Topics in Fourier Analysis: DFT & FFT, Wavelets, Laplace Transform

by Peter J. Olver
University of Minnesota

1. Introduction.

In addition to their inestimable importance in mathematics and its applications, Fourier series also serve as the entry point into the wonderful world of Fourier analysis and its wide-ranging extensions and generalizations. An entire industry is devoted to further developing the theory and enlarging the scope of applications of Fourier–inspired methods. New directions in Fourier analysis continue to be discovered and exploited in a broad range of physical, mathematical, engineering, chemical, biological, financial, and other systems. In this chapter, we will concentrate on four of the most important variants: discrete Fourier sums leading to the Fast Fourier Transform (FFT); the modern theory of wavelets; the Fourier transform; and, finally, its cousin, the Laplace transform. In addition, more general types of eigenfunction expansions associated with partial differential equations in higher dimensions will appear in the following chapters.

Modern digital media, such as CD's, DVD's and MP3's, are based on discrete data, not continuous functions. One typically samples an analog signal at equally spaced time intervals, and then works exclusively with the resulting discrete (digital) data. The associated discrete Fourier representation re-expresses the data in terms of sampled complex exponentials; it can, in fact, be handled by finite-dimensional vector space methods, and so, technically, belongs back in the linear algebra portion of this text. However, the insight gained from the classical continuous Fourier theory proves to be essential in understanding and analyzing its discrete digital counterpart. An important application of discrete Fourier sums is in signal and image processing. Basic data compression and noise removal algorithms are applied to the sample's discrete Fourier coefficients, acting on the observation that noise tends to accumulate in the high frequency Fourier modes, while most important features are concentrated at low frequencies. The first Section 2 develops the basic Fourier theory in this discrete setting, culminating in the Fast Fourier Transform (FFT), which produces an efficient numerical algorithm for passing between a signal and its discrete Fourier coefficients.

One of the inherent limitations of classical Fourier methods, both continuous and discrete, is that they are not well adapted to localized data. (In physics, this lack of localization is the basis of the Heisenberg Uncertainty Principle.) As a result, Fourier-based signal processing algorithms tend to be inaccurate and/or inefficient when confronting highly localized signals or images. In the second section, we introduce the modern theory of wavelets, which is a recent extension of Fourier analysis that more naturally incorporates

multiple scales and localization. Wavelets are playing an increasingly dominant role in many modern applications; for instance, the new JPEG digital image compression format is based on wavelets, as are the computerized FBI fingerprint data used in law enforcement in the United States.

The Laplace transform is a basic tool in engineering applications. To mathematicians, the Fourier transform is the more fundamental of the two, while the Laplace transform is viewed as a certain real specialization. Both transforms change differentiation into multiplication, thereby converting linear differential equations into algebraic equations. The Fourier transform is primarily used for solving boundary value problems on the real line, while initial value problems, particularly those involving discontinuous forcing terms, are effectively handled by the Laplace transform.

2. Discrete Fourier Analysis and the Fast Fourier Transform.

In modern digital media — audio, still images or video — continuous signals are sampled at discrete time intervals before being processed. Fourier analysis decomposes the sampled signal into its fundamental periodic constituents — sines and cosines, or, more conveniently, complex exponentials. The crucial fact, upon which all of modern signal processing is based, is that the sampled complex exponentials form an orthogonal basis. The section introduces the Discrete Fourier Transform, and concludes with an introduction to the Fast Fourier Transform, an efficient algorithm for computing the discrete Fourier representation and reconstructing the signal from its Fourier coefficients.

We will concentrate on the one-dimensional version here. Let $f(x)$ be a function representing the signal, defined on an interval $a \leq x \leq b$. Our computer can only store its measured values at a finite number of *sample points* $a \leq x_0 < x_1 < \cdots < x_n \leq b$. In the simplest and, by far, the most common case, the sample points are equally spaced, and so

$$x_j = a + j\,h, \qquad j = 0, \ldots, n, \qquad \text{where} \qquad h = \frac{b-a}{n}$$

indicates the sample rate. In signal processing applications, x represents time instead of space, and the x_j are the times at which we sample the signal $f(x)$. Sample rates can be very high, e.g., every 10–20 milliseconds in current speech recognition systems.

For simplicity, we adopt the "standard" interval of $0 \leq x \leq 2\pi$, and the n equally spaced sample points[†]

$$x_0 = 0, \qquad x_1 = \frac{2\pi}{n}, \qquad x_2 = \frac{4\pi}{n}, \quad \ldots \quad x_j = \frac{2j\pi}{n}, \quad \ldots \quad x_{n-1} = \frac{2(n-1)\pi}{n}. \tag{2.1}$$

(Signals defined on other intervals can be handled by simply rescaling the interval to have length 2π.) Sampling a (complex-valued) signal or function $f(x)$ produces the *sample vector*

$$\mathbf{f} = \big(\, f_0, f_1, \ldots, f_{n-1} \,\big)^T = \big(\, f(x_0), f(x_1), \ldots, f(x_{n-1}) \,\big)^T,$$

[†] We will find it convenient to omit the final sample point $x_n = 2\pi$ from consideration.

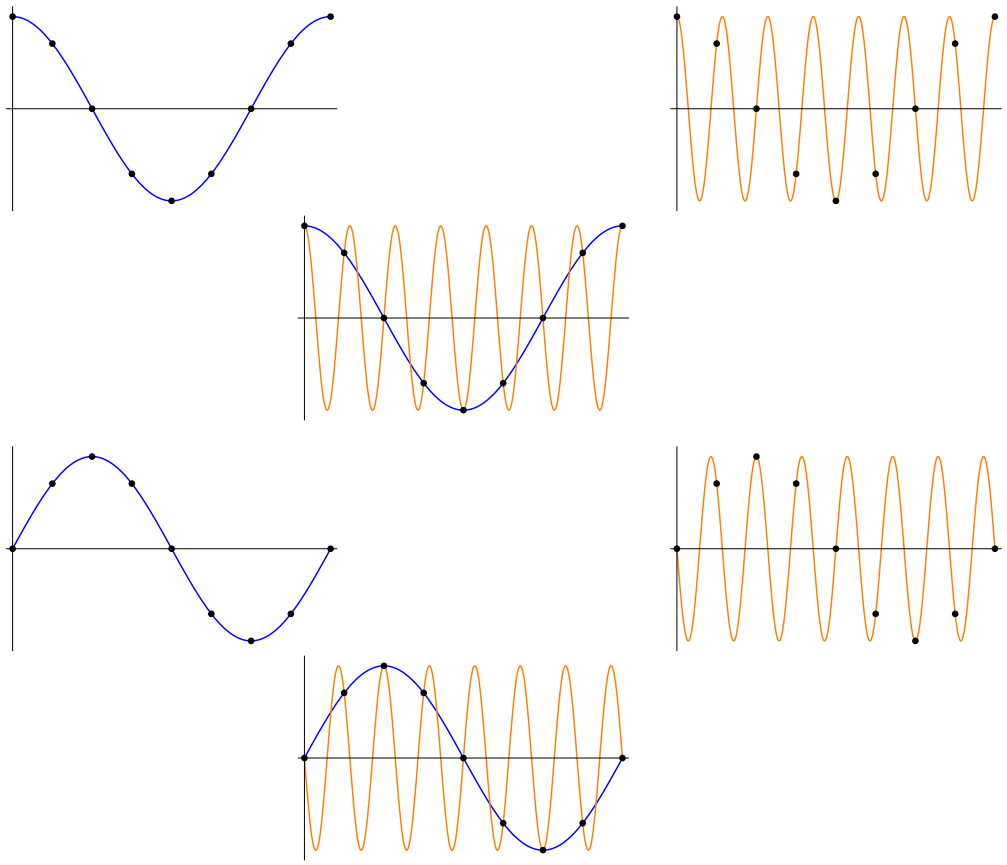

Figure 1. Sampling $e^{-\mathrm{i}\,x}$ and $e^{7\,\mathrm{i}\,x}$ on $n = 8$ sample points.

where

$$f_j = f(x_j) = f\left(\frac{2\,j\,\pi}{n}\right). \tag{2.2}$$

Sampling cannot distinguish between functions that have the same values at all of the sample points — from the sampler's point of view they are identical. For example, the periodic complex exponential function

$$f(x) = e^{\mathrm{i}\,n\,x} = \cos n\,x + \mathrm{i}\,\sin n\,x$$

has sampled values

$$f_j = f\left(\frac{2\,j\,\pi}{n}\right) = \exp\left(\mathrm{i}\,n\,\frac{2\,j\,\pi}{n}\right) = e^{2\,j\,\pi\,\mathrm{i}} = 1 \qquad \text{for all} \qquad j = 0, \dots, n-1,$$

and hence is indistinguishable from the constant function $c(x) \equiv 1$ — both lead to the *same* sample vector $(\,1, 1, \dots, 1\,)^T$. This has the important implication that sampling at n equally spaced sample points *cannot* detect periodic signals of frequency n. More generally, the two complex exponential signals

$$e^{\mathrm{i}\,(k+n)x} \qquad \text{and} \qquad e^{\mathrm{i}\,k\,x}$$

are also indistinguishable when sampled. This has the important consequence that we need only use the first n periodic complex exponential functions

$$f_0(x) = 1, \qquad f_1(x) = e^{\,\mathrm{i}\,x}, \qquad f_2(x) = e^{2\,\mathrm{i}\,x}, \qquad \dots \qquad f_{n-1}(x) = e^{(n-1)\,\mathrm{i}\,x}, \tag{2.3}$$

in order to represent any $2\,\pi$ periodic sampled signal. In particular, exponentials $e^{-\,\mathrm{i}\,k\,x}$ of "negative" frequency can all be converted into positive versions, namely $e^{\,\mathrm{i}\,(n-k)x}$, by the same sampling argument. For example,

$$e^{-\,\mathrm{i}\,x} = \cos x - \,\mathrm{i}\, \sin x \qquad \text{and} \qquad e^{(n-1)\,\mathrm{i}\,x} = \cos(n-1)\,x + \,\mathrm{i}\, \sin(n-1)\,x$$

have identical values on the sample points (2.1). However, off of the sample points, they are quite different; the former is slowly varying, while the latter represents a high frequency oscillation. In Figure 1, we compare $e^{-\,\mathrm{i}\,x}$ and $e^{7\,\mathrm{i}\,x}$ when there are $n = 8$ sample values, indicated by the dots on the graphs. The top row compares the real parts, $\cos x$ and $\cos 7\,x$, while the bottom row compares the imaginary parts, $\sin x$ and $-\sin 7\,x$. Note that both functions have the same pattern of sample values, even though their overall behavior is strikingly different.

This effect is commonly referred to as *aliasing*[†]. If you view a moving particle under a stroboscopic light that flashes only eight times, you would be unable to determine which of the two graphs the particle was following. Aliasing is the cause of a well-known artifact in movies: spoked wheels can appear to be rotating backwards when our brain interprets the discretization of the high frequency forward motion imposed by the frames of the film as an equivalently discretized low frequency motion in reverse. Aliasing also has important implications for the design of music CD's. We must sample an audio signal at a sufficiently high rate that all audible frequencies can be adequately represented. In fact, human appreciation of music also relies on inaudible high frequency tones, and so a much higher sample rate is actually used in commercial CD design. But the sample rate that was selected remains controversial; hi fi aficionados complain that it was not set high enough to fully reproduce the musical quality of an analog LP record!

The *discrete Fourier representation* decomposes a sampled function $f(x)$ into a linear combination of complex exponentials. Since we cannot distinguish sampled exponentials of frequency higher than n, we only need consider a finite linear combination

$$f(x) \;\sim\; p(x) \;=\; c_0 + c_1\, e^{\,\mathrm{i}\,x} + c_2\, e^{2\,\mathrm{i}\,x} + \;\cdots\; + c_{n-1}\, e^{(n-1)\,\mathrm{i}\,x} \;=\; \sum_{k=0}^{n-1} c_k\, e^{\,\mathrm{i}\,k\,x} \tag{2.4}$$

of the first n exponentials (2.3). The symbol $\sim$ in (2.4) means that the function $f(x)$ and the sum $p(x)$ agree on the sample points:

$$f(x_j) = p(x_j), \qquad\qquad j = 0, \dots, n-1. \tag{2.5}$$

Therefore, $p(x)$ can be viewed as a (complex-valued) *interpolating trigonometric polynomial* of degree $\le n-1$ for the sample data $f_j = f(x_j)$.

[†] In computer graphics, the term "aliasing" is used in a much broader sense that covers a variety of artifacts introduced by discretization — particularly, the jagged appearance of lines and smooth curves on a digital monitor.

Remark: If $f(x)$ is real, then $p(x)$ is also real on the sample points, but may very well be complex-valued in between. To avoid this unsatisfying state of affairs, we will usually discard its imaginary component, and regard the real part of $p(x)$ as "the" interpolating trigonometric polynomial. On the other hand, sticking with a purely real construction unnecessarily complicates the analysis, and so we will retain the complex exponential form (2.4) of the discrete Fourier sum.

Since we are working in the finite-dimensional vector space $\mathbb{C}^n$ throughout, we may reformulate the discrete Fourier series in vectorial form. Sampling the basic exponentials (2.3) produces the complex vectors

$$\begin{aligned} \boldsymbol{\omega}_k &= \left(e^{\mathrm{i}\, k x_0}, e^{\mathrm{i}\, k x_1}, e^{\mathrm{i}\, k x_2}, \dots, e^{\mathrm{i}\, k x_{n-1}} \right)^T \\ &= \left(1, e^{2k\pi\,\mathrm{i}/n}, e^{4k\pi\,\mathrm{i}/n}, \dots, e^{2(n-1)k\pi\,\mathrm{i}/n} \right)^T, \end{aligned} \qquad k = 0, \dots, n-1. \tag{2.6}$$

The interpolation conditions (2.5) can be recast in the equivalent vector form

$$\mathbf{f} = c_0\, \boldsymbol{\omega}_0 + c_1\, \boldsymbol{\omega}_1 + \, \cdots \, + c_{n-1}\, \boldsymbol{\omega}_{n-1}. \tag{2.7}$$

In other words, to compute the discrete Fourier coefficients $c_0, \dots, c_{n-1}$ of f, all we need to do is rewrite its sample vector $\mathbf{f}$ as a linear combination of the sampled exponential vectors $\boldsymbol{\omega}_0, \dots, \boldsymbol{\omega}_{n-1}$.

Now, the absolutely crucial property is the orthonormality of the basis elements $\boldsymbol{\omega}_0, \dots, \boldsymbol{\omega}_{n-1}$. Were it not for the power of orthogonality, Fourier analysis might have remained a mere mathematical curiosity, rather than today's indispensable tool.

Proposition 2.1. *The sampled exponential vectors $\boldsymbol{\omega}_0, \dots, \boldsymbol{\omega}_{n-1}$ form an orthonormal basis of $\mathbb{C}^n$ with respect to the inner product*

$$\langle\, \mathbf{f}\,, \mathbf{g}\,\rangle \;=\; \frac{1}{n} \sum_{j=0}^{n-1} f_j\, \overline{g_j} \;=\; \frac{1}{n} \sum_{j=0}^{n-1} f(x_j)\, \overline{g(x_j)}\,, \qquad \mathbf{f}, \mathbf{g} \in \mathbb{C}^n. \tag{2.8}$$

Remark: The inner product (2.8) is a rescaled version of the standard Hermitian dot product between complex vectors. We can interpret the inner product between the sample vectors $\mathbf{f}, \mathbf{g}$ as the *average* of the sampled values of the product signal $f(x)\, \overline{g(x)}$.

Proof: The crux of the matter relies on properties of the remarkable complex numbers

$$\zeta_n = e^{2\pi\,\mathrm{i}/n} = \cos \frac{2\pi}{n} + \mathrm{i}\, \sin \frac{2\pi}{n}\,, \qquad \text{where} \qquad n = 1, 2, 3, \dots\,. \tag{2.9}$$

Particular cases include

$$\zeta_2 = -1, \qquad \zeta_3 = -\tfrac{1}{2} + \tfrac{\sqrt{3}}{2}\, \mathrm{i}\,, \qquad \zeta_4 = \mathrm{i}\,, \qquad \text{and} \qquad \zeta_8 = \tfrac{\sqrt{2}}{2} + \tfrac{\sqrt{2}}{2}\, \mathrm{i}\,. \tag{2.10}$$

The n^{th} power of ζ_n is

$$\zeta_n^n = \left(e^{2\pi\,\mathrm{i}/n} \right)^n = e^{2\pi\,\mathrm{i}} = 1,$$

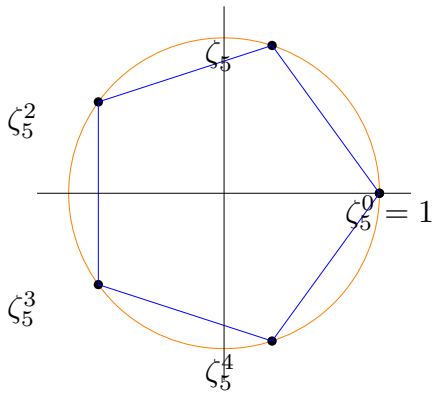

Figure 2. The Fifth Roots of Unity.

and hence ζ_n is one of the complex n^{th} *roots of unity*: $\zeta_n = \sqrt[n]{1}$. There are, in fact, n different complex n^{th} roots of 1, including 1 itself, namely the powers of ζ_n:

$$\zeta_n^k = e^{2\,k\,\pi\,\mathrm{i}\,/n} = \cos\frac{2\,k\,\pi}{n} + \mathrm{i}\,\sin\frac{2\,k\,\pi}{n}\,, \qquad\qquad k = 0, \dots, n-1. \tag{2.11}$$

Since it generates all the others, ζ_n is known as the *primitive* n^{th} *root of unity*. Geometrically, the n^{th} roots (2.11) lie on the vertices of a regular unit n–gon in the complex plane; see Figure 2. The primitive root ζ_n is the first vertex we encounter as we go around the n–gon in a counterclockwise direction, starting at 1. Continuing around, the other roots appear in their natural order $\zeta_n^2, \zeta_n^3, \dots, \zeta_n^{n-1}$, and finishing back at $\zeta_n^n = 1$. The complex conjugate of ζ_n is the "last" n^{th} root

$$e^{-2\,\pi\,\mathrm{i}\,/n} = \overline{\zeta}_n = \frac{1}{\zeta_n} = \zeta_n^{n-1} = e^{2(n-1)\pi\,\mathrm{i}\,/n}. \tag{2.12}$$

The complex numbers (2.11) are a complete set of roots of the polynomial $z^n - 1$, which can therefore be factored:

$$z^n - 1 = (z-1)(z-\zeta_n)(z-\zeta_n^2)\,\cdots\,(z-\zeta_n^{n-1}).$$

On the other hand, elementary algebra provides us with the real factorization

$$z^n - 1 = (z-1)(1 + z + z^2 + \,\cdots\, + z^{n-1}).$$

Comparing the two, we conclude that

$$1 + z + z^2 + \,\cdots\, + z^{n-1} = (z-\zeta_n)(z-\zeta_n^2)\cdots(z-\zeta_n^{n-1}).$$

Substituting $z = \zeta_n^k$ into both sides of this identity, we deduce the useful formula

$$1 + \zeta_n^k + \zeta_n^{2\,k} + \,\cdots\, + \zeta_n^{(n-1)\,k} = \begin{cases} n, & k = 0, \\ 0, & 0 < k < n. \end{cases} \tag{2.13}$$

Since $\zeta_n^{n+k} = \zeta_n^k$, this formula can easily be extended to general integers k; the sum is equal to n if n evenly divides k and is 0 otherwise.

