^2 \, dt < \infty. \tag{1.52}$$

This guarantees that the inner product (f, g) is defined for all pairs $f, g \in V$. However, equation (1.52) presumes that the function $|f(t)|^2$ can be meaningfully integrated using the Riemann integral. This may seem like a technicality, but while continuous functions are automatically Riemann integrable, arbitrary functions are not. We thus require Riemann integrability of f (which ensures that $|f|^2$ is Riemann integrable). Our notion of $L^2(a, b)$ is thus enlarged to include many types of discontinuous functions.

Complications This enlargement of the space creates a problem of its own though. Consider a function f that is zero at all but one point in $[a, b]$. Such a function, as well as $|f(t)|^2$, is Riemann integrable, and indeed the integral of $|f(t)|^2$ will be zero. In short, $\|f\| = 0$ and also $(f, f) = 0$, even though f is not identically zero. This violates property 1 for the norm and property 3 for the inner product. More generally, functions f and g that differ from each other at only finitely many (perhaps even more) points satisfy $\|f - g\| = 0$, even though $f \neq g$. Enlarging our inner product space has destroyed the inner product structure and notion of distance! This wasn't a problem when the functions were assumed to be continuous; recall Example 1.15.

The fix for this problem requires us to redefine what we mean by "$f = g$" in $L^2(a, b)$; recall the remark at the end of Example 1.3. Functions f_1 and f_2 that differ at sufficiently few points are considered identical under an equivalence relation "$\sim$," where "$f_1 \sim f_2$" means $\int |f_1 - f_2|^2 \, dt = 0$. The elements of the space of $L^2(a, b)$ thus consist of equivalence classes of functions that differ at sufficiently few points (just as the rational numbers consist of "fractions," but with a_1/b_1 and a_2/b_2 identified under an equivalence relation, namely, $a_1/b_1 \sim a_2/b_2$ meaning $a_1 b_2 = a_2 b_1$; thus $\frac{1}{2}$, $\frac{2}{4}$, and $\frac{3}{6}$ are all the same rational number). As a consequence functions in $L^2(a, b)$ do not have well-defined point values. When we carry out computations in $L^2(a, b)$, it is almost always via integration or inner products.

These changes also require some technical modifications to our notion of integration. It turns out Riemann integration is not sufficient for the task, especially when we need to deal with sequences or series of functions in $L^2(a, b)$. The usual approach requires replacing the Riemann integral with the Lebesgue integral, a slightly more versatile approach to integration. The modifications also have the happy consequence of "completing" the inner product space by guaranteeing that Cauchy sequences in the space converge to a limit within the space, which greatly simplifies the analysis. A complete inner product space is known as a *Hilbert space*. The Hilbert space obtained in this situation is known as $L^2(a, b)$. These spaces play an important role in applied mathematics and physics.

Despite all of these complications, we will use the notation $L^2(a, b)$ for the set of all Riemann integrable functions that satisfy inequality (1.52). Indeed, if we limit our attention to the subset of piecewise continuous functions, then real no difficulties arise. For the details of the full definition of $L^2(a, b)$ the interested reader can refer to [20] or any advanced analysis text.

A Converse to Parseval Parseval's identity (1.46) shows that if ϕ_k, $k \geq 0$, is an orthonormal basis for $L^2(a, b)$ and $f \in L^2(a, b)$, then f can be expanded as $f = \sum_k c_k \phi_k$ and $\sum_k |c_k|^2 = \|f\|^2 < \infty$. That is, every function in $L^2(a, b)$ generates a sequence $\mathbf{c} = (c_0, c_1, \ldots)$ in $L^2(\mathbb{N})$. In our new and improved space $L^2(a, b)$ it turns out that the converse is true: if we choose any sequence $\mathbf{c} = (c_0, c_1, \ldots)$ with $\sum_k c_k^2 < \infty$, then the sum

$$\sum_{k=0}^{\infty} c_k \phi_k \tag{1.53}$$

converges to some function $f \in L^2(a, b)$, in the sense that

$$\lim_{n \to \infty} \left\| f - \sum_{k=0}^{n} c_k \phi_k \right\| = 0.$$

In short, the functions in $L^2(a, b)$ and sequences in $L^2(\mathbb{N})$ can be matched up one to one (the correspondence depends on the basis ϕ_k). This is worth stating as a theorem.

Theorem 1.10.2 *Let ϕ_k, $k \geq 0$, be an orthonormal basis for $L^2(a, b)$. There is an invertible linear mapping $\Phi : L^2(a, b) \to L^2(\mathbb{N})$ defined by*

$$\Phi(f) = \mathbf{c},$$

where $\mathbf{c} = (c_0, c_1, \ldots)$ with $c_k = (f, \phi_k)$. Moreover $\|f\|_{L^2(a,b)} = \|\mathbf{c}\|_{L^2(\mathbb{N})}$ (the mapping is an "isometry," that is, length-preserving.)

1.11 MATLAB PROJECT

This section is designed as a series of Matlab explorations that allow the reader to play with some of the ideas in the text. It also introduces a few Matlab commands and techniques we'll find useful later. Matlab commands that should executed are in a bold `typewriter font` and usually displayed prominently.