Now, let us apply what we've learned to prove Proposition 2.1. First, in view of (2.11), the sampled exponential vectors (2.6) can all be written in terms of the n^{th} roots of unity:

$$\boldsymbol{\omega}_k = \big(\, 1, \zeta_n^k, \zeta_n^{2\,k}, \zeta_n^{3\,k}, \ \dots\ , \zeta_n^{(n-1)\,k} \,\big)^T, \qquad\qquad k = 0, \dots, n-1. \tag{2.14}$$

Therefore, applying (2.12, 13), we conclude that

$$\langle\, \boldsymbol{\omega}_k \,, \boldsymbol{\omega}_l \,\rangle = \frac{1}{n} \sum_{j=0}^{n-1} \zeta_n^{j\,k}\, \overline{\zeta_n^{j\,l}} = \frac{1}{n} \sum_{j=0}^{n-1} \zeta_n^{j(k-l)} = \begin{cases} 1, & k = l, \\ 0, & k \neq l, \end{cases} \qquad\qquad 0 \leq k, l < n,$$

which establishes orthonormality of the sampled exponential vectors. *Q.E.D.*

Orthonormality of the basis vectors implies that we can immediately compute the Fourier coefficients in the discrete Fourier sum (2.4) by taking inner products:

$$c_k = \langle\, \mathbf{f} \,, \boldsymbol{\omega}_k \,\rangle = \frac{1}{n} \sum_{j=0}^{n-1} f_j\, \overline{e^{\,\mathrm{i}\,k\,x_j}} = \frac{1}{n} \sum_{j=0}^{n-1} f_j\, e^{-\,\mathrm{i}\,k\,x_j} = \frac{1}{n} \sum_{j=0}^{n-1} \zeta_n^{-\,j\,k}\, f_j. \tag{2.15}$$

In other words, the discrete Fourier coefficient c_k is obtained by averaging the sampled values of the product function $f(x)\, e^{-\,\mathrm{i}\,k\,x}$. The passage from a signal to its Fourier coefficients is known as the *Discrete Fourier Transform* or DFT for short. The reverse procedure of reconstructing a signal from its discrete Fourier coefficients via the sum (2.4) (or (2.7)) is known as the *Inverse Discrete Fourier Transform* or IDFT.

Example 2.2. If $n = 4$, then $\zeta_4 = \mathrm{i}$. The corresponding sampled exponential vectors

$$\boldsymbol{\omega}_0 = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}, \qquad \boldsymbol{\omega}_1 = \begin{pmatrix} 1 \\ \mathrm{i} \\ -1 \\ -\,\mathrm{i} \end{pmatrix}, \qquad \boldsymbol{\omega}_2 = \begin{pmatrix} 1 \\ -1 \\ 1 \\ -1 \end{pmatrix}, \qquad \boldsymbol{\omega}_3 = \begin{pmatrix} 1 \\ -\,\mathrm{i} \\ -1 \\ \mathrm{i} \end{pmatrix},$$

form an orthonormal basis of $\mathbb{C}^4$ with respect to the averaged Hermitian dot product

$$\langle\, \mathbf{v} \,, \mathbf{w} \,\rangle = \tfrac{1}{4} \big(\, v_0\, \overline{w_0} + v_1\, \overline{w_1} + v_2\, \overline{w_2} + v_3\, \overline{w_3} \,\big), \qquad \text{where} \qquad \mathbf{v} = \begin{pmatrix} v_0 \\ v_1 \\ v_2 \\ v_3 \end{pmatrix}, \quad \mathbf{w} = \begin{pmatrix} w_0 \\ w_1 \\ w_2 \\ w_3 \end{pmatrix}.$$

Given the sampled function values

$$f_0 = f(0), \qquad f_1 = f\big(\tfrac{1}{2}\,\pi\big), \qquad f_2 = f(\pi), \qquad f_3 = f\big(\tfrac{3}{2}\,\pi\big),$$

we construct the discrete Fourier representation

$$\mathbf{f} = c_0\, \boldsymbol{\omega}_0 + c_1\, \boldsymbol{\omega}_1 + c_2\, \boldsymbol{\omega}_2 + c_3\, \boldsymbol{\omega}_3, \tag{2.16}$$

where

$$\begin{aligned} c_0 &= \langle\, \mathbf{f} \,, \boldsymbol{\omega}_0 \,\rangle = \tfrac{1}{4}(f_0 + f_1 + f_2 + f_3), & c_1 &= \langle\, \mathbf{f} \,, \boldsymbol{\omega}_1 \,\rangle = \tfrac{1}{4}(f_0 - \mathrm{i}\, f_1 - f_2 + \mathrm{i}\, f_3), \\ c_2 &= \langle\, \mathbf{f} \,, \boldsymbol{\omega}_2 \,\rangle = \tfrac{1}{4}(f_0 - f_1 + f_2 - f_3), & c_3 &= \langle\, \mathbf{f} \,, \boldsymbol{\omega}_3 \,\rangle = \tfrac{1}{4}(f_0 + \mathrm{i}\, f_1 - f_2 - \mathrm{i}\, f_3). \end{aligned}$$

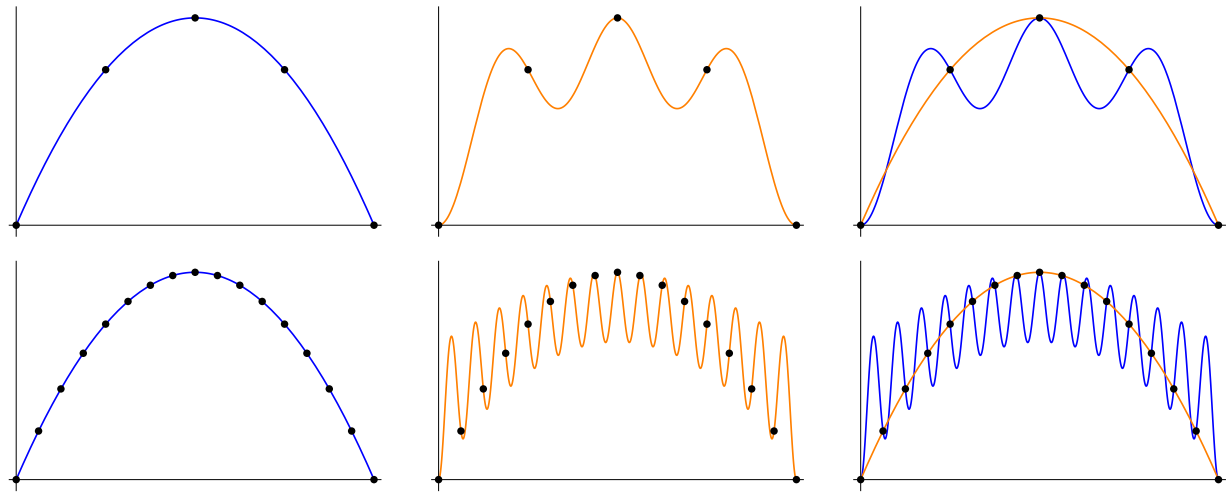

Figure 3. The Discrete Fourier Representation of $2\pi x - x^2$.

We interpret this decomposition as the complex exponential interpolant

$$f(x) \;\sim\; p(x) = c_0 + c_1\, e^{\,\mathrm{i}\,x} + c_2\, e^{2\,\mathrm{i}\,x} + c_3\, e^{3\,\mathrm{i}\,x}$$

that agrees with $f(x)$ on the sample points.

For instance, if

$$f(x) = 2\pi x - x^2,$$

then

$$f_0 = 0., \qquad f_1 = 7.4022, \qquad f_2 = 9.8696, \qquad f_3 = 7.4022,$$

and hence

$$c_0 = 6.1685, \qquad c_1 = -2.4674, \qquad c_2 = -1.2337, \qquad c_3 = -2.4674.$$

Therefore, the interpolating trigonometric polynomial is given by the real part of

$$p(x) = 6.1685 - 2.4674\, e^{\,\mathrm{i}\,x} - 1.2337\, e^{2\,\mathrm{i}\,x} - 2.4674\, e^{3\,\mathrm{i}\,x}, \tag{2.17}$$

namely,

$$\operatorname{Re} p(x) = 6.1685 - 2.4674\, \cos x - 1.2337\, \cos 2x - 2.4674\, \cos 3x. \tag{2.18}$$

In Figure 3, we compare the function, with the interpolation points indicated, and discrete Fourier representations (2.18) for both $n = 4$ and $n = 16$ points. The resulting graphs point out a significant difficulty with the Discrete Fourier Transform as developed so far. While the trigonometric polynomials do indeed correctly match the sampled function values, their pronounced oscillatory behavior makes them completely unsuitable for interpolation away from the sample points.

However, this difficulty can be rectified by being a little more clever. The problem is that we have not been paying sufficient attention to the frequencies that are represented in the Fourier sum. Indeed, the graphs in Figure 3 might remind you of our earlier observation

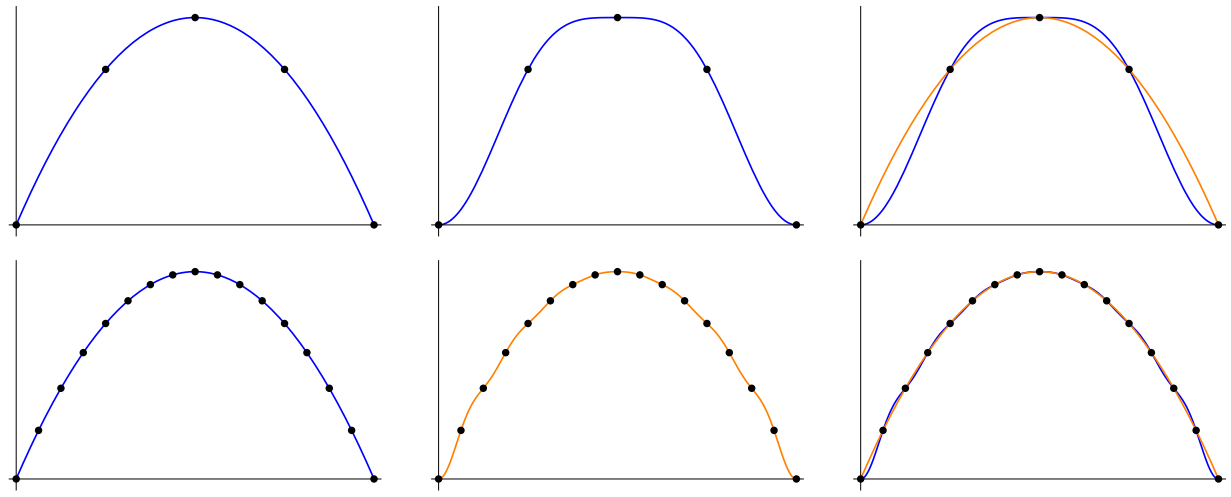

Figure 4. The Low–Frequency Discrete Fourier Representation of $x^2 - 2\pi x$.

that, due to aliasing, low and high frequency exponentials can have the same sample data, but differ wildly in between the sample points. While the first half of the summands in (2.4) represent relatively low frequencies, the second half do not, and can be replaced by equivalent lower frequency, and hence less oscillatory exponentials. Namely, if $0 < k \leq \frac{1}{2}n$, then $e^{-\mathrm{i}kx}$ and $e^{\mathrm{i}(n-k)x}$ have the same sample values, but the former is of lower frequency than the latter. Thus, for interpolatory purposes, we should replace the second half of the summands in the Fourier sum (2.4) by their low frequency alternatives. If $n = 2m+1$ is odd, then we take

$$\widehat{p}(x) \;=\; c_{-m}\, e^{-\mathrm{i}mx} + \cdots + c_{-1}\, e^{-\mathrm{i}x} + c_0 + c_1\, e^{\mathrm{i}x} + \cdots + c_m\, e^{\mathrm{i}mx} \;=\; \sum_{k=-m}^{m} c_k\, e^{\mathrm{i}kx} \tag{2.19}$$

as the equivalent low frequency interpolant. If $n = 2m$ is even — which is the most common case occurring in applications — then

$$\widehat{p}(x) \;=\; c_{-m}\, e^{-\mathrm{i}mx} + \cdots + c_{-1}\, e^{-\mathrm{i}x} + c_0 + c_1\, e^{\mathrm{i}x} + \cdots + c_{m-1}\, e^{\mathrm{i}(m-1)x} \;=\; \sum_{k=-m}^{m-1} c_k\, e^{\mathrm{i}kx} \tag{2.20}$$

will be our choice. (It is a matter of personal taste whether to use $e^{-\mathrm{i}mx}$ or $e^{\mathrm{i}mx}$ to represent the highest frequency term.) In both cases, the Fourier coefficients with negative indices are the same as their high frequency alternatives:

$$c_{-k} = c_{n-k} = \langle\, \mathbf{f}\,, \boldsymbol{\omega}_{n-k}\,\rangle = \langle\, \mathbf{f}\,, \boldsymbol{\omega}_{-k}\,\rangle, \tag{2.21}$$

where $\boldsymbol{\omega}_{-k} = \boldsymbol{\omega}_{n-k}$ is the sample vector for $e^{-\mathrm{i}kx} \sim e^{\mathrm{i}(n-k)x}$.

Returning to the previous example, for interpolating purposes, we should replace (2.17) by the equivalent low frequency interpolant

$$\widehat{p}(x) = -1.2337\, e^{-2\mathrm{i}x} - 2.4674\, e^{-\mathrm{i}x} + 6.1685 - 2.4674\, e^{\mathrm{i}x}, \tag{2.22}$$

with real part

$$\operatorname{Re} \widehat{p}\,(x) = 6.1685 - 4.9348 \cos x - 1.2337 \cos 2x.$$

Graphs of the $n = 4$ and 16 low frequency trigonometric interpolants can be seen in Figure 4. Thus, by utilizing only the lowest frequency exponentials, we successfully suppress the aliasing artifacts, resulting in a quite reasonable trigonometric interpolant to the given function.

Remark: The low frequency version also serves to unravel the reality of the Fourier representation of a real function $f(x)$. Since $\boldsymbol{\omega}_{-k} = \overline{\boldsymbol{\omega}_k}$, formula (2.21) implies that $c_{-k} = \overline{c_k}$, and so the common frequency terms

$$c_{-k}\, e^{-\mathrm{i}\, k x} + c_k\, e^{\mathrm{i}\, k x} = a_k \cos k x + b_k \sin k x$$

add up to a real trigonometric function. Therefore, the odd n interpolant (2.19) is a real trigonometric polynomial, whereas in the even version (2.20) only the highest frequency term $c_{-m}\, e^{-\mathrm{i}\, m x}$ produces a complex term — which is, in fact, 0 on the sample points.

Compression and Noise Removal

In a typical experimental signal, noise primarily affects the high frequency modes, while the authentic features tend to appear in the low frequencies. Think of the hiss and static you hear on an AM radio station or a low quality audio recording. Thus, a very simple, but effective, method for denoising a corrupted signal is to decompose it into its Fourier modes, as in (2.4), and then discard the high frequency constituents. A similar idea underlies the Dolby® recording system used on most movie soundtracks: during the recording process, the high frequency modes are artificially boosted, so that scaling them back when showing the movie in the theater has the effect of eliminating much of the extraneous noise. The one design issue is the specification of a cut-off between low and high frequency, that is, between signal and noise. This choice will depend upon the properties of the measured signal, and is left to the discretion of the signal processor.

A correct implementation of the denoising procedure is facilitated by using the unaliased forms (2.19, 20) of the trigonometric interpolant, in which the low frequency summands only appear when $|\, k\,|$ is small. In this version, to eliminate high frequency components, we replace the full summation by

$$q_l(x) \;=\; \sum_{k=-l}^{l} c_k\, e^{\mathrm{i}\, k x}, \tag{2.23}$$

where $l < \frac{1}{2}\,(n+1)$ specifies the selected cut-off frequency between signal and noise. The $2l + 1 \ll n$ low frequency Fourier modes retained in (2.23) will, in favorable situations, capture the essential features of the original signal while simultaneously eliminating the high frequency noise.

In Figure 5 we display a sample signal followed by the same signal corrupted by adding in random noise. We use $n = 2^9 = 512$ sample points in the discrete Fourier representation, and to remove the noise, we retain only the $2l + 1 = 11$ lowest frequency modes. In other words, instead of all $n = 512$ Fourier coefficients $c_{-256}, \dots, c_{-1}, c_0, c_1, \dots, c_{255}$, we only compute the 11 lowest order ones $c_{-5}, \dots, c_5$. Summing up just those 11 exponentials

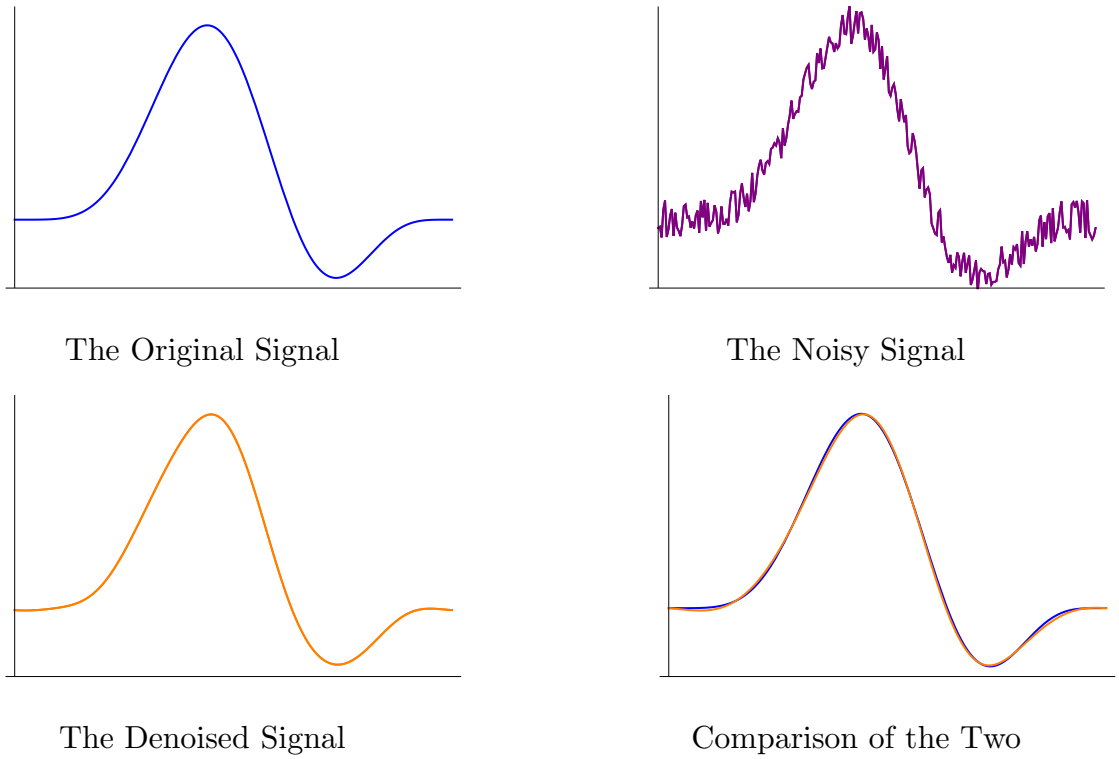

Figure 5. Denoising a Signal.

produces the denoised signal $q(x) = c_{-5}\, e^{-5\,\mathrm{i}\,x} + \cdots + c_5\, e^{5\,\mathrm{i}\,x}$. To compare, we plot both the original signal and the denoised version on the same graph. In this case, the maximal deviation is less than .15 over the entire interval $[\,0, 2\,\pi\,]$.

The same idea underlies many data compression algorithms for audio recordings, digital images and, particularly, video. The goal is efficient storage and/or transmission of the signal. As before, we expect all the important features to be contained in the low frequency constituents, and so discarding the high frequency terms will, in favorable situations, not lead to any noticeable degradation of the signal or image. Thus, to compress a signal (and, simultaneously, remove high frequency noise), we retain only its low frequency discrete Fourier coefficients. The signal is reconstructed by summing the associated truncated discrete Fourier series (2.23). A mathematical justification of Fourier-based compression algorithms relies on the fact that the Fourier coefficients of smooth functions tend rapidly to zero — the smoother the function, the faster the decay rate. Thus, the small high frequency Fourier coefficients will be of negligible importance.

In Figure 6, the same signal is compressed by retaining, respectively, $2l + 1 = 21$ and $2l + 1 = 7$ Fourier coefficients only instead of all $n = 512$ that would be required for complete accuracy. For the case of moderate compression, the maximal deviation between the signal and the compressed version is less than 1.5×10^{-4} over the entire interval, while even the highly compressed version deviates at most .05 from the original signal. Of course, the lack of any fine scale features in this particular signal means that a very high compression can be achieved — the more complicated or detailed the original signal, the

The Original Signal Moderate Compression High Compression

Figure 6. Compressing a Signal.

more Fourier modes need to be retained for accurate reproduction.

The Fast Fourier Transform

While one may admire an algorithm for its intrinsic beauty, in the real world, the bottom line is always efficiency of implementation: the less total computation, the faster the processing, and hence the more extensive the range of applications. Orthogonality is the first and most important feature of many practical linear algebra algorithms, and is *the* critical feature of Fourier analysis. Still, even the power of orthogonality reaches its limits when it comes to dealing with truly large scale problems such as three-dimensional medical imaging or video processing. In the early 1960's, James Cooley and John Tukey, [**4**], discovered† a much more efficient approach to the Discrete Fourier Transform, exploiting the rather special structure of the sampled exponential vectors. The resulting algorithm is known as the *Fast Fourier Transform*, often abbreviated FFT, and its discovery launched the modern revolution in digital signal and data processing, [**2**, **3**].

In general, computing all the discrete Fourier coefficients (2.15) of an n times sampled signal requires a total of n^2 complex multiplications and $n^2 - n$ complex additions. Note also that each complex addition

$$z + w = (x + \mathrm{i}\, y) + (u + \mathrm{i}\, v) = (x + u) + \mathrm{i}\,(y + v) \tag{2.24}$$

generally requires two real additions, while each complex multiplication

$$z\, w = (x + \mathrm{i}\, y)\,(u + \mathrm{i}\, v) = (x\, u - y\, v) + \mathrm{i}\,(x\, v + y\, u) \tag{2.25}$$

requires 4 real multiplications and 2 real additions, or, by employing the alternative formula

$$x\, v + y\, u = (x + y)\,(u + v) - x\, u - y\, v \tag{2.26}$$

for the imaginary part, 3 real multiplications and 5 real additions. (The choice of formula (2.25) or (2.26) will depend upon the processor's relative speeds of multiplication and addition.) Similarly, given the Fourier coefficients $c_0, \dots, c_{n-1}$, reconstruction of the sampled signal via (2.4) requires $n^2 - n$ complex multiplications and $n^2 - n$ complex additions. As a

† In fact, the key ideas can be found in Gauss' hand computations in the early 1800's, but his insight was not fully appreciated until modern computers arrived on the scene.

result, both computations become quite labor intensive for large n. Extending these ideas to multi-dimensional data only exacerbates the problem.

In order to explain the method without undue complication, we return to the original, aliased form of the discrete Fourier representation (2.4). (Once one understands how the FFT works, one can easily adapt the algorithm to the low frequency version (2.20).) The seminal observation is that if the number of sample points

$$n = 2m$$

is even, then the primitive m^{th} root of unity $\zeta_m = \sqrt[m]{1}$ equals the square of the primitive n^{th} root:

$$\zeta_m = \zeta_n^2.$$

We use this fact to split the summation (2.15) for the order n discrete Fourier coefficients into two parts, collecting together the even and the odd powers of ζ_n^k:

$$\begin{aligned} c_k &= \frac{1}{n}\left(f_0 + f_1\,\zeta_n^{-k} + f_2\,\zeta_n^{-2\,k} + \,\cdots\, + f_{n-1}\,\zeta_n^{-(n-1)\,k}\right) \\ &= \frac{1}{n}\left(f_0 + f_2\,\zeta_n^{-2\,k} + f_4\,\zeta_n^{-4\,k} + \,\cdots\, + f_{2m-2}\,\zeta_n^{-(2\,m-2)\,k}\right) + \\ &\qquad + \zeta_n^{-k}\,\frac{1}{n}\left(f_1 + f_3\,\zeta_n^{-2\,k} + f_5\,\zeta_n^{-4\,k} + \,\cdots\, + f_{2m-1}\,\zeta_n^{-(2\,m-2)\,k}\right) \\ &= \frac{1}{2}\left\{ \frac{1}{m}\left(f_0 + f_2\,\zeta_m^{-k} + f_4\,\zeta_m^{-2\,k} + \,\cdots\, + f_{2m-2}\,\zeta_m^{-(m-1)\,k}\right)\right\} + \\ &\qquad + \frac{\zeta_n^{-k}}{2}\left\{ \frac{1}{m}\left(f_1 + f_3\,\zeta_m^{-k} + f_5\,\zeta_m^{-2\,k} + \,\cdots\, + f_{2m-1}\,\zeta_m^{-(m-1)\,k}\right)\right\}. \end{aligned} \tag{2.27}$$

Now, observe that the expressions in braces are the order m Fourier coefficients for the sample data

$$\begin{aligned} \mathbf{f}^e &= \left(f_0, f_2, f_4, \ldots, f_{2m-2}\right)^T = \left(f(x_0), f(x_2), f(x_4), \ldots, f(x_{2m-2})\right)^T, \\ \mathbf{f}^o &= \left(f_1, f_3, f_5, \ldots, f_{2m-1}\right)^T = \left(f(x_1), f(x_3), f(x_5), \ldots, f(x_{2m-1})\right)^T. \end{aligned} \tag{2.28}$$

Note that $\mathbf{f}^e$ is obtained by sampling $f(x)$ on the *even* sample points x_{2j}, while $\mathbf{f}^o$ is obtained by sampling the same function $f(x)$, but now at the *odd* sample points x_{2j+1}. In other words, we are splitting the original sampled signal into two "half-sampled" signals obtained by sampling on every other point. The even and odd Fourier coefficients are

$$\begin{aligned} c_k^e &= \frac{1}{m}\left(f_0 + f_2\,\zeta_m^{-k} + f_4\,\zeta_m^{-2\,k} + \,\cdots\, + f_{2m-2}\,\zeta_m^{-(m-1)\,k}\right), \\ c_k^o &= \frac{1}{m}\left(f_1 + f_3\,\zeta_m^{-k} + f_5\,\zeta_m^{-2\,k} + \,\cdots\, + f_{2m-1}\,\zeta_m^{-(m-1)\,k}\right), \end{aligned} \qquad k = 0, \ldots, m-1. \tag{2.29}$$

Since they contain just m data values, both the even and odd samples require only m distinct Fourier coefficients, and we adopt the identification

$$c_{k+m}^e = c_k^e, \qquad c_{k+m}^o = c_k^o, \qquad k = 0, \ldots, m-1. \tag{2.30}$$

Therefore, the order $n = 2m$ discrete Fourier coefficients (2.27) can be constructed from a pair of order m discrete Fourier coefficients via

$$c_k = \tfrac{1}{2}\big(c^e_k + \zeta_n^{-k}\, c^o_k \big), \qquad k = 0, \dots, n-1. \tag{2.31}$$

Now if $m = 2l$ is also even, then we can play the same game on the order m Fourier coefficients (2.29), reconstructing each of them from a pair of order l discrete Fourier coefficients — obtained by sampling the signal at every fourth point. If $n = 2^r$ is a power of 2, then this game can be played all the way back to the start, beginning with the trivial order 1 discrete Fourier representation, which just samples the function at a single point. The result is the desired algorithm. After some rearrangement of the basic steps, we arrive at the Fast Fourier Transform, which we now present in its final form.

We begin with a sampled signal on $n = 2^r$ sample points. To efficiently program the Fast Fourier Transform, it helps to write out each index $0 \le j < 2^r$ in its binary (as opposed to decimal) representation

$$j = j_{r-1}\, j_{r-2}\, \cdots\, j_2\, j_1\, j_0, \qquad \text{where} \qquad j_\nu = 0 \text{ or } 1; \tag{2.32}$$

the notation is shorthand for its r digit binary expansion

$$j = j_0 + 2\, j_1 + 4\, j_2 + 8\, j_3 + \ \cdots\ + 2^{r-1}\, j_{r-1}.$$

We then define the *bit reversal* map

$$\rho(j_{r-1}\, j_{r-2}\, \cdots\, j_2\, j_1\, j_0) = j_0\, j_1\, j_2\, \cdots\, j_{r-2}\, j_{r-1}. \tag{2.33}$$

For instance, if $r = 5$, and $j = 13$, with 5 digit binary representation 01101, then $\rho(j) = 22$ has the reversed binary representation 10110. Note especially that the bit reversal map $\rho = \rho_r$ depends upon the original choice of $r = \log_2 n$.

Secondly, for each $0 \le k < r$, define the maps

$$\begin{aligned} \alpha_k(j) &= j_{r-1}\, \cdots\, j_{k+1}\, 0\, j_{k-1}\, \cdots\, j_0, \\ \beta_k(j) &= j_{r-1}\, \cdots\, j_{k+1}\, 1\, j_{k-1}\, \cdots\, j_0 = \alpha_k(j) + 2^k, \end{aligned} \qquad \text{for} \qquad j = j_{r-1}\, j_{r-2}\, \cdots\, j_1\, j_0. \tag{2.34}$$

In other words, $\alpha_k(j)$ sets the k^{th} binary digit of j to 0, while $\beta_k(j)$ sets it to 1. In the preceding example, $\alpha_2(13) = 9$, with binary form 01001, while $\beta_2(13) = 13$ with binary form 01101. The bit operations (2.33, 34) are especially easy to implement on modern binary computers.

Given a sampled signal $f_0, \dots, f_{n-1}$, its discrete Fourier coefficients $c_0, \dots, c_{n-1}$ are computed by the following iterative algorithm:

$$c^{(0)}_j = f_{\rho(j)}, \qquad c^{(k+1)}_j = \tfrac{1}{2}\big(c^{(k)}_{\alpha_k(j)} + \zeta_{2^{k+1}}^{-j}\, c^{(k)}_{\beta_k(j)} \big), \qquad \begin{matrix} j = 0, \dots, n-1, \\ k = 0, \dots, r-1, \end{matrix} \tag{2.35}$$

in which $\zeta_{2^{k+1}}$ is the primitive 2^{k+1} root of unity. The final output of the iterative procedure, namely

$$c_j = c^{(r)}_j, \qquad j = 0, \dots, n-1, \tag{2.36}$$

are the discrete Fourier coefficients of our signal. The preprocessing step of the algorithm, where we define $c_j^{(0)}$, produces a more convenient rearrangement of the sample values. The subsequent steps successively combine the Fourier coefficients of the appropriate even and odd sampled subsignals together, reproducing (2.27) in a different notation. The following example should help make the overall process clearer.

Example 2.3. Consider the case $r = 3$, and so our signal has $n = 2^3 = 8$ sampled values $f_0, f_1, \dots, f_7$. We begin the process by rearranging the sample values

$$c_0^{(0)} = f_0,\ \ c_1^{(0)} = f_4,\ \ c_2^{(0)} = f_2,\ \ c_3^{(0)} = f_6,\ \ c_4^{(0)} = f_1,\ \ c_5^{(0)} = f_5,\ \ c_6^{(0)} = f_3,\ \ c_7^{(0)} = f_7,$$

in the order specified by the bit reversal map ρ. For instance $\rho(3) = 6$, or, in binary notation, $\rho(011) = 110$.