1. Start Matlab. Most of the computation in the text is based on indexing vectors beginning with index 0, but Matlab indexes vectors from 1. This won't usually cause any headaches, but keep it in mind.
2. Consider sampling the function $f(t) = \sin(2\pi(440)t)$ on the interval $0 \leq t < 1$, at 8192 points (sampling interval $\Delta T = 1/8192$) to obtain samples

$f_k = f(k\Delta T) = \sin(2\pi(440)k/8192)$ for $0 \leq k \leq 8191$. The samples can be arranged in a vector `f`. You can do this in Matlab with

```
f = sin(2*pi*440/8192*(0:8191));
```

Don't forget the semicolon or Matlab will print out `f`!

The sample vector `f` is stored in double precision floating point, about 15 significant figures. However, we'll consider `f` as not yet quantized. That is, the individual components f_k of `f` can be thought of as real numbers that vary continuously, since 15 digits is pretty close to continuous for our purposes.

a. What is the frequency of the sinewave $\sin(2\pi(440)t)$, in Hertz?

b. Plot the sampled signal with the command `plot(f)`. It probably doesn't look too good, as it goes up and down 440 times in the plot range. You can plot a smaller range, say the first 100 samples, with `plot(f(1:100))`.

c. At the sampling rate 8192 Hertz, what is the Nyquist frequency? Is the frequency of $f(t)$ above or below the Nyquist frequency?

d. Type `sound(f)` to play the sound out of the computer speaker. By default, Matlab plays all sound files at 8192 samples per second, and assumes the sampled audio signal is in the range -1 to 1. Our signal satisfies these conditions.

e. As an example of aliasing, consider a second signal $g(t) = \sin(2\pi(440 + 8192)t)$. Repeat parts (a) through (d) with sampled signal

```
g = sin(2*pi*(440+8192)/8192*(0:8191));
```

The analog signal $g(t)$ oscillates much faster than $f(t)$, and we could expect it to yield a higher pitch. However, when sampled at frequency 8192 Hertz, $f(t)$ and $g(t)$ are aliased and yield precisely the same sampled vectors **f** and **g**. They should sound the same too.

f. To illustrate the effect of quantization error, let us construct a 2-bit (4 quantization levels) version of the audio signal $f(t)$ as in the scheme of Example 1.26. With that notation we have minimum value $m = -1$ and maximum value $M = 1$ for our signal, with $r = 4$. The command

```
qf = ceil(2*(f+1))-1;
```

produces the quantized signal $q($`f`$)$. Sample values of $f(t)$ in the ranges $(-1, -0.5]$, $(-0.5, 0]$, $(0, 0.5]$, and $(0.5, 1]$ are mapped to the integers $0, 1, 2, 3$, respectively.

To approximately reconstruct the quantized signal, we apply the dequantization formula to construct $\widetilde{\texttt{f}}$ as

```
ftilde = -1 + 0.5*(qf+0.5);
```

This maps the integers 0, 1, 2, and 3 to values -0.75, -0.25, 0.25, and 0.75, respectively (the codewords in this scheme).

g. Plot the first hundred values of $\widetilde{\texttt{f}}$ with `plot(ftilde(1:100));`. Play the quantized signal with `sound(ftilde);`. It should sound harsh compared to `f`.

h. Compute the distortion (as a percentage) of the quantized/dequantized signal using equation (1.39). In Matlab this is implemented as

```
100*norm(f-ftilde)^2/norm(f)^2
```

The `norm` command computes the standard Euclidean norm of a vector.

i. Repeat parts (f) through (h) using 3-,4-,5-,6-,7-, and 8-bit quantization. For example, 5-bit quantization is accomplished with `qf=ceil (16*(f+1))-1`, dequantization with `ftilde=-1+(qf+0.5)/16`. Here $16 = 2^{5-1}$. Make sure to play the sound in each case. Make up a table showing the number of bits in the quantization scheme, the corresponding distortion, and your subjective rating of the sound quality.

At what point can your ear no longer distinguish the original audio signal from the quantized version?

3. Type `load('splat')`. This loads in an audio signal sampled at 8192 samples per second. By default the signal is loaded into a vector “`y`” in Matlab, while the sampling rate is loaded into a variable “`Fs`”. Execute the Matlab command `whos` to verify this (or look at the workspace window that shows the currently defined variables).

a. Play the sampled signal with `sound(y)`.

b. Plot the sampled signal. Based on the size of `y` and sample rate 8192 Hertz, what is T, the length (in seconds) of the sampled sound?

c. The audio signal was sampled at frequency 8192. We can mimic a lower sampling rate $8192/m$ by taking every mth entry of the original vector `y`; this is called *downsampling*. In particular, let’s try $m = 2$. Execute `y2 = y(1:2:10001);`. The downsampled vector `y2` is the same sound, but sampled at frequency $8192/2 = 4096$ Hertz.