The first stage of the iteration is based on $\zeta_2 = -1$. Equation (2.35) gives

$$\begin{aligned}
&c_0^{(1)} = \tfrac{1}{2}(c_0^{(0)} + c_1^{(0)}), \quad c_1^{(1)} = \tfrac{1}{2}(c_0^{(0)} - c_1^{(0)}), \quad c_2^{(1)} = \tfrac{1}{2}(c_2^{(0)} + c_3^{(0)}), \quad c_3^{(1)} = \tfrac{1}{2}(c_2^{(0)} - c_3^{(0)}),\\
&c_4^{(1)} = \tfrac{1}{2}(c_4^{(0)} + c_5^{(0)}), \quad c_5^{(1)} = \tfrac{1}{2}(c_4^{(0)} - c_5^{(0)}), \quad c_6^{(1)} = \tfrac{1}{2}(c_6^{(0)} + c_7^{(0)}), \quad c_7^{(1)} = \tfrac{1}{2}(c_6^{(0)} - c_7^{(0)}),
\end{aligned}$$

where we combine successive pairs of the rearranged sample values. The second stage of the iteration has $k = 1$ with $\zeta_4 = \mathrm{i}$. We find

$$\begin{aligned}
&c_0^{(2)} = \tfrac{1}{2}(c_0^{(1)} + c_2^{(1)}), \quad c_1^{(2)} = \tfrac{1}{2}(c_1^{(1)} - \mathrm{i}\, c_3^{(1)}), \quad c_2^{(2)} = \tfrac{1}{2}(c_0^{(1)} - c_2^{(1)}), \quad c_3^{(2)} = \tfrac{1}{2}(c_1^{(1)} + \mathrm{i}\, c_3^{(1)}),\\
&c_4^{(2)} = \tfrac{1}{2}(c_4^{(1)} + c_6^{(1)}), \quad c_5^{(2)} = \tfrac{1}{2}(c_5^{(1)} - \mathrm{i}\, c_7^{(1)}), \quad c_6^{(2)} = \tfrac{1}{2}(c_4^{(1)} - c_6^{(1)}), \quad c_7^{(2)} = \tfrac{1}{2}(c_5^{(1)} + \mathrm{i}\, c_7^{(1)}).
\end{aligned}$$

Note that the indices of the combined pairs of coefficients differ by 2. In the last step, where $k = 2$ and $\zeta_8 = \frac{\sqrt{2}}{2}(1 + \mathrm{i}\,)$, we combine coefficients whose indices differ by $4 = 2^2$; the final output

$$\begin{aligned}
c_0 &= c_0^{(3)} = \tfrac{1}{2}(c_0^{(2)} + c_4^{(2)}), & c_4 &= c_4^{(3)} = \tfrac{1}{2}(c_0^{(2)} - c_4^{(2)}),\\
c_1 &= c_1^{(3)} = \tfrac{1}{2}\big(c_1^{(2)} + \tfrac{\sqrt{2}}{2}(1 - \mathrm{i}\,)\, c_5^{(2)}\big), & c_5 &= c_5^{(3)} = \tfrac{1}{2}\big(c_1^{(2)} - \tfrac{\sqrt{2}}{2}(1 - \mathrm{i}\,)\, c_5^{(2)}\big),\\
c_2 &= c_2^{(3)} = \tfrac{1}{2}\big(c_2^{(2)} - \mathrm{i}\, c_6^{(2)}\big), & c_6 &= c_6^{(3)} = \tfrac{1}{2}\big(c_2^{(2)} + \mathrm{i}\, c_6^{(2)}\big),\\
c_3 &= c_3^{(3)} = \tfrac{1}{2}\big(c_3^{(2)} - \tfrac{\sqrt{2}}{2}(1 + \mathrm{i}\,)\, c_7^{(2)}\big), & c_7 &= c_7^{(3)} = \tfrac{1}{2}\big(c_3^{(2)} + \tfrac{\sqrt{2}}{2}(1 + \mathrm{i}\,)\, c_7^{(2)}\big),
\end{aligned}$$

is the complete set of discrete Fourier coefficients.

Let us count the number of arithmetic operations required in the Fast Fourier Transform algorithm. At each stage in the computation, we must perform $n = 2^r$ complex additions/subtractions and the same number of complex multiplications. (Actually, the number of multiplications is slightly smaller since multiplications by ± 1 and $\pm\,\mathrm{i}$ are extremely simple. However, this does not significantly alter the final operations count.) There are $r = \log_2 n$ stages, and so we require a total of $r\, n = n \log_2 n$ complex additions/subtractions and the same number of multiplications. Now, when n is large, $n \log_2 n$ is *significantly* smaller than n^2, which is the number of operations required for the direct algorithm. For instance, if $n = 2^{10} = 1,024$, then $n^2 = 1,048,576$, while $n \log_2 n = 10,240$ — a net savings of 99%. As a result, many large scale computations that would be intractable using the direct approach are immediately brought into the realm of feasibility.

This is the reason why all modern implementations of the Discrete Fourier Transform are based on the FFT algorithm and its variants.

The reconstruction of the signal from the discrete Fourier coefficients $c_0, \dots, c_{n-1}$ is speeded up in exactly the same manner. The only differences are that we replace $\zeta_n^{-1} = \overline{\zeta_n}$ by ζ_n, and drop the factors of $\frac{1}{2}$ since there is no need to divide by n in the final result (2.4). Therefore, we apply the slightly modified iterative procedure

$$f_j^{(0)} = c_{\rho(j)}, \qquad f_j^{(k+1)} = f_{\alpha_k(j)}^{(k)} + \zeta_{2^{k+1}}^{j}\, f_{\beta_k(j)}^{(k)}, \qquad \begin{array}{l} j = 0, \dots, n-1, \\ k = 0, \dots, r-1, \end{array} \tag{2.37}$$

and finish with

$$f(x_j) = f_j = f_j^{(r)}, \qquad j = 0, \dots, n-1. \tag{2.38}$$

Example 2.4. The reconstruction formulae in the case of $n = 8 = 2^3$ Fourier coefficients $c_0, \dots, c_7$, which were computed in Example 2.3, can be implemented as follows. First, we rearrange the Fourier coefficients in bit reversed order:

$$f_0^{(0)} = c_0, \;\; f_1^{(0)} = c_4, \;\; f_2^{(0)} = c_2, \;\; f_3^{(0)} = c_6, \;\; f_4^{(0)} = c_1, \;\; f_5^{(0)} = c_5, \;\; f_6^{(0)} = c_3, \;\; f_7^{(0)} = c_7,$$

Then we begin combining them in successive pairs:

$$\begin{array}{llll} f_0^{(1)} = f_0^{(0)} + f_1^{(0)}, & f_1^{(1)} = f_0^{(0)} - f_1^{(0)}, & f_2^{(1)} = f_2^{(0)} + f_3^{(0)}, & f_3^{(1)} = f_2^{(0)} - f_3^{(0)}, \\ f_4^{(1)} = f_4^{(0)} + f_5^{(0)}, & f_5^{(1)} = f_4^{(0)} - f_5^{(0)}, & f_6^{(1)} = f_6^{(0)} + f_7^{(0)}, & f_7^{(1)} = f_6^{(0)} - f_7^{(0)}. \end{array}$$

Next,

$$\begin{array}{llll} f_0^{(2)} = f_0^{(1)} + f_2^{(1)}, & f_1^{(2)} = f_1^{(1)} + \mathrm{i}\, f_3^{(1)}, & f_2^{(2)} = f_0^{(1)} - f_2^{(1)}, & f_3^{(2)} = f_1^{(1)} - \mathrm{i}\, f_3^{(1)}, \\ f_4^{(2)} = f_4^{(1)} + f_6^{(1)}, & f_5^{(2)} = f_5^{(1)} + \mathrm{i}\, f_7^{(1)}, & f_6^{(2)} = f_4^{(1)} - f_6^{(1)}, & f_7^{(2)} = f_5^{(1)} - \mathrm{i}\, f_7^{(1)}. \end{array}$$

Finally, the sampled signal values are

$$\begin{array}{ll} f(x_0) = f_0^{(3)} = f_0^{(2)} + f_4^{(2)}, & f(x_4) = f_4^{(3)} = f_0^{(2)} - f_4^{(2)}, \\ f(x_1) = f_1^{(3)} = f_1^{(2)} + \frac{\sqrt{2}}{2}\,(1 + \mathrm{i}\,)\, f_5^{(2)}, & f(x_5) = f_5^{(3)} = f_1^{(2)} - \frac{\sqrt{2}}{2}\,(1 + \mathrm{i}\,)\, f_5^{(2)}, \\ f(x_2) = f_2^{(3)} = f_2^{(2)} + \mathrm{i}\, f_6^{(2)}, & f(x_6) = f_6^{(3)} = f_2^{(2)} - \mathrm{i}\, f_6^{(2)}, \\ f(x_3) = f_3^{(3)} = f_3^{(2)} - \frac{\sqrt{2}}{2}\,(1 - \mathrm{i}\,)\, f_7^{(2)}, & f(x_7) = f_7^{(3)} = f_3^{(2)} + \frac{\sqrt{2}}{2}\,(1 - \mathrm{i}\,)\, f_7^{(2)}. \end{array}$$

3. Wavelets.

Trigonometric Fourier series, both continuous and discrete, are amazingly powerful, but they do suffer from one potentially serious defect. The basis functions $e^{\mathrm{i}\,k\,x} = \cos k\,x + \mathrm{i}\,\sin k\,x$ are spread out over the entire interval $[\,-\pi, \pi\,]$, and so are not well-suited to processing localized signals — meaning data that are concentrated in a relatively small regions. Indeed, the most concentrated data of all — a single delta function — has every Fourier component of equal magnitude in its Fourier series and its high degree of localization is completely obscured. Ideally, one would like to construct a system of functions that is orthogonal, and so has all the advantages of the Fourier trigonometric functions, but, in

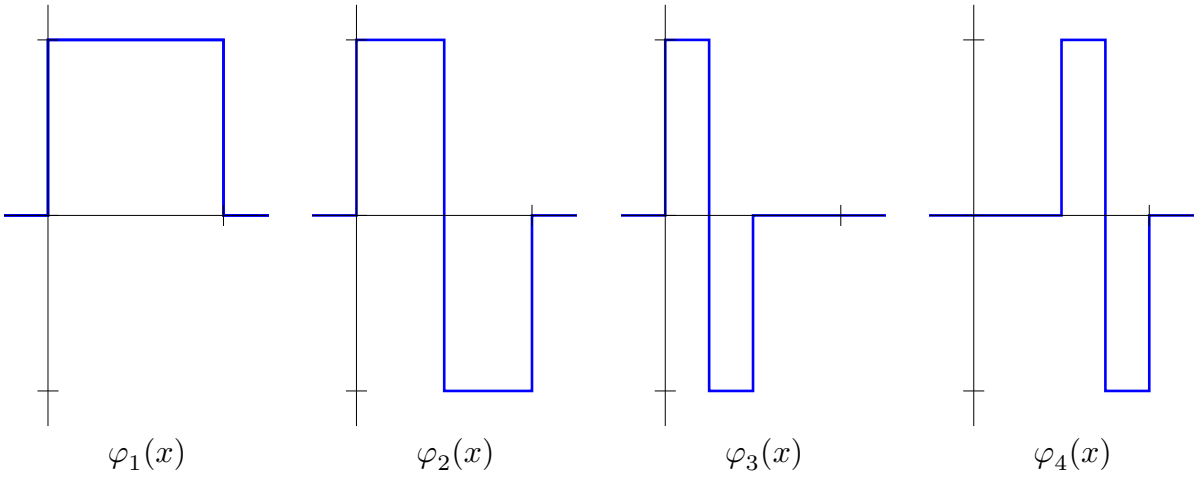

Figure 7. The First Four Haar Wavelets.

addition, adapts to localized structures in signals. This dream was the inspiration for the development of the modern theory of wavelets.

The Haar Wavelets

Although the modern era of wavelets started in the mid 1980's, the simplest example of a wavelet basis was discovered by the Hungarian mathematician Alfréd Haar in 1910, [**6**]. We consider the space of functions (signals) defined the interval $[\,0,1\,]$, equipped with the standard L^2 inner product

$$\langle\, f\,,g\,\rangle = \int_0^1 f(x)\,g(x)\,dx. \tag{3.1}$$

This choice is merely for convenience, being slightly better suited to our construction than $[\,-\pi,\pi\,]$ or $[\,0,2\,\pi\,]$. Moreover, the usual scaling arguments can be used to adapt the wavelet formulas to any other interval.

The *Haar wavelets* are certain piecewise constant functions. The initial four are graphed in The first is the *box function*

$$\varphi_1(x) = \varphi(x) = \begin{cases} 1, & 0 < x \le 1, \\ 0, & \text{otherwise}, \end{cases} \tag{3.2}$$

known as the *scaling function*, for reasons that shall appear shortly. Although we are only interested in the value of $\varphi(x)$ on the interval $[\,0,1\,]$, it will be convenient to extend it, and all the other wavelets, to be zero outside the basic interval. Its values at the points of discontinuity, i.e., $0,1$, is not critical, but, unlike the Fourier series midpoint value, it will be more convenient to consistently choose the left hand limiting value. The second Haar function

$$\varphi_2(x) = w(x) = \begin{cases} 1, & 0 < x \le \frac{1}{2}, \\ -1, & \frac{1}{2} < x \le 1, \\ 0, & \text{otherwise}, \end{cases} \tag{3.3}$$

is known as the *mother wavelet*. The third and fourth Haar functions are compressed versions of the mother wavelet:

$$\varphi_3(x) = w(2x) = \begin{cases} 1, & 0 < x \le \frac{1}{4}, \\ -1, & \frac{1}{4} < x \le \frac{1}{2}, \\ 0, & \text{otherwise}, \end{cases} \qquad \varphi_4(x) = w(2x-1) = \begin{cases} 1, & \frac{1}{2} < x \le \frac{3}{4}, \\ -1, & \frac{3}{4} < x \le 1, \\ 0, & \text{otherwise}, \end{cases}$$

called *daughter wavelets*. One can easily check, by direct evaluation of the integrals, that the four Haar wavelet functions are orthogonal with respect to the L^2 inner product (3.1): $\langle\, \varphi_i \,,\, \varphi_j \,\rangle = 0$ when $i \neq j$.

The scaling transformation $x \mapsto 2x$ serves to compress the wavelet function, while the translation $2x \mapsto 2x - 1$ moves the compressed version to the right by a half a unit. Furthermore, we can represent the mother wavelet by compressing and translating the scaling function:

$$w(x) = \varphi(2x) - \varphi(2x-1). \tag{3.4}$$

It is these two operations of scaling and compression — coupled with the all-important orthogonality — that underlies the power of wavelets.

The Haar wavelets have an evident discretization. If we decompose the interval $(0, 1]$ into the four subintervals

$$\left(0\,,\tfrac{1}{4}\right], \qquad \left(\tfrac{1}{4}\,,\tfrac{1}{2}\right], \qquad \left(\tfrac{1}{2}\,,\tfrac{3}{4}\right], \qquad \left(\tfrac{3}{4}\,,1\right], \tag{3.5}$$

on which the four wavelet functions are constant, then we can represent each of them by a vector in $\mathbb{R}^4$ whose entries are the values of each wavelet function sampled at the left endpoint of each subinterval. In this manner, we obtain the wavelet sample vectors

$$\mathbf{v}_1 = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}, \qquad \mathbf{v}_2 = \begin{pmatrix} 1 \\ 1 \\ -1 \\ -1 \end{pmatrix}, \qquad \mathbf{v}_3 = \begin{pmatrix} 1 \\ -1 \\ 0 \\ 0 \end{pmatrix}, \qquad \mathbf{v}_4 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ -1 \end{pmatrix}. \tag{3.6}$$

Orthogonality of the vectors (3.6) with respect to the standard Euclidean dot product is equivalent to orthogonality of the Haar wavelet functions with respect to the inner product (3.1). Indeed, if

$$f(x) \;\sim\; \mathbf{f} = (f_1, f_2, f_3, f_4) \qquad \text{and} \qquad g(x) \;\sim\; \mathbf{g} = (g_1, g_2, g_3, g_4)$$

are *piecewise constant* real functions that achieve the indicated values on the four subintervals (3.5), then their L^2 inner product

$$\langle\, f \,,\, g \,\rangle = \int_0^1 f(x)\, g(x)\, dx = \tfrac{1}{4}\left(\, f_1\, g_1 + f_2\, g_2 + f_3\, g_3 + f_4\, g_4 \,\right) = \tfrac{1}{4}\, \mathbf{f} \cdot \mathbf{g},$$

is equal to the averaged dot product of their sample values — the real form of the inner product (2.8) that was used in the discrete Fourier transform.

Since the vectors (3.6) form an orthogonal basis of $\mathbb{R}^4$, we can uniquely decompose any such piecewise constant function as a linear combination of wavelets

$$f(x) = c_1\, \varphi_1(x) + c_2\, \varphi_2(x) + c_3\, \varphi_3(x) + c_4\, \varphi_4(x),$$

or, equivalently, in terms of the sample vectors,

$$\mathbf{f} = c_1 \mathbf{v}_1 + c_2 \mathbf{v}_2 + c_3 \mathbf{v}_3 + c_4 \mathbf{v}_4.$$

The required coefficients

$$c_k = \frac{\langle\, f \,,\, \varphi_k \,\rangle}{\| \,\varphi_k\,\|^2} = \frac{\mathbf{f} \cdot \mathbf{v}_k}{\| \,\mathbf{v}_k\,\|^2}$$

are found using orthogonality. of the sample vectors Explicitly,

$$\begin{aligned} c_1 &= \tfrac{1}{4}\,(f_1 + f_2 + f_3 + f_4), & \qquad c_3 &= \tfrac{1}{2}\,(f_1 - f_2),\\ c_2 &= \tfrac{1}{4}\,(f_1 + f_2 - f_3 - f_4), & \qquad c_4 &= \tfrac{1}{2}\,(f_3 - f_4). \end{aligned}$$

Before proceeding to the more general case, let us introduce an important analytical definition that quantifies precisely how localized a function is.

Definition 3.1. The *support* of a function $f(x)$, written $\operatorname{supp} f$, is the closure of the set where $f(x) \neq 0$.

Thus, a point will belong to the support of $f(x)$, provided f is not zero there, or at least is not zero at nearby points. More precisely:

Lemma 3.2. *If $f(a) \neq 0$, then $a \in \operatorname{supp} f$. More generally, a point $a \in \operatorname{supp} f$ if and only if there exist a convergent sequence $x_n \to a$ such that $f(x_n) \neq 0$. Conversely, $a \notin \operatorname{supp} f$ if and only if $f(x) \equiv 0$ on an interval $a - \delta < x < a + \delta$ for some $\delta > 0$.*

Intuitively, the smaller the support of a function, the more localized it is. For example, the support of the Haar mother wavelet (3.3) is $\operatorname{supp} w = [0, 1]$ — the point $x = 0$ is included, even though $w(0) = 0$, because $w(x) \neq 0$ at nearby points. The two daughter wavelets have smaller support:

$$\operatorname{supp} \varphi_3 = \left[\, 0 \,,\, \tfrac{1}{2} \,\right], \qquad\qquad \operatorname{supp} \varphi_4 = \left[\, \tfrac{1}{2} \,,\, 1 \,\right],$$

and so are twice as localized. An extreme case is the delta function, whose support is a single point. In contrast, the support of the Fourier trigonometric basis functions is all of $\mathbb{R}$, since they only vanish at isolated points.

The effect of scalings and translations on the support of a function is easily discerned.

Lemma 3.3. *If* $\operatorname{supp} f = [a, b]$, *and*

$$g(x) = f(r\,x - \delta), \qquad \textit{then} \qquad \operatorname{supp} g = \left[\, \frac{a + \delta}{r} \,,\, \frac{b + \delta}{r} \,\right].$$

In other words, scaling x by a factor r compresses the support of the function by a factor $1/r$, while translating x translates the support of the function.

The key requirement for a wavelet basis is that it contains functions with arbitrarily small support. To this end, the full Haar wavelet basis is obtained from the mother wavelet by iterating the scaling and translation processes. We begin with the scaling function

$$\varphi(x), \tag{3.7}$$

from which we construct the mother wavelet via (3.4). For any "generation" $j \geq 0$, we form the wavelet offspring by first compressing the mother wavelet so that its support fits into an interval of length 2^{-j},

$$w_{j,0}(x) = w(2^j x), \qquad \text{so that} \qquad \operatorname{supp} w_{j,0} = [\, 0, 2^{-j} \,], \tag{3.8}$$

and then translating $w_{j,0}$ so as to fill up the entire interval $[\, 0, 1 \,]$ by 2^j subintervals, each of length 2^{-j}, defining

$$w_{j,k}(x) = w_{j,0}(x - k) = w(2^j x - k), \qquad \text{where} \qquad k = 0, 1, \dots 2^j - 1. \tag{3.9}$$

Lemma 3.3 implies that $\operatorname{supp} w_{j,k} = [\, 2^{-j} k, 2^{-j} (k+1) \,]$, and so the combined supports of all the j^{th} generation of wavelets is the entire interval: $\bigcup_{k=0}^{2^j - 1} \operatorname{supp} w_{j,k} = [\, 0, 1 \,]$. The primal generation, $j = 0$, just consists of the mother wavelet

$$w_{0,0}(x) = w(x).$$

The first generation, $j = 1$, consists of the two daughter wavelets already introduced as φ_3 and φ_4, namely

$$w_{1,0}(x) = w(2x), \qquad\qquad w_{1,1}(x) = w(2x - 1).$$

The second generation, $j = 2$, appends four additional granddaughter wavelets to our basis:

$$w_{2,0}(x) = w(4x), \quad w_{2,1}(x) = w(4x - 1), \quad w_{2,2}(x) = w(4x - 2), \quad w_{2,3}(x) = w(4x - 3).$$

The 8 Haar wavelets $\varphi, w_{0,0}, w_{1,0}, w_{1,1}, w_{2,0}, w_{2,1}, w_{2,2}, w_{2,3}$ are constant on the 8 subintervals of length $\frac{1}{8}$, taking the successive sample values indicated by the columns of the matrix

$$W_8 = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & -1 & 0 & 0 & 0 \\ 1 & 1 & -1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & -1 & 0 & 0 & -1 & 0 & 0 \\ 1 & -1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & -1 & 0 & 1 & 0 & 0 & -1 & 0 \\ 1 & -1 & 0 & -1 & 0 & 0 & 0 & 1 \\ 1 & -1 & 0 & -1 & 0 & 0 & 0 & -1 \end{pmatrix}. \tag{3.10}$$

Orthogonality of the wavelets is manifested in the orthogonality of the columns of W_8. (Unfortunately, terminological constraints prevent us from calling W_8 an orthogonal matrix because its columns are not orthonormal!)

The n^{th} stage consists of 2^{n+1} different wavelet functions comprising the scaling functions and all the generations up to the n^{th} : $w_0(x) = \varphi(x)$ and $w_{j,k}(x)$ for $0 \leq j \leq n$ and $0 \leq k < 2^j$. They are all constant on each subinterval of length 2^{-n-1}.

Theorem 3.4. *The wavelet functions $\varphi(x)$, $w_{j,k}(x)$ form an orthogonal system with respect to the inner product* (3.1).

Proof: First, note that each wavelet $w_{j,k}(x)$ is equal to $+1$ on an interval of length 2^{-j-1} and to -1 on an adjacent interval of the same length. Therefore,

$$\langle\, w_{j,k}\,,\varphi\,\rangle = \int_0^1 w_{j,k}(x)\,dx = 0, \tag{3.11}$$

since the $+1$ and -1 contributions cancel each other. If two different wavelets $w_{j,k}$ and $w_{l,m}$ with, say $j \le l$, have supports which are either disjoint, or just overlap at a single point, then their product $w_{j,k}(x)\,w_{l,m}(x) \equiv 0$, and so their inner product is clearly zero:

$$\langle\, w_{j,k}\,,w_{l,m}\,\rangle = \int_0^1 w_{j,k}(x)\,w_{l,m}(x)\,dx = 0.$$

Otherwise, except in the case when the two wavelets are identical, the support of $w_{l,m}$ is entirely contained in an interval where $w_{j,k}$ is constant and so $w_{j,k}(x)\,w_{l,m}(x) = \pm\, w_{l,m}(x)$. Therefore, by (3.11),

$$\langle\, w_{j,k}\,,w_{l,m}\,\rangle = \int_0^1 w_{j,k}(x)\,w_{l,m}(x)\,dx = \pm \int_0^1 w_{l,m}(x)\,dx = 0.$$

Finally, we compute

$$\|\,\varphi\,\|^2 = \int_0^1 dx = 1, \qquad\qquad \|\,w_{j,k}\,\|^2 = \int_0^1 w_{j,k}(x)^2\,dx = 2^{-j}. \tag{3.12}$$

The second formula follows from the fact that $|\,w_{j,k}(x)\,| = 1$ on an interval of length 2^{-j} and is 0 elsewhere. *Q.E.D.*

In direct analogy with the trigonometric Fourier series, the *wavelet series* of a signal $f(x)$ is given by

$$f(x) \;\sim\; c_0\,\varphi(x) \;+\; \sum_{j=0}^{\infty}\sum_{k=0}^{2^j-1} c_{j,k}\,w_{j,k}(x). \tag{3.13}$$

Orthogonality implies that the wavelet coefficients $c_0, c_{j,k}$ can be immediately computed using the standard inner product formula coupled with (3.12):

$$\begin{aligned} c_0 &= \frac{\langle\, f\,,\varphi\,\rangle}{\|\,\varphi\,\|^2} = \int_0^1 f(x)\,dx, \\ c_{j,k} &= \frac{\langle\, f\,,w_{j,k}\,\rangle}{\|\,w_{j,k}\,\|^2} = 2^j \int_{2^{-j}k}^{2^{-j}k+2^{-j-1}} f(x)\,dx \;-\; 2^j \int_{2^{-j}k+2^{-j-1}}^{2^{-j}(k+1)} f(x)\,dx. \end{aligned} \tag{3.14}$$

The convergence properties of the wavelet series (3.13) are similar to those of Fourier series; details can be found in [**5**, **14**].

Example 3.5. In Figure 8, we plot the Haar expansions of the signal in the first plot. The next plots show the partial sums over $j = 0, \dots, r$ with $r = 2, 3, 4, 5, 6$. We have used a discontinuous signal to demonstrate that there is no nonuniform Gibbs phenomenon in a Haar wavelet expansion. Indeed, since the wavelets are themselves discontinuous, they do not have any difficult uniformly converging to a discontinuous function. On the other

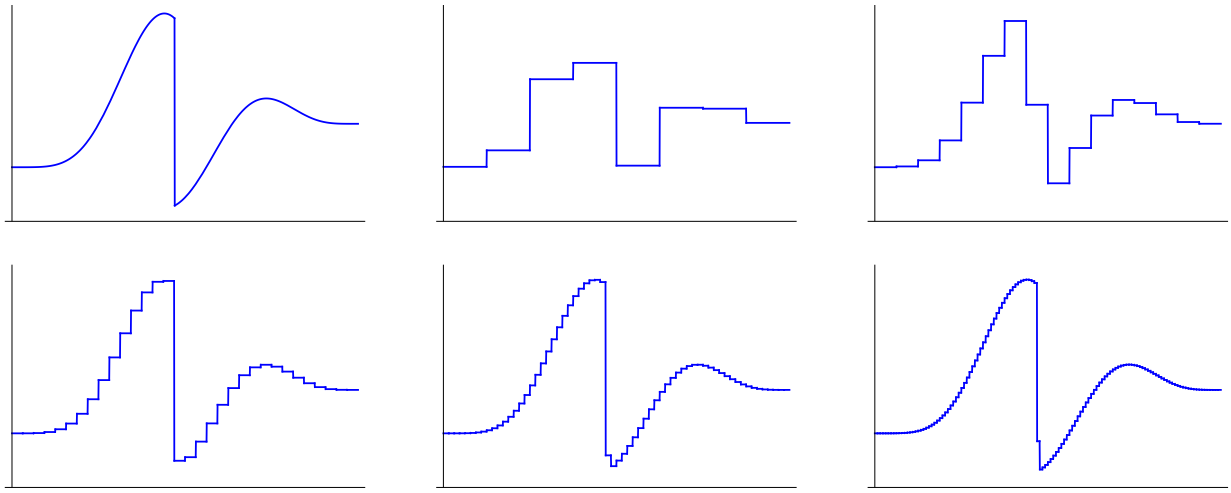

Figure 8. Haar Wavelet Expansion.

hand, it takes quite a few wavelets to begin to accurately reproduce the signal. In the last plot, we combine a total of $2^6 = 64$ Haar wavelets, which is considerably more than would be required in a comparably accurate Fourier expansion (excluding points very close to the discontinuity).

Remark: To the novice, there may appear to be many more wavelets than trigonometric functions. But this is just another illusion of the magic show of infinite dimensional space. The point is that they both form a countably infinite set of functions, and so could, if necessary, but less conveniently, be numbered in order $1, 2, 3, \dots$. On the other hand, accurate reproduction of functions usually does require many more Haar wavelets. This handicap makes the Haar system of less use in practical situations, and served to motivate the search for a more sophisticated choice of wavelet basis.

Just as the discrete Fourier representation arises from a sampled version of the full Fourier series, so there is a discrete wavelet transformation for suitably sampled signals. To take full advantage of the wavelet basis, we sample the signal $f(x)$ at $n = 2^r$ equally spaced sample points $x_k = k/2^r$, for $k = 0, \dots, n-1$, on the interval $[0, 1]$. As before, we can identify the sampled signal with the vector

$$\mathbf{f} = \big(f(x_0), f(x_1), \dots, f(x_{n-1}) \big)^T = \big(f_0, f_1, \dots, f_{n-1} \big)^T \in \mathbb{R}^n. \tag{3.15}$$

Since we are only sampling on intervals of length 2^{-r}, the discrete wavelet transform of our sampled signal will only use the first $n = 2^r$ wavelets $\varphi(x)$ and $w_{j,k}(x)$ for $j = 0, \dots, r-1$ and $k = 0, \dots, 2^j - 1$. Let $\mathbf{w}_0 \sim \varphi(x)$ and $\mathbf{w}_{j,k} \sim w_{j,k}(x)$ denote the corresponding sampled wavelet vectors; all of their entries are either $+1$, -1, or 0. (When $r = 3$, so $n = 8$, these are the columns of the wavelet matrix (3.10).) Moreover, orthogonality of the wavelets immediately implies that the wavelet vectors form an orthogonal basis of $\mathbb{R}^n$ with $n = 2^r$. We can decompose our sample vector (3.15) as a linear combination of the

sampled wavelets,

$$\mathbf{f} = \widehat{c}_0\, \mathbf{w}_0 \;+\; \sum_{j=0}^{r-1} \sum_{k=0}^{2^j-1} \widehat{c}_{j,k}\, \mathbf{w}_{j,k}, \tag{3.16}$$

where, by our usual orthogonality formulae,

$$\begin{aligned} \widehat{c}_0 &= \frac{\langle\, f\,, \mathbf{w}_0\,\rangle}{\|\,\mathbf{w}_0\,\|^2} = \frac{1}{2^r} \sum_{i=0}^{2^r-1} f_i, \\ \widehat{c}_{j,k} &= \frac{\langle\, f\,, \mathbf{w}_{j,k}\,\rangle}{\|\,\mathbf{w}_{j,k}\,\|^2} = 2^{j-r} \left(\sum_{i=k}^{k+2^{r-j-1}-1} f_i \;-\; \sum_{i=k+2^{r-j-1}}^{k+2^{r-j}-1} f_i \right). \end{aligned} \tag{3.17}$$

These are the basic formulae connecting the functions $f(x)$, or, rather, its sample vector $\mathbf{f}$, and its *discrete wavelet transform* consisting of the 2^r coefficients $\widehat{c}_0, \widehat{c}_{j,k}$. The reconstructed function

$$\widetilde{f}(x) = \widehat{c}_0\, \varphi(x) + \sum_{j=0}^{r-1} \sum_{k=0}^{2^j-1} \widehat{c}_{j,k}\, w_{j,k}(x) \tag{3.18}$$

is constant on each subinterval of length 2^{-r}, and has the same value

$$\widetilde{f}(x) = \widetilde{f}(x_i) = f(x_i) = f_i, \qquad\qquad x_i = 2^{-r}\, i \le x < x_{i+1} = 2^{-r}\,(i+1),$$

as our signal at the left hand endpoint of the interval. In other words, we are interpolating the sample points by a piecewise *constant* (and thus discontinuous) function.

Modern Wavelets

The main defect of the Haar wavelets is that they do not provide a very efficient means of representing even very simple functions — it takes quite a large number of wavelets to reproduce signals with any degree of precision. The reason for this is that the Haar wavelets are piecewise constant, and so even an affine function $y = \alpha\, x + \beta$ requires many sample values, and hence a relatively extensive collection of Haar wavelets, to be accurately reproduced. In particular, compression and denoising algorithms based on Haar wavelets are either insufficiently precise or hopelessly inefficient, and hence of minor practical value.

For a long time it was thought that it was impossible to simultaneously achieve the requirements of localization, orthogonality and accurate reproduction of simple functions. The breakthrough came in 1988, when, in her Ph.D. thesis, the Dutch mathematician Ingrid Daubechies produced the first examples of wavelet bases that realized all three basic criteria. Since then, wavelets have developed into a sophisticated and burgeoning industry with major impact on modern technology. Significant applications include compression, storage and recognition of fingerprints in the FBI's data base, and the JPEG2000 image format, which, unlike earlier Fourier-based JPEG standards, incorporates wavelet technology in its image compression and reconstruction algorithms. In this section, we will present a brief outline of the basic ideas underlying Daubechies' remarkable construction.

The recipe for any wavelet system involves two basic ingredients — a scaling function and a mother wavelet. The latter can be constructed from the scaling function by a

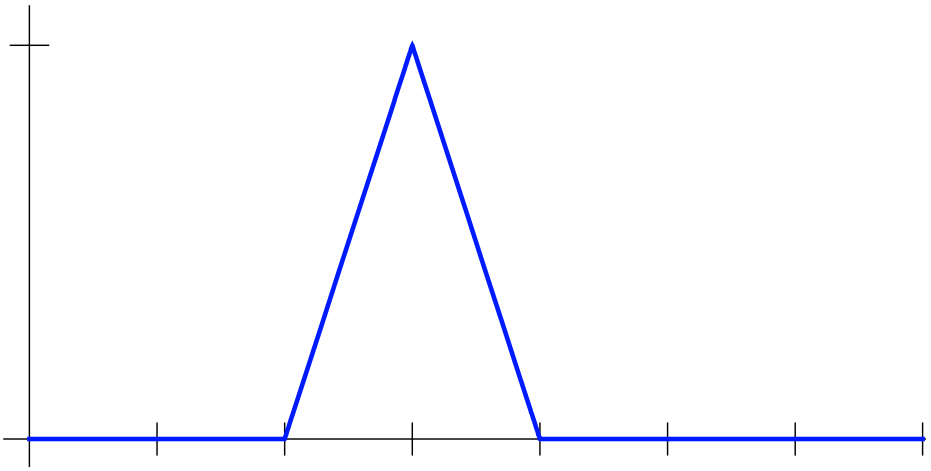

Figure 9. The Hat Function.

prescription similar to that in (3.4), and therefore we first concentrate on the properties of the scaling function. The key requirement is that the scaling function must solve a *dilation equation* of the form

$$\varphi(x) = \sum_{k=0}^{p} c_k \, \varphi(2x - k) = c_0 \, \varphi(2x) + c_1 \, \varphi(2x - 1) + \, \cdots \, + c_p \, \varphi(2x - p) \tag{3.19}$$

for some collection of constants $c_0, \ldots, c_p$. The dilation equation relates the function $\varphi(x)$ to a finite linear combination of its compressed translates. The coefficients $c_0, \ldots, c_p$ are not arbitrary, since the properties of orthogonality and localization will impose certain rather stringent requirements.

Example 3.6. The Haar or box scaling function (3.2) satisfies the dilation equation (3.19) with $c_0 = c_1 = 1$, namely

$$\varphi(x) = \varphi(2x) + \varphi(2x - 1). \tag{3.20}$$

We recommend that you convince yourself of the validity of this identity before continuing.