Play the downsampled `y2` with `sound(y2,4096)`. The second argument to the `sound` command indicates the sound should be played back at the corresponding rate (in Hertz).

d. Comment: why does the whistling sound in the original audio signal fall steadily in pitch, while the downsampled version seems to rise and fall?

4. You can clear from Matlab memory all variables defined to this point with the command `clear`. Do so now.

a. Find an image (JPEG will do) and store it in Matlab’s current working directory. You can load the image into Matlab with

```
z = imread('myimage.jpg');
```

(change the name to whatever your image is called!) If the image is color, the “`imread`” command automatically converts the $m \times n$ pixel image (in whatever conventional format it exists) into three $m \times n$ arrays, one for each of red, blue, and green as in Section 1.3.5. Each array consists of unsigned eight-bit integers, that is, integers in the range 0 to 255. Thus the variable z is now an $m \times n \times 3$ array. If the image is grayscale you'll get only one array.

b. The command

```
image(z);
```

displays the image in color, under the convention that `z` consists of unsigned integers and the colors are scaled in the 0 to 255 range. If we pass an $m \times n$ by 3 array of floating point numbers to the `image` command, it is assumed that each color array is scaled from 0.0 to 1.0. Consult the help page for the "image" command for more information.

c. A simple way to construct an artificial grayscale image is by picking off one of the color components and using it as a grayscale intensity, for example, `zg = double(z(:,:,1));`. However, a slightly more natural result is obtained by taking a weighted average of the color components, as

```
zg = 0.2989*double(z(:,:,1))
+0.5870*double(z(:,:,2))+0.1140*double(z(:,:,3));
```

The `double` command indicates that the array should be converted from unsigned integers to double precision floating point numbers, for example, 13 becomes 13.0. It's not strictly necessary unless we want to do floating point arithmetic on `zg` (which we do). The weighting coefficients above stem from the NTSC (television) color scheme; see [14] for more information. Now we set up an appropriate grayscale color map with the commands

```
L = 255;
colormap([(0:L)/L; (0:L)/L; (0:L)/L]');
```

Type `help colormap` for more information on this command. Very briefly, the array that is passed to the colormap command should have three columns, any number of rows. The three entries in the kth row should be scaled in the 0.0 to 1.0 range, and indicate the intensity of red, green, and blue that should be displayed for a pixel that is assigned integer value k. In our `colormap` command above, the kth row consists of elements $((k-1)/255, (k-1)/255, (k-1)/255)$, which correspond to a shade of gray (equal amounts of red, blue, and green), with $k = 0$ as black and $k = 255$ as white.

Now display the grayscale image with

```
image(zg);
```

It's not necessary to execute the `colormap` command prior to displaying every image—once will do, unless you close the display window, in which case you must re-initialize the color map.

d. The image is currently quantized at eight bit precision (each pixel's graylevel specified by one of 0, 1, 2, . . . , 255). We can mimic a cruder quantization level, say six-bit, with

```
qz = 4*floor(zg/4);
```

followed by `image(qz);`. This command has the effect of rounding each entry of `zg` to the next lowest multiple of 4, so each pixel is now encoded as one of the 64 numbers 0, 4, 8, . . . , 252. Can you tell the difference in the image?

Compute the percent distortion introduced with

```
100*norm(zg-qz,'fro').^2/norm(zg,'fro').^2
```

The "`fro`" argument indicates that the Frobenius norm of Example 1.14 should be used for the matrices, that is, take the square root of the sum of the squares of the matrix entries.

Repeat the above computations for other b-bit quantizations with $b = 1, 2, 3, 4, 5$. Display the image in case, and compute the distortion. At what point does the quantization become objectionable?

e. We can add noise to the image with

```
zn = zg + 50*(rand(size(zg))-0.5);
```

which should, of course, be followed by `image(zn);`. The `size` command returns the dimensions of the matrix `zg`, and the `rand` command generates a matrix of that size consisting of uniformly distributed random numbers (double precision floats) on the interval 0 to 1. Thus `50*(rand(size(zg))-0.5)` yields an array of random numbers in the range −25 to 25 that is added to `zg`. Any values in `zn` that are out of range (less than 0 or greater than 255) are "clipped" to 0 or 255, respectively.

5. This exercise illustrates aliasing for images. First, clear all variables with `clear`, and then execute the commands `L = 255;` and

```
colormap([(0:L)/L;(0:L)/L;(0:L)/L]');
```

as in the previous exercise.

Let's sample and display an analog image, say

$$f(x, y) = 128(1 + \sin(2\pi(20)x)\sin(2\pi(30)y)),$$

on the square $0 \leq x, y \leq 1$; we've chosen f to span the range 0 to 256. We will sample on an m by n grid for various values of m and n. This can be accomplished with

```
m = 50; X = [0:m-1]/m;
n = 50; Y = [0:n-1]/n;
f = 128⋆(1 + sin(2⋆pi⋆30*Y)'⋆sin(2⋆pi⋆20⋆X));
```

The first portion `sin(2⋆pi⋆30⋆Y)'` are the y values of the grid points (as a column vector) and `sin(2⋆pi⋆20⋆X)` are the x values (as a row vector). Plot the image with `image(f)`.

Try various values of m and n from 10 to 500. Large values of m and n should produce a more "faithful" image, while small values should produce obvious visual artifacts; in particular, try $m = 20$, $n = 30$. The effect is highly dependent on screen resolution.

EXERCISES

JPEG Compression

1.1 Find at least five different color JPEG images on a computer (or with a Web browser); they'll have a "`.jpg`" suffix on the file name. Try to find a variety of images, for example, one that is "mostly dark" (astronomical images are a good bet). For each image determine the following:

- Its pixel by pixel size—how many pixels wide, how many tall. This can usually be determined by right mouse clicking on the image and selecting "Properties."
- The memory storage requirements for the image if it was stored "naively" as in the example in the text above.
- The actual memory storage requirement (the file size).
- The ratio of the actual and "naive" storage requirements.

Summarize your work in a table. Note the range of compression ratios obtained. Can you detect any correlation between the nature or quality of the images and the compression ratios achieved?

Complex Numbers

1.2 Use Euler's identity (1.11) to prove that if x is real, then

- $e^{-ix} = \overline{e^{ix}}$,
- $e^{2\pi ix} = 1$ if and only if x is an integer.

1.3 The complex number z is an Nth root of unity if and only if $z^N = 1$. Draw the eighth roots of unity on the unit circle.

1.4 Prove that the entries of $\mathbf{E}_{N,k}$ are Nth roots of unity (see Exercise 1.3).

1.5 Suppose that

$$\begin{aligned} x(t) &= a\cos(\omega t) + b\sin(\omega t) \\ &= ce^{i\omega t} + de^{-i\omega t} \end{aligned}$$

for all real t. Show that

$$a = c + d, \qquad b = ic - id,$$
$$c = \frac{a - ib}{2}, \quad d = \frac{a + ib}{2}.$$

Sampling and Quantization

1.6 Let $x(t) = 1.3t$ and $y(t) = \sin(\frac{\pi}{2}t)$ be analog signals on the interval $0 \leq t \leq 1$.

a. Sample $x(t)$ at times $t = 0, 0.25, 0.5, 0.75$ to produce sampled vector $\mathbf{x} = (x(0), x(0.25), x(0.5), x(0.75)) \in \mathbb{R}^4$. Sample $y(t)$ at the same times to product vector $\mathbf{y} \in \mathbb{R}^4$.

Verify that the sampled version (same times) of the analog signal $x(t) + y(t)$ is just $\mathbf{x} + \mathbf{y}$ (this should be painfully clear).

b. Let q denote a function that takes any real number r and rounds it to the nearest integer, a simple form of quantization. Use q to quantize $\mathbf{x}$ from part (a) component by component, to produce a quantized vector $q(\mathbf{x}) = (q(x_0), q(x_1), q(x_2), q(x_3))$. Do the same for $\mathbf{y}$ and $\mathbf{x} + \mathbf{y}$.

Show that $q(\mathbf{x}) + q(\mathbf{y}) \neq q(\mathbf{x} + \mathbf{y})$, and also that $q(2\mathbf{x}) \neq 2q(\mathbf{x})$. Quantization is a nonlinear operation!

Vector Spaces

1.7 Is the set of all quadratic polynomials in x with real-valued coefficients (with polynomial addition and scalar multiplication defined in the usual way) a vector space over $\mathbb{R}$? Why or why not? (Consider something like $a_0 + a_1x + 0x^2$ a quadratic polynomial.)

1.8 Is the set of all continuous real-valued functions $f(x)$ defined on [0, 1] that satisfy

$$\int_0^1 f(x)\,dx = 3$$

a vector space over $\mathbb{R}$? Assume function addition and scalar multiplication are defined as usual.

1.9 Clearly, $\mathbb{R}^n$ is a subset of $\mathbb{C}^n$. Is $\mathbb{R}^n$ a subspace of $\mathbb{C}^n$ (where $\mathbb{C}^n$ is considered a vector space over $\mathbb{C}$)?

1.10 Verify that the set in Example 1.3 with given operations is a vector space.

1.11 Verify that the set $L^2(\mathbb{N})$ in Example 1.5 with given operations is a vector space. *Hint:* This closely parallels Example 1.8, with summation in place of integrals.

Explain why $L^2(\mathbb{N})$ is a subspace of $L^\infty(\mathbb{N})$.

1.12 The point of this exercise is to prove the assertions in Proposition 1.4.1 for an abstract vector space.

a. Show that the $\mathbf{0}$ vector is unique. To do this, suppose that there are two vectors, $\mathbf{0}_1$ and $\mathbf{0}_2$, both of which play the role of the zero vector. Show that $\mathbf{0}_1 = \mathbf{0}_2$. *Hint:* Consider $\mathbf{0}_1 + \mathbf{0}_2$.

b. Below is a proof that $0\mathbf{u} = \mathbf{0}$ in any vector space. In this proof $-\mathbf{u}$ denotes the additive inverse for $\mathbf{u}$, so $\mathbf{u} + (-\mathbf{u}) = \mathbf{0}$. What property or properties of the eight listed in Definition 1.4.1 justifies each step?