Example 3.7. Another example of a scaling function is the *hat function*

$$\varphi(x) = \begin{cases} x, & 0 \le x \le 1, \\ 2 - x, & 1 \le x \le 2, \\ 0, & \text{otherwise}, \end{cases} \tag{3.21}$$

graphed in Figure 9, whose variants play a starring role in the finite element method, [**10**, **13**]. The hat function satisfies the dilation equation

$$\varphi(x) = \tfrac{1}{2} \, \varphi(2x) + \varphi(2x - 1) + \tfrac{1}{2} \, \varphi(2x - 2), \tag{3.22}$$

which is (3.19) with $c_0 = \frac{1}{2}$, $c_1 = 1$, $c_2 = \frac{1}{2}$. Again, the reader should be able to check this identity by hand.

The dilation equation (3.19) is a kind of *functional equation*, and, as such, is not so easy to solve. Indeed, the mathematics of functional equations remains much less well developed than that of differential equations or integral equations. Even to prove that (nonzero) solutions exist is a nontrivial analytical problem. Since we already know two explicit examples, let us defer the discussion of solution techniques until we understand how the dilation equation can be used to construct a wavelet basis.

Given a solution to the dilation equation, we define the *mother wavelet* to be

$$\begin{aligned} w(x) &= \sum_{k=0}^{p} (-1)^k c_{p-k}\, \varphi(2\,x - k) \\ &= c_p\, \varphi(2\,x) - c_{p-1}\, \varphi(2\,x - 1) + c_{p-2}\, \varphi(2\,x - 2) + \,\cdots\, \pm c_0\, \varphi(2\,x - p), \end{aligned} \tag{3.23}$$

This formula directly generalizes the Haar wavelet relation (3.4), in light of its dilation equation (3.20). The daughter wavelets are then all found, as in the Haar basis, by iteratively compressing and translating the mother wavelet:

$$w_{j,k}(x) = w(2^j\, x - k). \tag{3.24}$$

In the general framework, we do not necessarily restrict our attention to the interval $[\,0,1\,]$ and so j and k can, in principle, be arbitrary integers.

Let us investigate what sort of conditions should be imposed on the dilation coefficients $c_0, \dots, c_p$ in order that we obtain a viable wavelet basis by this construction. First, localization of the wavelets requires that the scaling function has bounded support, and so $\varphi(x) \equiv 0$ when x lies outside some bounded interval $[\,a,b\,]$. If we integrate both sides of (3.19), we find

$$\int_a^b \varphi(x)\, dx = \int_{-\infty}^{\infty} \varphi(x)\, dx = \sum_{k=0}^{p} c_k \int_{-\infty}^{\infty} \varphi(2\,x - k)\, dx. \tag{3.25}$$

Now using the change of variables $y = 2\,x - k$, with $dx = \frac{1}{2}\, dy$, we find

$$\int_{-\infty}^{\infty} \varphi(2\,x - k)\, dx = \frac{1}{2} \int_{-\infty}^{\infty} \varphi(y)\, dy = \frac{1}{2} \int_a^b \varphi(x)\, dx, \tag{3.26}$$

where we revert to x as our (dummy) integration variable. We substitute this result back into (3.25). Assuming that $\displaystyle\int_a^b \varphi(x)\, dx \neq 0$, we discover that the dilation coefficients must satisfy

$$c_0 + \,\cdots\, + c_p = 2. \tag{3.27}$$

Example 3.8. Once we impose the constraint (3.27), the very simplest version of the dilation equation is

$$\varphi(x) = 2\, \varphi(2\,x) \tag{3.28}$$

where $c_0 = 2$ is the only (nonzero) coefficient. Up to constant multiple, the only "solutions" of the functional equation (3.28) with bounded support are scalar multiples of the delta function $\delta(x)$. Other solutions, such as $\varphi(x) = 1/x$, are not localized, and thus not useful for constructing a wavelet basis.

The second condition we require is orthogonality of the wavelets. For simplicity, we only consider the standard L^2 inner product[†]

$$\langle\, f\,,g\,\rangle = \int_{-\infty}^{\infty} f(x)\, g(x)\, dx.$$

It turns out that the orthogonality of the complete wavelet system is guaranteed once we know that the scaling function $\varphi(x)$ is orthogonal to all its integer translates:

$$\langle\, \varphi(x)\,, \varphi(x-m)\,\rangle = \int_{-\infty}^{\infty} \varphi(x)\, \varphi(x-m)\, dx = 0 \qquad \text{for all} \qquad m \neq 0. \tag{3.29}$$

We first note the formula

$$\begin{aligned} \langle\, \varphi(2\,x-k)\,, \varphi(2\,x-l)\,\rangle &= \int_{-\infty}^{\infty} \varphi(2\,x-k)\, \varphi(2\,x-l)\, dx \\ &= \frac{1}{2}\int_{-\infty}^{\infty} \varphi(x)\, \varphi(x+k-l)\, dx = \frac{1}{2}\,\langle\, \varphi(x)\,, \varphi(x+k-l)\,\rangle \end{aligned} \tag{3.30}$$

follows from the same change of variables $y = 2\,x - k$ used in (3.26). Therefore, since φ satisfies the dilation equation (3.19),

$$\langle\, \varphi(x)\,, \varphi(x-m)\,\rangle = \left\langle \sum_{j=0}^{p} c_j\, \varphi(2\,x-j)\,,\, \sum_{k=0}^{p} c_k\, \varphi(2\,x-2\,m-k)\, \right\rangle \tag{3.31}$$

$$= \sum_{j,k=0}^{p} c_j\, c_k\, \langle\, \varphi(2\,x-j)\,, \varphi(2\,x-2\,m-k)\,\rangle = \frac{1}{2}\sum_{j,k=0}^{p} c_j\, c_k\, \langle\, \varphi(x)\,, \varphi(x+j-2\,m-k)\,\rangle.$$

If we require orthogonality (3.29) of all the integer translates of φ, then the left hand side of this identity will be 0 unless $m = 0$, while only the summands with $j = 2\,m + k$ will be nonzero on the right. Therefore, orthogonality requires that

$$\sum_{0\,\le\,k\,\le\,p-2m} c_{2m+k}\, c_k = \begin{cases} 2, & m = 0, \\ 0, & m \neq 0. \end{cases} \tag{3.32}$$

The algebraic equations (3.27, 32) for the dilation coefficients are the key requirements for the construction of an orthogonal wavelet basis.

For example, if we have just two nonzero coefficients c_0, c_1, then (3.27, 32) reduce to

$$c_0 + c_1 = 2, \qquad\qquad c_0^2 + c_1^2 = 2,$$

and so $c_0 = c_1 = 1$ is the only solution, resulting in the Haar dilation equation (3.20). If we have three coefficients c_0, c_1, c_2, then (3.27), (3.32) require

$$c_0 + c_1 + c_2 = 2, \qquad\qquad c_0^2 + c_1^2 + c_2^2 = 2, \qquad\qquad c_0\, c_2 = 0.$$

[†] In all instances, the functions have bounded support, and so the inner product integral can be reduced to an integral over a finite interval where both f and g are nonzero.

Thus either $c_2 = 0,\ c_0 = c_1 = 1$, and we are back to the Haar case, or $c_0 = 0,\ c_1 = c_2 = 1$, and the resulting dilation equation is a simple reformulation of the Haar case. In particular, the hat function (3.21) does *not* give rise to orthogonal wavelets.

The remarkable fact, discovered by Daubechies, is that there *is* a nontrivial solution for four (and, indeed, any even number) of nonzero coefficients c_0, c_1, c_2, c_3. The basic equations (3.27), (3.32) require

$$c_0 + c_1 + c_2 + c_3 = 2, \qquad c_0^2 + c_1^2 + c_2^2 + c_3^2 = 2, \qquad c_0\, c_2 + c_1\, c_3 = 0. \tag{3.33}$$

The particular values

$$c_0 = \frac{1+\sqrt{3}}{4}, \qquad c_1 = \frac{3+\sqrt{3}}{4}, \qquad c_2 = \frac{3-\sqrt{3}}{4}, \qquad c_3 = \frac{1-\sqrt{3}}{4}, \tag{3.34}$$

solve (3.33). These coefficients correspond to the *Daubechies dilation equation*

$$\varphi(x) = \frac{1+\sqrt{3}}{4}\,\varphi(2\,x) + \frac{3+\sqrt{3}}{4}\,\varphi(2\,x-1) + \frac{3-\sqrt{3}}{4}\,\varphi(2\,x-2) + \frac{1-\sqrt{3}}{4}\,\varphi(2\,x-3). \tag{3.35}$$

A nonzero solution of bounded support to this remarkable functional equation will give rise to a scaling function $\varphi(x)$, a mother wavelet

$$w(x) = \frac{1-\sqrt{3}}{4}\,\varphi(2\,x) - \frac{3-\sqrt{3}}{4}\,\varphi(2\,x-1) + \frac{3+\sqrt{3}}{4}\,\varphi(2\,x-2) - \frac{1+\sqrt{3}}{4}\,\varphi(2\,x-3), \tag{3.36}$$

and then, by compression and translation (3.24), the complete system of orthogonal wavelets $w_{j,k}(x)$.

Before explaining how to solve the Daubechies dilation equation, let us complete our discussion of orthogonality. It is easy to see that, by translation invariance of the inner product integral, since $\varphi(x)$ and $\varphi(x-m)$ are orthogonal whenever $m \neq 0$, so are $\varphi(x-k)$ and $\varphi(x-l)$ for all $k \neq l$. Next we seek to establish orthogonality of $\varphi(x-m)$ and $w(x)$. Combining the dilation equation (3.19) and the definition (3.23) of w, and then using (3.29, 30), produces

$$\begin{aligned}
\langle\, w(x)\,, \varphi(x-m)\,\rangle &= \left\langle\, \sum_{j=0}^{p} (-1)^j c_{p-j}\, \varphi(2\,x-j)\,,\, \sum_{k=0}^{p} c_k\, \varphi(2\,x-2\,m-k)\,\right\rangle \\
&= \sum_{j,k=0}^{p} (-1)^j\, c_{p-j}\, c_k\, \langle\, \varphi(2\,x-j)\,, \varphi(2\,x-2\,m-k)\,\rangle \\
&= \sum_{j,k=0}^{p} (-1)^j\, c_{p-j}\, c_k\, \langle\, \varphi(x)\,, \varphi(x+j-2\,m-k)\,\rangle = \frac{1}{2}\sum_{k} (-1)^k\, c_{p-2m-k}\, c_k\, \|\,\varphi\,\|^2,
\end{aligned}$$

where the sum is over all $0 \leq k \leq p$ such that $0 \leq 2\,m + k \leq p$. Now, if $p = 2q+1$ is odd, then each term in the final summation appears twice, with opposite signs, and hence the result is always zero — no matter what the coefficients $c_0, \dots, c_p$ are! On the other hand, if $p = 2\,q$ is even, then orthogonality requires all $c_0 = \cdots = c_p = 0$, and hence

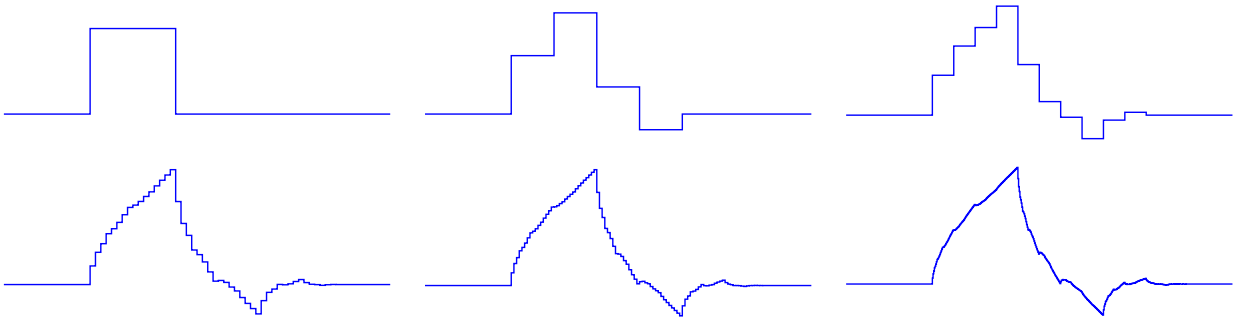

Figure 10. Approximating the Daubechies Wavelet.

$\varphi(x) \equiv 0$ is completely trivial and not of interest. Indeed, the particular cases $m = \pm q$ require $c_0 = c_p = 0$; with this, setting $m = \pm(q-1)$ requires $c_1 = c_{p-1} = 0$, and so on. Thus, to ensure orthogonality of the wavelet basis, the dilation equation (3.19) necessarily has an even number of terms, meaning that p must be an odd integer, as it is in the Haar and Daubechies versions (but not for the hat function). The proof of orthogonality of the translates $w(x-m)$ of the mother wavelet, along with all her wavelet descendants $w(2^j x - k)$, relies on a similar argument, and the details are left as an exercise for the reader.

Solving the Dilation Equation

Let us next discuss how to solve the dilation equation (3.19). The solution we are after does not have an elementary formula, and we require a slightly sophisticated approach to recover it. The key observation is that (3.19) has the form of a fixed point equation

$$\varphi = F[\varphi],$$

not in ordinary Euclidean space, but in an infinite-dimensional function space. With luck, the fixed point (or, more correctly, fixed function) will be stable, and so starting with a suitable initial guess $\varphi_0(x)$, the successive iterates

$$\varphi_{n+1} = F[\varphi_n]$$

will converge to the desired solution: $\varphi_n(x) \longrightarrow \varphi(x)$. In detail, the iterative version of the dilation equation (3.19) reads

$$\varphi_{n+1}(x) = \sum_{k=0}^{p} c_k \, \varphi_n(2x-k), \qquad n = 0, 1, 2, \dots . \tag{3.37}$$

Before attempting to prove convergence of this iterative procedure to the Daubechies scaling function, let us experimentally investigate what happens.

A reasonable choice for the initial guess might be the Haar scaling or box function

$$\varphi_0(x) = \begin{cases} 1, & 0 < t \le 1. \\ 0, & \text{otherwise.} \end{cases}$$

In Figure 10, the subsequent iterates $\varphi_1(x), \varphi_2(x), \varphi_4(x), \varphi_5(x), \varphi_7(x)$. There clearly appears to be convergence to some function $\varphi(x)$, although the final result looks a little bizarre. Bolstered by this preliminary experimental evidence, we can now try to prove convergence of the iterative scheme. This turns out to be true; a fully rigorous proof relies on the Fourier transform, and can be found in [**5**].

Theorem 3.9. *The functions converge $\varphi_n(x)$ defined by the iterative functional equation* (3.37) *converge uniformly to a continuous function $\varphi(x)$, called the* Daubechies scaling function.

Once we have established convergence, we are now able to verify that the scaling function and consequential system of wavelets form an orthogonal system of functions.

Proposition 3.10. *All integer translates $\varphi(x-k)$, for $k \in \mathbb{Z}$ of the Daubechies scaling function, and all wavelets $w_{j,k}(x) = w(2^j x - k)$, $j \geq 0$, are mutually orthogonal functions with respect to the* L^2 *inner product. Moreover, $\| \varphi \|^2 = 1$, while $\| w_{j,k} \|^2 = 2^{-j}$.*

Proof: As noted earlier, the orthogonality of the entire wavelet system will follow once we know the orthogonality (3.29) of the scaling function and its integer translates. We use induction to prove that this holds for all the iterates $\varphi_n(x)$, and so, in view of uniform convergence, the limiting scaling function also satisfies this property. We already know that the orthogonality property holds for the Haar scaling function $\varphi_0(x)$. To demonstrate the induction step, we repeat the computation in (3.31), but now the left hand side is $\langle\, \varphi_{n+1}(x)\,, \varphi_{n+1}(x-m)\,\rangle$, while all other terms involve the previous iterate φ_n. In view of the the algebraic constraints (3.32) on the wavelet coefficients and the induction hypothesis, we deduce that $\langle\, \varphi_{n+1}(x)\,, \varphi_{n+1}(x-m)\,\rangle = 0$ whenever $m \neq 0$, while when $m = 0$, $\| \varphi_{n+1} \|^2 = \| \varphi_n \|^2$. Since $\| \varphi_0 \| = 1$, we further conclude that all the iterates, and hence the limiting scaling function, all have unit L^2 norm. The proof of formula for the norms of the mother and daughter wavelets is left as an exercise for the reader. *Q.E.D.*

In practical computations, the limiting procedure for constructing the scaling function is not so convenient, and an alternative means of computing its values is employed. The starting point is to determine its values at integer points. First, the initial box function has values $\varphi_0(m) = 0$ for all integers $m \in \mathbb{Z}$ except $\varphi_0(1) = 1$. The iterative functional equation (3.37) will then produce the values of the iterates $\varphi_n(m)$ at integer points $m \in \mathbb{Z}$. A simple induction will convince you that $\varphi_n(m) = 0$ except for $m = 1$ and $m = 2$, and, therefore, by (3.37),

$$\varphi_{n+1}(1) = \frac{3+\sqrt{3}}{4}\,\varphi_n(1) + \frac{1+\sqrt{3}}{4}\,\varphi_n(2), \qquad \varphi_{n+1}(2) = \frac{1-\sqrt{3}}{4}\,\varphi_n(1) + \frac{3-\sqrt{3}}{4}\,\varphi_n(2),$$

since all other terms are 0. This has the form of a linear iterative system

$$\mathbf{v}^{(n+1)} = A\,\mathbf{v}^{(n)} \tag{3.38}$$

with coefficient matrix

$$A = \begin{pmatrix} \dfrac{3+\sqrt{3}}{4} & \dfrac{1+\sqrt{3}}{4} \\[2ex] \dfrac{1-\sqrt{3}}{4} & \dfrac{3-\sqrt{3}}{4} \end{pmatrix} \qquad \text{and where} \qquad \mathbf{v}^{(n)} = \begin{pmatrix} \varphi_n(1) \\ \varphi_n(2) \end{pmatrix}.$$

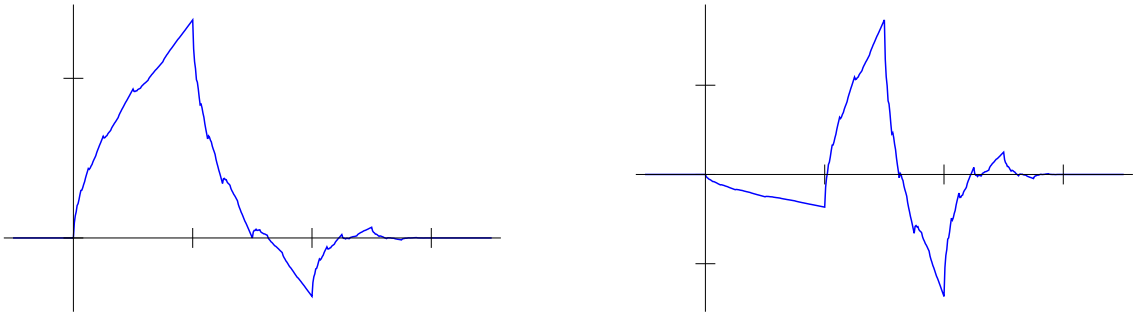

Figure 11. The Daubechies Scaling Function and Mother Wavelet.