$$\begin{aligned}
(1+0)\mathbf{u} &= 1\mathbf{u} + 0\mathbf{u} \\
1\mathbf{u} &= \mathbf{u} + 0\mathbf{u} \\
\mathbf{u} &= \mathbf{u} + 0\mathbf{u}, \\
\mathbf{u} + (-\mathbf{u}) &= (\mathbf{u} + (-\mathbf{u})) + 0\mathbf{u} \\
\mathbf{0} &= \mathbf{0} + 0\mathbf{u} \\
\mathbf{0} &= 0\mathbf{u}.
\end{aligned}$$

c. Show that if $\mathbf{u} + \mathbf{v} = \mathbf{0}$, then $\mathbf{v} = (-1)\mathbf{u}$ (this shows that the additive inverse of $\mathbf{u}$ is $(-1)\mathbf{u}$).

Basic Analog and Discrete Waveforms

1.13 Write out the basic waveforms $\mathbf{E}_{2,k}$ for $k = -2, -1, 0, 1, 2$, and verify that the resulting vectors are periodic with period 2 with respect to the index k.

Repeat for $\mathbf{E}_{3,k}$ (same k range). Verify periodicity with period 3.

1.14 Let $a = re^{i\theta}$ be a complex number (where $r > 0$ and θ are real).

a. Show that the function $f(t) = ae^{i\omega t}$ satisfies $|f(t)| = r$ for all t.

b. Show that $f(t) = ae^{i\omega t}$ is shifted a fraction θ/ω cycles or periods to the left, compared to $|r|e^{i\omega t}$.

1.15 Show that each of the four types of waveforms in (1.19) can be expressed as a linear combination of waveforms $e^{\pm i\alpha x \pm i\beta y}$ of the form (1.17).

1.16 Let $\mathbf{C}_k$ be the vector obtained by sampling the function $\cos(2\pi kt)$ at the points $t = 0, 1/N, 2/N, \ldots, (N-1)/N$, and let $\mathbf{S}_k$ be similarly defined with respect to the sine function. Prove the following vector analogs of the equations (1.12), (1.13), and (1.14) relating the exponential and trigonometric wave forms.

$$\begin{aligned}
&\mathbf{E}_k = \mathbf{C}_k + i\mathbf{S}_k,\ \overline{\mathbf{E}_k} = \mathbf{C}_k - i\mathbf{S}_k,\\
&\mathbf{C}_k = \frac{1}{2}(\mathbf{E}_k + \overline{\mathbf{E}_k}),\ \mathbf{S}_k = \frac{1}{2i}(\mathbf{E}_k - \overline{E_k}),\\
&\mathbf{C}_k = \mathrm{Re}(\mathbf{E}_{\mathrm{k}}),\ \mathbf{S}_{\mathrm{k}} = \mathrm{Im}(\mathbf{E}_{\mathrm{k}}),
\end{aligned}$$

where $\mathbf{E}_k = \mathbf{E}_{N,k}$ is as defined in equation (1.22).

1.17 Show that we can factor the basic two-dimensional waveform $\mathcal{E}_{m,n,k,l}$ as

$$\mathcal{E}_{m,n,k,l} = \mathbf{E}_{m,k}\mathbf{E}_{n,l}^T$$

(recall superscript T denotes the matrix/vector transpose operation), where the vectors $\mathbf{E}_{m,k}$ and $\mathbf{E}_{m,k}$ are the discrete basic waveforms in one-dimension as defined in equation (1.22), as column vectors.

1.18 Consider an exponential waveform

$$f(x, y) = e^{2\pi i(px+qy)}$$

as was discussed in Section 1.5.2 (p and q need not be integers). Figure 1.7 in that section indicates that this waveform has a natural "direction" and "wavelength." The goal of this problem is to understand the sense in which this is true, and how these quantities depend on p and q.

Define $\mathbf{v} = (p, q)$, so $\mathbf{v}$ is a two-dimensional vector. Consider a line L through an arbitrary point (x_0, y_0) in the direction of a unit vector $\mathbf{u} = (u_1, u_2)$ (so $\|\mathbf{u}\| = 1$). The line L can be parameterized with respect to arc length as

$$x(t) = x_0 + tu_1, \quad y(t) = y_0 + tu_2.$$

a. Show that the function $g(t) = f(x(t), y(t))$ with $x(t)$, $y(t)$ as above (i.e., f evaluated along the line L) is given by

$$g(t) = Ae^{2\pi i\|\mathbf{v}\|\cos(\theta)t},$$

where A is some complex number that doesn't depend on t and θ is the angle between $\mathbf{v}$ and $\mathbf{u}$. *Hint:* Use equation (1.30).

b. Show that if L is orthogonal to $\mathbf{v}$, then the function g (and so f) remains constant.

c. Find the frequency (oscillations per unit distance moved) of g as a function of t, in terms of p, q, and θ.

d. Find the value of θ that maximizes the frequency at which $g(t)$ oscillates. This θ dictates the direction one should move, relative to $\mathbf{v}$ so that f

oscillates as rapidly as possible. How does this value of θ compare to the θ value in question (b)? What is this maximal frequency of oscillation, in terms of p and q?

e. Find the "peak-to-peak" distance or wavelength of the waveform $f(x, y)$, in terms of p and q.