Referring to [**11**; Chapter 9], the solution to such an iterative system is specified by the eigenvalues and eigenvectors of the coefficient matrix, which are

$$\lambda_1 = 1, \qquad \mathbf{v}_1 = \begin{pmatrix} \frac{1+\sqrt{3}}{4} \\ \frac{1-\sqrt{3}}{4} \end{pmatrix}, \qquad \lambda_2 = \tfrac{1}{2}, \qquad \mathbf{v}_2 = \begin{pmatrix} -1 \\ 1 \end{pmatrix}.$$

We write the initial condition as a linear combination of the eigenvectors

$$\mathbf{v}^{(0)} = \begin{pmatrix} \varphi_0(1) \\ \varphi_0(2) \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} = 2\,\mathbf{v}_1 - \frac{1-\sqrt{3}}{2}\,\mathbf{v}_2.$$

The solution is

$$\mathbf{v}^{(n)} = A^n\,\mathbf{v}^{(0)} = 2\,A^n\mathbf{v}_1 - \frac{1-\sqrt{3}}{2}\,A^n\mathbf{v}_2 = 2\,\mathbf{v}_1 - \frac{1}{2^n}\,\frac{1-\sqrt{3}}{2}\,\mathbf{v}_2.$$

The limiting vector

$$\begin{pmatrix} \varphi(1) \\ \varphi(2) \end{pmatrix} = \lim_{n\to\infty} \mathbf{v}^{(n)} = 2\,\mathbf{v}_1 = \begin{pmatrix} \frac{1+\sqrt{3}}{2} \\ \frac{1-\sqrt{3}}{2} \end{pmatrix}$$

gives the desired values of the scaling function:

$$\varphi(1) = \frac{1-\sqrt{3}}{2} = 1.366025\ldots, \qquad \varphi(2) = \frac{-1+\sqrt{3}}{2} = -.366025\ldots, \qquad \varphi(m) = 0, \quad \text{for all} \quad m \neq 1, 2. \tag{3.39}$$

With this in hand, the Daubechies dilation equation (3.35) then prescribes the function values $\varphi\big(\frac{1}{2}\,m\big)$ at all half integers, because when $x = \frac{1}{2}\,m$ then $2\,x - k = m - k$ is an integer. Once we know its values at the half integers, we can re-use equation (3.35) to give its values at quarter integers $\frac{1}{4}\,m$. Continuing onwards, we determine the values of $\varphi(x)$ at all *dyadic points*, meaning rational numbers of the form $x = m/2^j$ for $m, j \in \mathbb{Z}$. Continuity will then prescribe its value at any other $x \in \mathbb{R}$ since x can be written as the limit of dyadic numbers x_n — namely those obtained by truncating its binary (base 2) expansion at the n^{th} digit beyond the decimal (or, rather "binary") point. But, in practice, this latter step is unnecessary, since all computers are ultimately based on the binary number system, and so only dyadic numbers actually reside in a computer's memory. Thus, there is no real need to determine the value of φ at non-dyadic points.

The preceding scheme was used to produce the graphs of the Daubechies scaling function in Figure 11. It is continuous, but non-differentiable function — and its graph has a very jagged, fractal-like appearance when viewed at close range. The Daubechies scaling function is, in fact, a close relative of the famous example of a continuous, nowhere differentiable function originally due to Weierstrass, [**7**, **8**], whose construction also relies on a similar scaling argument.

With the values of the Daubechies scaling function on a sufficiently dense set of dyadic points in hand, the consequential values of the mother wavelet are given by formula (3.36). Note that $\operatorname{supp} \varphi = \operatorname{supp} w = [0, 3]$. The daughter wavelets are then found by the usual compression and translation procedure (3.24).

The Daubechies wavelet expansion of a function whose support is contained in[†] $[0, 1]$ is then given by

$$f(x) \;\sim\; c_0\, \varphi(x) + \sum_{j=0}^{\infty} \sum_{k=-2}^{2^j-1} c_{j,k}\, w_{j,k}(x). \tag{3.40}$$

The inner summation begins at $k = -2$ so as to include *all* the wavelet offspring $w_{j,k}$ whose support has a nontrivial intersection with the interval $[0, 1]$. The wavelet coefficients $c_0, c_{j,k}$ are computed by the usual orthogonality formula

$$\begin{aligned} c_0 &= \langle\, f\,, \varphi\,\rangle = \int_0^3 f(x)\, \varphi(x)\, dx, \\ c_{j,k} &= \langle\, f\,, w_{j,k}\,\rangle = 2^j \int_{2^{-j}\,k}^{2^{-j}\,(k+3)} f(x)\, w_{j,k}(x)\, dx = \int_0^3 f\big(\, 2^{-j}\,(x+k)\,\big)\, w(x)\, dx, \end{aligned} \tag{3.41}$$

where we agree that $f(x) = 0$ whenever $x < 0$ or $x > 1$. In practice, one employs a numerical integration procedure, e.g., the trapezoid rule, based on dyadic nodes to speedily evaluate the integrals (3.41). A proof of completeness of the resulting wavelet basis functions can be found in [**5**]. Compression and denoising algorithms based on retaining only low frequency modes proceed as before, and are left as exercises for the reader to implement.

Example 3.11. In Figure 12, we plot the Daubechies wavelet expansions of the same signal for Example 3.5. The first plot is the original signal, and the following show the partial sums of (3.40) over $j = 0, \dots, r$ with $r = 2, 3, 4, 5, 6$. Unlike the Haar expansion, the Daubechies wavelets do exhibit a nonuniform Gibbs phenomenon, where the expansion noticeably overshoots near the discontinuity, [**10**], which can be observed at the interior discontinuity as well as the endpoints, since the function is set to 0 outside the interval $[0, 1]$. Indeed, the Daubechies wavelets are continuous, and so cannot converge uniformly to a discontinuous function.

[†] For functions with larger support, one should include additional terms in the expansion corresponding to further translates of the wavelets so as to cover the entire support of the function. Alternatively, one can translate and rescale x to fit the function's support inside $[0, 1]$.

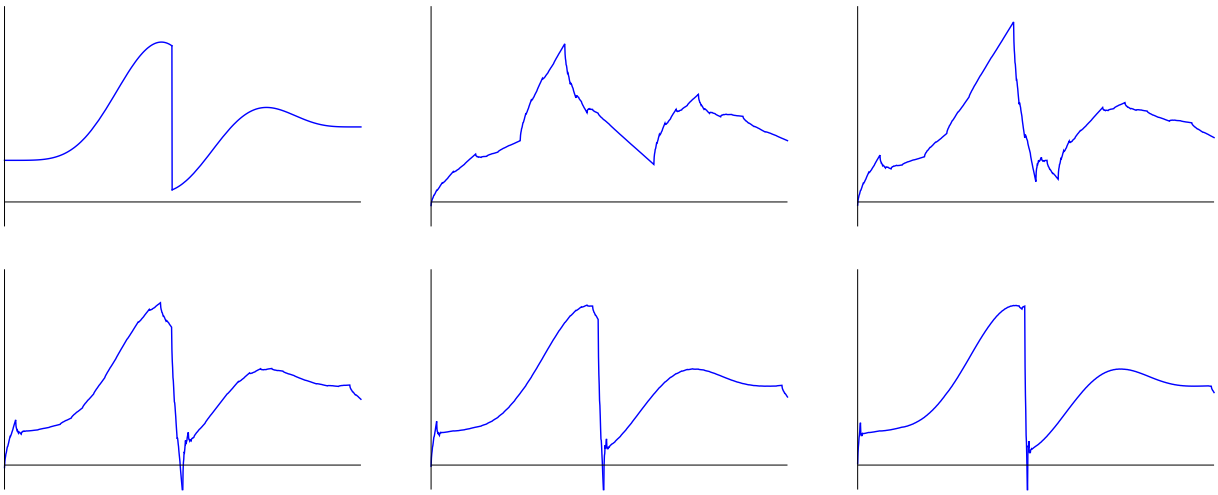

Figure 12. Daubechies Wavelet Expansion.

4. The Laplace Transform.

In engineering applications, the Fourier transform is often overshadowed by a close relative. The Laplace transform plays an essential role in control theory, linear systems analysis, electronics, and many other fields of practical engineering and science. However, the Laplace transform is most properly interpreted as a particular real form of the more fundamental Fourier transform. When the Fourier transform is evaluated along the imaginary axis, the complex exponential factor becomes real, and the result is the Laplace transform, which maps real-valued functions to real-valued functions. Since it is so closely allied to the Fourier transform, the Laplace transform enjoys many of its featured properties, including linearity. Moreover, derivatives are transformed into algebraic operations, which underlies its applications to solving differential equations. The Laplace transform is one-sided; it only looks forward in time and prefers functions that decay — transients. The Fourier transform looks in both directions and prefers oscillatory functions. For this reason, while the Fourier transform is used to solve boundary value problems on the real line, the Laplace transform is much better adapted to initial value problems.

Since we will be applying the Laplace transform to initial value problems, we switch our variable from x to t to emphasize this fact. Suppose $f(t)$ is a (reasonable) function which vanishes on the negative axis, so $f(t) = 0$ for all $t < 0$. The Fourier transform of f is

$$\widehat{f}(k) = \frac{1}{\sqrt{2\,\pi}} \int_0^\infty f(t)\, e^{-\,\mathrm{i}\,k\,t}\, dt,$$

since, by our assumption, its negative t values make no contribution to the integral. The *Laplace transform* of such a function is obtained by replacing $\mathrm{i}\,k$ by a real[†] variable s,

[†] One can also define the Laplace transform at complex values of s, but this will not be required in the applications discussed here.

leading to

$$F(s) = \mathcal{L}[\, f(t)\,] = \int_0^\infty f(t)\, e^{-\,s\,t}\, dt, \tag{4.1}$$

where, in accordance with the standard convention, the factor of $\sqrt{2\,\pi}$ has been omitted. By allowing complex values of the Fourier frequency variable k, we may identify the Laplace transform with $\sqrt{2\,\pi}$ times the evaluation of the Fourier transform for values of $k = -\,\mathrm{i}\,s$ on the imaginary axis:

$$F(s) = \sqrt{2\,\pi}\;\widehat{f}\,(-\,\mathrm{i}\,s). \tag{4.2}$$

Since the exponential factor in the integral has become real, the Laplace transform $\mathcal{L}$ takes real functions to real functions. Moreover, since the integral kernel $e^{-\,s\,t}$ is exponentially decaying for $s > 0$, we are no longer required to restrict our attention to functions that decay to zero as $t \to \infty$.

Example 4.1. Consider an exponential function $f(t) = e^{\alpha\,t}$, where the exponent α is allowed to be complex. Its Laplace transform is

$$F(s) = \int_0^\infty e^{(\alpha - s)\,t}\, dt = \frac{1}{s - \alpha}\,. \tag{4.3}$$

Note that the integrand is exponentially decaying, and hence the integral converges, if and only if $\mathrm{Re}\,(\alpha - s) < 0$. Therefore, the Laplace transform (4.3) is, strictly speaking, only defined at sufficiently large $s > \mathrm{Re}\,\alpha$. In particular, for an oscillatory exponential,

$$\mathcal{L}[\, e^{\,\mathrm{i}\,\omega\,t}\,] = \frac{1}{s - \,\mathrm{i}\,\omega} = \frac{s + \,\mathrm{i}\,\omega}{s^2 + \omega^2} \qquad \text{provided} \qquad s > 0.$$

Taking real and imaginary parts of this identity, we discover the formulae for the Laplace transforms of the simplest trigonometric functions:

$$\mathcal{L}[\cos \omega\, t\,] \;=\; \frac{s}{s^2 + \omega^2}\,, \qquad\qquad \mathcal{L}[\sin \omega\, t\,] \;=\; \frac{\omega}{s^2 + \omega^2}\,. \tag{4.4}$$

Two additional important transforms are

$$\mathcal{L}[\,1\,] = \int_0^\infty e^{-\,s\,t}\, dt = \frac{1}{s}\,, \qquad \mathcal{L}[\,t\,] = \int_0^\infty t\, e^{-\,s\,t}\, dt = \frac{1}{s}\int_0^\infty e^{-\,s\,t}\, dt = \frac{1}{s^2}\,. \tag{4.5}$$

The second computation relies on an integration by parts, making sure that the boundary terms at $s = 0, \infty$ vanish.

Remark: In every case, we really mean the Laplace transform of the function whose values are given for $t > 0$ and is equal to 0 for all negative t. Therefore, the function 1 in reality signifies the step function

$$\sigma(t) = \begin{cases} 1, & t > 0, \\ 0, & t < 0, \end{cases} \tag{4.6}$$

and so the first formula in (4.5) should more properly be written

$$\mathcal{L}[\,\sigma(t)\,] = \frac{1}{s}\,. \tag{4.7}$$

However, in the traditional approach to the Laplace transform, one only considers the functions on the positive t axis, and so the step function and the constant function are, from this viewpoint, indistinguishable. However, once one moves beyond a purely mechanistic approach, any deeper understanding of the properties of the Laplace transform requires keeping this distinction firmly in mind.

Let us now pin down the precise class of functions to which the Laplace transform can be applied.

Definition 4.2. A function $f(t)$ is said to have *exponential growth* of *order* a if

$$| f(t) | < M e^{a t}, \qquad \text{for all} \qquad t > t_0, \tag{4.8}$$

for some $M > 0$ and $t_0 > 0$.

Note that the exponential growth condition only depends upon the function's behavior for large values of t. If $a < 0$, then f is, in fact, exponentially decaying as $x \to \infty$. Since $e^{a t} < e^{b t}$ for $a < b$ and all $t > 0$, if $f(t)$ has exponential growth of order a, it automatically has exponential growth of any higher order $b > a$. All polynomial, trigonometric, and exponential functions (with linear argument) have exponential growth. The simplest example of a function that does not satisfy any exponential growth bound is $f(t) = e^{t^2}$, since it grows faster than any simple exponential $e^{a t}$.

The following result guarantees the existence of the Laplace transform, at least for sufficiently large values of the transform variable s, for a rather broad class of functions that includes almost all of the functions that arise in applications.

Theorem 4.3. *If $f(t)$ is piecewise continuous and has exponential growth of order a, then its Laplace transform $F(s) = \mathcal{L}[\, f(t)\,]$ is defined for all $s > a$.*

Proof: The exponential growth inequality (4.8) implies that we can bound the integrand in (4.1) by $| f(t)\, e^{-s t} | < M e^{(a-s) t}$. Therefore, as soon as $s > a$, the integrand is exponentially decaying as $t \to \infty$, and this suffices to ensure the convergence of the Laplace transform integral. *Q.E.D.*

Theorem 4.3 is an existential result, and of course, in practice, we may not be able to explicitly evaluate the Laplace transform integral. Nevertheless, the Laplace transforms of most common functions are not hard to find, and extensive lists have been tabulated, [**9**]. An abbreviated table of Laplace transforms can be found on the following page. Nowadays, the most convenient sources of transform formulas are computer algebra packages, including MATHEMATICA and MAPLE.

According to [**10**; Theorem 8.15], when it exists, the Fourier transform uniquely specifies the function, except possibly at jump discontinuities where the limiting value must be half way in between. An analogous result can be established for the Laplace transform.

Lemma 4.4. *If $f(t)$ and $g(t)$ are piecewise continuous functions that are of exponential growth, and $\mathcal{L}[\, f(t)\,] = \mathcal{L}[\, g(t)\,]$ for all s sufficiently large, then $f(t) = g(t)$ at all points of continuity $t > 0$.*

Table of Laplace Transforms

$f(t)$	$F(s)$
1	$\dfrac{1}{s}$
t	$\dfrac{1}{s^2}$
t^n	$\dfrac{n!}{s^{n+1}}$
$\delta(t-c)$	$e^{-s\,c}$
$e^{\alpha\,t}$	$\dfrac{1}{s-\alpha}$
$\cos\omega\,t$	$\dfrac{s}{s^2+\omega^2}$
$\sin\omega\,t$	$\dfrac{\omega}{s^2+\omega^2}$
$e^{c\,t}\,f(t)$	$F(s-c)$
$\sigma(t-c)\,f(t-c)$	$e^{-s\,c}\,F(s)$
$t\,f(t)$	$-F'(s)$
$f'(t)$	$s\,F(s)-f(0)$
$f^{(n)}(t)$	$s^n\,F(s)-s^{n-1}\,f(0)-$ $-s^{n-2}\,f'(0)-\cdots-f^{(n-1)}(0)$
$f(t)*g(t)$	$F(s)\,G(s)$

In this table, n is a non-negative integer, ω is any real number, while $c \geq 0$ is any non-negative real number.