Aliasing and Basic Waveforms

1.19 Write out the vectors $\mathbf{E}_{6,0}, \mathbf{E}_{6,1}, \ldots, \mathbf{E}_{6,5}$ as in Section 1.7.1. Determine all aliasing relations or redundancies (including conjugate aliasing) you can from the chart. (Remember to index the vector components from 0 to 5.)

1.20 For a pure $1D$ wave form of N samples prove the aliasing relation

$$\mathbf{E}_{N-k} = \overline{\mathbf{E}_k}.$$

1.21 Find all the aliasing relations you can (including conjugate aliasing) for $\mathcal{E}_{m,n,k,l}$. This can be done directly, or you might use equation (1.26) and the aliasing relations for the $\mathbf{E}_{N,k}$.

Inner Products and Norms

1.22 Let $S = \{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$, where $\mathbf{v}_1 = (1, 1, 0), \mathbf{v}_2 = (-1, 1, 1)$, and $\mathbf{v}_3 = (1, -1, 2)$ are vectors in $\mathbb{R}^3$.

a. Verify that S is orthogonal with respect to the usual inner product. This shows that S must be a basis for $\mathbb{R}^3$.

b. Write the vector $\mathbf{w} = (3, 4, 5)$ as a linear combination of the basis vectors in S. Verify that the linear combination you obtain actually reproduces $\mathbf{w}$!

c. Rescale the vectors in S as per Remark 1.11 on page 39 to produce an equivalent set S' of orthonormal vectors.

d. Write the vector $\mathbf{w} = (3, 4, 5)$ as a linear combination of the basis vectors in S'.

e. Use the results of part (d) to check that Parseval's identity holds.

1.23 Let $S = \{\mathbf{E}_{4,0}, \mathbf{E}_{4,1}, \mathbf{E}_{4,2}, \mathbf{E}_{4,3}\}$ (these vectors are written out explicitly just prior to equation (1.24)). The set S is orthogonal and a basis for $\mathbb{R}^4$.

a. Use Theorem 1.8.3 to write the vector $\mathbf{v} = (1, 5, -2, 3)$ as a linear combination of the basis vectors in S.

b. Rescale the vectors in S as per Remark 1.11 on page 39 to produce an equivalent set S' of orthonormal vectors.

c. Write the vector $\mathbf{v} = (1, 5, -2, 3)$ as a linear combination of the basis vectors in S'.

d. Use the results of part (c) to check that Parseval's identity holds.

1.24 There are infinitely many other inner products on $\mathbb{R}^n$ besides the standard dot product, and they can be quite useful too.

Let $\mathbf{d} = (d_1, d_2, \ldots, d_n) \in \mathbb{R}^n$. Suppose that $d_k > 0$ for $1 \leq k \leq n$.

a. Let $\mathbf{v} = (v_1, v_2, \ldots, v_n)$ and $\mathbf{w} = (w_1, w_2, \ldots, w_n)$ be vectors in $\mathbb{R}^n$. Show that the function

$$(\mathbf{v}, \mathbf{w})_d = \sum_{k=1}^{n} d_k v_k w_k$$

defines an inner product on $\mathbb{R}^n$. Write out the corresponding norm.

b. Let $\mathbf{d} = (1, 5)$ in $\mathbb{R}^2$, and let $S = \{\mathbf{v}_1, \mathbf{v}_2\}$ with $\mathbf{v}_1 = (2, 1)$, $\mathbf{v}_2 = (5, -2)$. Show that S is orthogonal with respect to the $(,)_d$ inner product.

c. Find the length of each vector in S with respect to the norm induced by this inner product.

d. Write the vector $\mathbf{w} = (-2, 5)$ as a linear combination of the basis vectors in S. Verify that the linear combination you obtain actually reproduces $\mathbf{w}$!

1.25 Let $\mathbf{v}$ and $\mathbf{w}$ be elements of a normed vector space. Prove the reverse triangle inequality,

$$|\|\mathbf{v}\| - \|\mathbf{w}\|| \leq \|\mathbf{v} - \mathbf{w}\|. \tag{1.54}$$

Hint: Start with $\mathbf{v} = (\mathbf{v} - \mathbf{w}) + \mathbf{w}$, take the norm of both sides, and use the usual triangle inequality.

1.26 Let $d(t)$ be a real-valued, positive, continuous function that is bounded away from 0 on an interval $[a, b]$; that is, $d(t) \geq \delta > 0$ for some δ and all $t \in [a, b]$. Verify that

$$(f, g)_d = \int_a^b d(t) f(t) \overline{g(t)}\, dt$$

defines an inner product on $C[a, b]$ (with complex-valued functions). Write out the corresponding norm.

1.27 Suppose that V is an inner product space with inner product $(\mathbf{v}, \mathbf{w})$. Show that if we define

$$\|\mathbf{v}\| = \sqrt{(\mathbf{v}, \mathbf{v})},$$

then $\|\mathbf{v}\|$ satisfies the properties of a norm. *Hints:* All the properties are straightforward, except the triangle inequality. To show this, note that

$$\|\mathbf{v} + \mathbf{w}\|^2 = \|\mathbf{v}\|^2 + \|\mathbf{w}\|^2 + (\mathbf{v}, \mathbf{w}) + (\mathbf{w}, \mathbf{v}).$$

Apply the Cauchy-Schwarz inequality to both inner products on the right side above, and note that for any $z \in \mathbb{C}$ we have $|\mathrm{Re(z)}| \leq |\mathrm{z}|$.

1.28 Suppose that V is an inner product space, with a norm $\|\mathbf{v}\| = \sqrt{(\mathbf{v}, \mathbf{v})}$ that comes from the inner product.

a. Show that this norm must satisfy the *parallelogram* identity

$$2\|\mathbf{u}\|^2 + 2\|\mathbf{v}\|^2 = \|\mathbf{u} + \mathbf{v}\|^2 + \|\mathbf{u} - \mathbf{v}\|^2.$$

b. Let $f(x) = x$ and $g(x) = x(1 - x)$ be elements of the normed vector space $C[0, 1]$ with the supremum norm. Compute each of $\|f\|_\infty$, $\|g\|_\infty$, $\|f + g\|_\infty$, and $\|f - g\|_\infty$, and verify that the parallelogram identity from part (a) does not hold. Hence the supremum norm cannot come from an inner product.

1.29 Suppose that S is an orthogonal but not orthonormal basis for $\mathbb{R}^n$ consisting of vectors $\mathbf{v}_k$, $1 \leq k \leq n$. Show that Parseval's identity becomes

$$\|\mathbf{v}\|^2 = \sum_{k=1}^{n} |\alpha_k|^2 \|\mathbf{v}_k\|^2.$$

where $\mathbf{v} = \sum_{k=1}^{n} \alpha_k \mathbf{v}_k$. *Hint:* Just chase through the derivation of equation (1.36).

1.30 For the basic waveforms defined by equation (1.22) show that

$$(\mathbf{E}_{N,k}, \mathbf{E}_{N,k}) = N$$

with the inner product defined in Example 1.13.

1.31 For 2D wave forms which are $m \times n$ matrices defined by equation (1.25) prove that

$$(\mathcal{E}_{k,l}, \mathcal{E}_{p,q}) = 0$$

when $k \neq p$ or $l \neq q$, and

$$(\mathcal{E}_{k,l}, \mathcal{E}_{k,l}) = mn$$

with the inner product on $M_{m,n}(\mathbb{C})$ defined in Example 1.14. It may be helpful to look at Example 1.20.

1.32 Let $\mathbf{v}_k$, $1 \leq k \leq n$, be any set of vectors in $\mathbb{C}^n$, considered as column vectors. Define the $n \times n$ matrix

$$\mathbf{A} = \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_n \end{bmatrix}.$$

Let $\mathbf{A}^* = (\overline{\mathbf{A}})^T$ (conjugate and transpose entries). What is the relationship between the entries $b_{m,k}$ of the $n \times n$ matrix $\mathbf{B} = (\mathbf{A}^*)(\mathbf{A})$ and the inner products $(\mathbf{v}_k, \mathbf{v}_m)$ (inner product defined as in Example 1.13)?

1.33 Use the result of the last exercise to show that if S is an orthonormal set of vectors $\mathbf{v}_k \in \mathbb{C}^n$, $1 \leq k \leq n$, and $\mathbf{A}$ is the matrix from the previous problem, then $(\mathbf{A}^*)(\mathbf{A}) = \mathbf{I}$, where $\mathbf{I}$ denotes the $n \times n$ identity matrix.

Infinite-Dimensional Inner Product Spaces and Fourier Series

1.34 Let ϕ_k, $1 \leq k < \infty$ be an orthonormal set in an inner product space V (no assumption that ϕ_k is complete). Let $f \in V$ and define $c_k = (f, \phi_k)$. Prove Bessel's inequality,

$$\sum_{k=1}^{\infty} |c_k|^2 \leq \|f\|^2. \tag{1.55}$$

Hint: Start with

$$0 \leq \left(f - \sum_{k=1}^{n} c_k \phi_k, f - \sum_{k=1}^{n} c_k \phi_k \right)$$

(first explain why this inequality is true).

1.35 Suppose that V is a normed vector space and $\mathbf{v}_n$ a sequence of elements in V that converge to $\mathbf{v} \in V$, that is,

$$\lim_{n\to\infty} \|\mathbf{v}_n - \mathbf{v}\| = 0.$$

Show that

$$\lim_{n\to\infty} \|\mathbf{v}_n\| = \|\mathbf{v}\|.$$

Hint: Use equation (1.54) from Exercise 1.25.

Show that the converse of this statement is false (provide a counterexample).

1.36 We can endow the vector space $L^2(\mathbb{N})$ in Example 1.5 (see also Exercise 1.11) with an inner product

$$(\mathbf{x}, \mathbf{y}) = \sum_{k=0}^{\infty} x_k y_k,$$

where $\mathbf{x} = (x_0, x_1, x_2, \ldots)$ and $\mathbf{y} = (y_0, y_1, y_2 \ldots)$ and all components are real-valued.

a. Verify that this really does define an inner product. In particular, you should first show that the inner product of any two elements is actually defined; that is, the infinite sum converges. (*Hint:* Use $|x_k y_k| \leq (x_k^2 + y_k^2)/2$ as in Example 1.8.)

What is the corresponding norm on $L^2(\mathbb{N})$?

b. Let $\mathbf{e}_k$ denote that element of $L^2(\mathbb{N})$ that has $x_k = 1$ and all other $x_m = 0$. Show that $S = \cup_{k=0}^{\infty} \mathbf{e}_k$ is an orthonormal set.

c. For an arbitrary $\mathbf{x} \in L^2(\mathbb{N})$, show that

$$\mathbf{x} = \sum_{k=0}^{\infty} \alpha_k \mathbf{e}_k$$

for a suitable choice of the α_k. *Hint:* This is very straightforward; just use Theorem 1.10.1.

1.37 Let V be an infinite-dimensional inner product space with orthonormal basis ϕ_k, $k \geq 1$. Suppose that $\mathbf{x}$ and $\mathbf{y}$ are elements of V and

$$\mathbf{x} = \sum_{k=1}^{\infty} a_k \phi_k \quad \mathbf{y} = \sum_{k=1}^{\infty} b_k \phi_k.$$

Of course, $a_k = (\mathbf{x}, \phi_k)$ and $b_k = (\mathbf{y}, \phi_k)$.

a. Define partial sums

$$\mathbf{x}_N = \sum_{k=1}^{N} a_k \phi_k \quad \mathbf{y}_N = \sum_{k=1}^{N} b_k \phi_k.$$

Show that

$$(\mathbf{x}_N, \mathbf{y}_N) = \sum_{k=1}^{N} a_k b_k.$$

b. Show that

$$(\mathbf{x}, \mathbf{y}) = \sum_{k=1}^{\infty} a_k b_k.$$

Hint: Note that for any N,

$$(\mathbf{x}, \mathbf{y}) - (\mathbf{x}_N, \mathbf{y}_N) = (\mathbf{x}, \mathbf{y} - \mathbf{y}_N) + (\mathbf{x} - \mathbf{x}_N, \mathbf{y}_N),$$

and of course, $\mathbf{x}_N \to \mathbf{x}$ and $\mathbf{y}_N \to \mathbf{y}$. Use the Cauchy–Schwarz inequality to show that the right side above goes to zero; then invoke part (a). The result of Exercise 1.35 may also be helpful.

1.38 Show that if $\mathbf{v}$ and $\mathbf{w}$ are nonzero vectors, then equality is obtained in the Cauchy–Schwarz inequality (i.e., $|(\mathbf{v}, \mathbf{w})| = \|\mathbf{v}\|\|\mathbf{w}\|$) if and only if $\mathbf{v} = c\mathbf{w}$ for some scalar c. (One direction is easy, the other is somewhat challenging.)

1.39 Let $\phi_k(t) = e^{i\pi kt}/\sqrt{2}$ for $k \in \mathbb{Z}$.

a. Verify that the ϕ_k form an orthonormal set in $C[-1, 1]$ (complex-valued functions) with the inner product defined in Example 1.15.

b. Find the Fourier coefficients α_k of the function $f(t) = t$ with respect to the ϕ_k explicitly in terms of k by using equation (1.48). *Hint:* It's a pretty easy integration by parts, and α_0 doesn't fit the pattern.

c. Use the result of part (b) to prove the amusing result that

$$\sum_{k=1}^{\infty} \frac{1}{k^2} = \frac{\pi^2}{6}.$$

1.40 Let $f(t)$ be a real-valued continuous function defined on an interval $[-T, T]$.

a. Show that if $f(t)$ is an even function ($f(-t) = f(t)$), then the Fourier series in (1.51) contains only cosine terms (and the constant term).

b. Show that if $f(t)$ is an odd function ($f(-t) = -f(t)$), then the Fourier series in (1.51) contains only sine terms.

1.41 Let $\phi_k(t) = \cos(k\pi t)$ for $k \geq 0$.

a. Verify that the ϕ_k are orthogonal on $[0, 1]$ with respect to the usual inner product (note $\phi_0(t) = 1$).

b. Show that this set of functions is complete as follows (where we'll make use of the given fact that the functions $e^{i\pi kt}$ are complete in $C[-1, 1]$): Let $f(t)$ be a function in $C[0, 1]$. We can extend f to an even function $\tilde{f}(t)$ in $C[-1, 1]$ as

$$\tilde{f}(t) = \begin{cases} f(t), & t \geq 0, \\ f(-t), & t < 0. \end{cases}$$

Now use the result of Exercise 1.40 to show that $\tilde{f}$ has a Fourier expansion in appropriate cosines on $[-1, 1]$. Why does this show that $\cos(k\pi t)$ is complete on $[0, 1]$?