In fact, there is an explicit formula for the inverse Laplace transform, which follows from its identification, (4.2), with the Fourier transform along the imaginary axis. Under suitable hypotheses, a given function $F(s)$ is the Laplace transform of the function $f(t)$ determined by the complex integral formula[†]

$$f(t) = \frac{1}{2\,\pi\,\mathrm{i}} \int_{-\,\mathrm{i}\,\infty}^{\,\mathrm{i}\,\infty} F(s)\, e^{s\,t}\, ds, \qquad\qquad t > 0. \tag{4.9}$$

In practice, one hardly ever uses this complicated formula to compute the inverse Laplace transform. Rather, one simply relies on tables of known Laplace transforms, coupled with a few basic rules that will be covered in the following subsection.

The Laplace Transform Calculus

The first observation is that the Laplace transform is a linear operator, and so

$$\mathcal{L}[\,f + g\,] = \mathcal{L}[\,f\,] + \mathcal{L}[\,g\,], \qquad\qquad \mathcal{L}[\,c\,f\,] = c\,\mathcal{L}[\,f\,], \tag{4.10}$$

for any constant c. Moreover, just like its Fourier progenitor, the Laplace transform converts calculus into algebra. In particular, differentiation turns into multiplication by the transform variable, but with one additional term that depends upon the value of the function at $t = 0$.

Theorem 4.5. *Let $f(t)$ be continuously differentiable for $t > 0$ and have exponential growth of order a. If $\mathcal{L}[\,f(t)\,] = F(s)$ then, for $s > a$,*

$$\mathcal{L}[\,f'(t)\,] = s\,F(s) - f(0). \tag{4.11}$$

Proof: The proof relies on an integration by parts:

$$\begin{aligned} \mathcal{L}[\,f'(t)\,] &= \int_0^\infty f'(t)\, e^{-\,s\,t}\, dt = \, f(t)\, e^{-\,s\,t}\,\Big|_{t\,=\,0}^{\infty} + s \int_0^\infty f(t)\, e^{-\,s\,t}\, dt \\ &= \lim_{t\,\to\,\infty} f(t)\, e^{-\,s\,t} - f(0) + s\,F(s). \end{aligned}$$

The exponential growth inequality (4.8) implies that first term vanishes for $s > a$, and the remaining terms agree with (4.11). *Q.E.D.*

Example 4.6. According to Example 4.1, the Laplace transform of the function $\sin \omega\, t$ is

$$\mathcal{L}[\sin \omega\, t\,] = \frac{\omega}{s^2 + \omega^2}\,.$$

Its derivative is

$$\frac{d}{dt} \sin \omega\, t = \omega \cos \omega\, t,$$

[†] See [**1**, **12**] for details on complex integration. The stated formula doesn't apply to all functions of exponential growth. A more universally valid inverse Laplace transform formula is obtained by shifting the complex contour to run from $b - \mathrm{i}\,\infty$ to $b + \mathrm{i}\,\infty$ for some $b > a$, the order of exponential growth of f.

and therefore

$$\mathcal{L}[\,\omega\cos\omega t\,] = s\,\mathcal{L}[\,\sin\omega t\,] = \frac{\omega\, s}{s^2+\omega^2}\,,$$

since $\sin\omega t$ vanishes at $t = 0$. The result agrees with (4.4). On the other hand,

$$\frac{d}{dt}\cos\omega t = -\omega\sin\omega t,$$

and so

$$\mathcal{L}[\,-\omega\sin\omega t\,] \;=\; s\,\mathcal{L}[\,\cos\omega t\,] - 1 = \frac{s^2}{s^2+\omega^2} - 1 = -\,\frac{\omega^2}{s^2+\omega^2}\,,$$

again in agreement with the known formula.

Remark: The final term $-f(0)$ in (4.11) is a manifestation of the discontinuity in $f(t)$ at $t = 0$. Keep in mind that the Laplace transform only applies to functions with $f(t) = 0$ for all $t < 0$, and so if $f(0) \neq 0$, then $f(t)$ has a jump discontinuity of magnitude $f(0)$ at $t = 0$. Therefore, by the calculus of generalized functions, its derivative $f'(t)$ should include a delta function term, namely $f(0)\,\delta(0)$, which accounts for the additional constant term in its transform. In the practical approach to the Laplace transform calculus, one suppresses the delta function when computing the derivative $f'(t)$. However, its effect must reappear on the other side of the differentiation formula (4.11), and the upshot is the extra term $-f(0)$.

Laplace transforms of higher order derivatives are found by iterating the first order formula (4.11). For example, if $f \in \mathrm{C}^2$, then

$$\mathcal{L}[\,f''(t)\,] = s\,\mathcal{L}[\,f'(t)\,] - f'(0) = s^2\,F(s) - s\,f(0) - f'(0). \tag{4.12}$$

In general, for an n times continuously differentiable function,

$$\mathcal{L}[\,f^{(n)}(t)\,] = s^n F(s) - s^{n-1} f(0) - s^{n-2} f'(0) - \;\cdots\; - f^{(n-1)}(0). \tag{4.13}$$

On the other hand, multiplying the function by t corresponds to differentiation of its Laplace transform (up to a change of sign):

$$\mathcal{L}[\,t\,f(t)\,] = -\,F'(s). \tag{4.14}$$

The proof of this formula is left as an exercise for the reader.

Conversely, integration corresponds to dividing the Laplace transform by s, so

$$\mathcal{L}\left[\,\int_0^t f(\tau)\,d\tau\,\right] = \frac{F(s)}{s}\,. \tag{4.15}$$

Unlike the Fourier transform, there are no additional terms in the integration formula as long as we start the integral at $t = 0$. For instance,

$$\mathcal{L}[\,t^2\,] = \frac{1}{s}\,\mathcal{L}[\,2t\,] = \frac{2}{s^3}\,, \qquad \text{and, more generally,} \qquad \mathcal{L}[\,t^n\,] = \frac{n!}{s^{n+1}}\,. \tag{4.16}$$

There is also a shift formula, but with one important caveat. Since all functions must vanish for $t < 0$, we are only allowed to shift them to the right, a shift to the left would

produce nonzero function values for some $t < 0$. In general, the Laplace transform of the function $f(t-c)$ shifted to the right by an amount $c > 0$, is

$$\mathcal{L}[\,f(t-c)\,] = \int_0^\infty f(t-c)\,e^{-s\,t}\,dt = \int_{-c}^\infty f(t)\,e^{-s\,(t+c)}\,dt \tag{4.17}$$
$$= \int_{-c}^0 f(t)\,e^{-s\,(t+c)}\,dt + \int_0^\infty f(t)\,e^{-s\,(t+c)}\,dt = e^{-s\,c}\int_0^\infty f(t)\,e^{-s\,t}\,dt = e^{-s\,c}\,F(s).$$

In this computation, we first used a change of variables in the integral, replacing $t-c$ by t; then, the fact that $f(t) \equiv 0$ for $t < 0$ was used to eliminate the integral from $-c$ to 0. When using the shift formula in practice, it is important to keep in mind that $c > 0$ and the function $f(t-c)$ vanishes for all $t < c$. In the table, the factor $\sigma(t-c)$ is used to remind the user of this fact.

Example 4.7. Consider the square wave pulse $f(t) = \begin{cases} 1, & b < t < c, \\ 0, & \text{otherwise}, \end{cases}$ for some $0 < b < c$. To compute its Laplace transform, we write it as the difference

$$f(t) = \sigma(t-b) - \sigma(t-c)$$

of shifted versions of the step function (4.6). Combining the shift formula (4.17) and the formula (4.7) for the Laplace transform of the step function, we find

$$\mathcal{L}[\,f(t)\,] = \mathcal{L}[\,\sigma(t-b)\,] - \mathcal{L}[\,\sigma(t-c)\,] = \frac{e^{-s\,b} - e^{-s\,c}}{s}\,. \tag{4.18}$$

We already noted that the Fourier transform of the convolution product of two functions is realized as the ordinary product of their individual transforms. A similar result holds for the Laplace transform. Let $f(t), g(t)$ be given functions. Since we are implicitly assuming that the functions vanish at all negative values of t, their convolution product reduces to a finite integral

$$h(t) = f(t) * g(t) = \int_0^t f(t-\tau)\,g(\tau)\,d\tau. \tag{4.19}$$

In particular $h(t) = 0$ for all $t < 0$ also. Further, it is not hard to show that the convolution of two functions of exponential growth also has exponential growth.

Theorem 4.8. *If $\mathcal{L}[\,f(t)\,] = F(s)$ and $\mathcal{L}[\,g(t)\,] = G(s)$, then the convolution $h(t) = f(t) * g(t)$ has Laplace transform given by the product $H(s) = F(s)\,G(s)$.*

The proof of the convolution theorem for the Laplace transform proceeds along the same lines as its Fourier transform version, [**10**; Theorem 8.13].

Applications to Initial Value Problems

The key application of the Laplace transform is to facilitate the solution of initial value problems for linear, constant coefficient ordinary differential equations. As a prototypical example, consider the second order initial value problem

$$a\,\frac{d^2u}{dt^2} + b\,\frac{du}{dt} + c\,u = f(t), \qquad u(0) = \alpha, \qquad \frac{du}{dt}\,(0) = \beta, \tag{4.20}$$

in which a, b, c are constant. We will solve the initial value problem by applying the Laplace transform to both sides of the differential equation. In view of the differentiation formulae (4.11, 12),

$$a\big(\, s^2 \mathcal{L}[\, u(t)\,] - s\, u(0) - \dot{u}(0)\,\big) + b\big(\, s\, \mathcal{L}[\, u(t)\,] - u(0)\,\big) + c\, \mathcal{L}[\, u(t)\,] = \mathcal{L}[\, f(t)\,].$$

Setting $\mathcal{L}[\, u(t)\,] = U(s)$ and $\mathcal{L}[\, f(t)\,] = F(s)$, and making use of the initial conditions, the preceding equation takes the form

$$(a\, s^2 + b\, s + c)\, U(s) = F(s) + (a\, s + b)\, \alpha + a\, \beta. \tag{4.21}$$

Thus, by applying the Laplace transform, we have effectively reduced the entire initial value problem to a single elementary algebraic equation! Solving for

$$U(s) = \frac{F(s) + (a\, s + b)\, \alpha + a\, \beta}{a\, s^2 + b\, s + c}\,, \tag{4.22}$$

we then recover solution $u(t)$ to the initial value problem by finding the inverse Laplace transform of $U(s)$. As noted earlier, in practice the inverse transform is found by suitably massaging the answer (4.22) to be a combination of known transforms.

Example 4.9. Consider the initial value problem

$$\ddot{u} + u = 10\, e^{-3\, t}, \qquad\qquad u(0) = 1, \qquad \dot{u}(0) = 2.$$

Taking the Laplace transform of the differential equation as above, we find

$$(s^2 + 1)\, U(s) - s - 2 = \frac{10}{s + 3}\,, \qquad \text{and so} \qquad U(s) = \frac{s + 2}{s^2 + 1} + \frac{10}{(s + 3)(s^2 + 1)}.$$

The second summand does not directly correspond to any of the entries in our table of Laplace transforms. However, we can use the method of partial fractions to write it as a sum

$$U(s) = \frac{s + 2}{s^2 + 1} + \frac{1}{s + 3} + \frac{3 - s}{s^2 + 1} = \frac{1}{s + 3} + \frac{5}{s^2 + 1}$$

of terms appearing in the table. We conclude that the solution to our initial value problem is

$$u(t) = e^{-3\, t} + 5 \sin t.$$

Of course, the last example is a problem that you can easily solve directly. The standard method learned in your first course on differential equations is just as effective in finding the final solution, and does not require all the extra Laplace transform machinery! The Laplace transform method is, however, particularly effective when dealing with complications that arise in cases of discontinuous forcing functions.

Example 4.10. Consider a mass vibrating on a spring with fixed stiffness $c = 4$. Assume that the mass starts at rest, is then subjected to a unit force over time interval $\frac{1}{2}\pi < t < 2\,\pi$, after which it left to vibrate on its own. The initial value problem is

$$\frac{d^2 u}{dt^2} + 4\, u = \begin{cases} 1, & \frac{1}{2}\pi < t < 2\,\pi, \\ 0, & \text{otherwise}, \end{cases} \qquad\qquad u(0) = \dot{u}(0) = 0.$$

Taking the Laplace transform of the differential equation, and using (4.18), we find

$$(s^2+4)\,U(s) = \frac{e^{-\pi s/2} - e^{-2\pi s}}{s}\,, \qquad \text{and so} \qquad U(s) = \frac{e^{-\pi s/2} - e^{-2\pi s}}{s(s^2+4)}\,.$$

Therefore, by the shift formula (4.17)

$$u(t) = h\big(t - \tfrac{1}{2}\pi\big) - h(t - 2\pi),$$

where $h(t)$ is the function with Laplace transform

$$\mathcal{L}[\,h(t)\,] = H(s) = \frac{1}{s(s^2+4)} = \frac{1}{4}\left(\frac{1}{s} - \frac{s}{s^2+4}\right),$$

which has been conveniently rewritten using partial fractions. Referring to our table of Laplace transforms,

$$h(t) = \begin{cases} \frac{1}{4} - \frac{1}{4}\cos 2t, & t > 0, \\ 0, & t < 0. \end{cases}$$

Therefore, our desired solution is

$$u(t) = \begin{cases} 0, & 0 \le t \le \frac{1}{2}\pi, \\ \frac{1}{4} + \frac{1}{4}\cos 2t, & \frac{1}{2}\pi \le t \le 2\pi, \\ \frac{1}{2}\cos 2t, & 2\pi \le t. \end{cases}$$

Note that the solution $u(t)$ is only C^1 at the points of discontinuity of the forcing function.

Remark: A direct solution of this problem would proceed as follows. One solves the differential equation on each interval of continuity of the forcing function, leading to a solution on that interval depending upon two integration constants. The integration constants are then adjusted so that the solution satisfies the initial conditions and is continuously differentiable at each point of discontinuity of the forcing function. The details are straightforward, but messy. The Laplace transform method successfully bypasses these intervening manipulations.

Example 4.11. Consider the initial value problem

$$\frac{d^2u}{dt^2} + \omega^2\, u = f(t), \qquad u(0) = \dot{u}(0) = 0,$$

involving a general forcing function $f(t)$. Applying the Laplace transform to the differential equation and using the initial conditions,

$$(s^2 + \omega^2)\,U(s) = F(s), \qquad \text{and hence} \qquad U(s) = \frac{F(s)}{s^2+\omega^2}\,.$$

The right hand side is the product of the Laplace transform of the forcing function $f(t)$ and that of the trigonometric function $k(t) = \dfrac{\sin \omega t}{\omega}$. Therefore, Theorem 4.8 implies that the solution can be written as their convolution

$$u(t) = f * k(t) = \int_0^t k(t-\tau)\, f(\tau)\, d\tau = \int_0^t \frac{\sin \omega(t-\tau)}{\omega}\, f(\tau)\, d\tau. \tag{4.23}$$

where

$$k(t) = \begin{cases} \dfrac{\sin \omega t}{\omega}, & t > 0, \\ 0, & t < 0. \end{cases}$$

The integral kernel $k(t - \tau)$ is known as the *fundamental solution* to the initial value problem, and prescribes the response of the system to a unit impulse force that is applied instantaneously at the time $t = \tau$. Note particularly that (unlike boundary value problems) the impulse only affect the solutions at later times $t > \tau$. For initial value problems, the fundamental solution plays a role similar to that of a Green's function in a boundary value problem. The convolution formula (4.23) can be viewed as a linear superposition of fundamental solution responses induced by expressing the forcing function

$$f(t) = \int_0^\infty f(\tau)\, \delta(t - \tau)\, d\tau$$

as a superposition of concentrated impulses over time.

This concludes our brief introduction to the Laplace transform and a few of its many applications to physical problems. More details can be found in almost all applied texts on mechanics, electronic circuits, signal processing, control theory, and many other areas.

References

[**1**] Ahlfors, L., *Complex Analysis*, McGraw–Hill, New York, 1966.

[**2**] Brigham, E.O., *The Fast Fourier Transform*, Prentice–Hall, Inc., Englewood Cliffs, N.J., 1974.

[**3**] Briggs, W.L., *Henson, V.E.*, The DFT. An Owner's Manual for the Discrete Fourier Transform; SIAM, Philadelphia, PA, 1995.

[**4**] Cooley, J.W., and Tukey, J.W., An algorithm for the machine computation of complex Fourier series, *Math. Comp.* **19** (1965), 297–301.

[**5**] Daubechies, I., *Ten Lectures on Wavelets*, SIAM, Philadelphia, PA, 1992.

[**6**] Haar, A., Zur Theorie der orthogonalen Funktionensysteme, *Math. Ann.* **69** (1910), 331–371.

[**7**] Hobson, E.W., *The Theory of Functions of a Real Variable and the Theory of Fourier's Series*, Dover Publ., New York, 1957.

[**8**] Mandelbrot, B.B., *The Fractal Geometry of Nature*, W.H. Freeman, New York, 1983.

[**9**] Oberhettinger, F., and Badii, L., *Tables of Laplace Transforms*, Springer-Verlag, New York, 1973.

[**10**] Olver, P.J., *Introduction to Partial Differential Equations*, Undergraduate Texts in Mathematics, Springer, New York, 2014.

[**11**] Olver, P.J., and Shakiban, C., *Applied Linear Algebra*, Second Edition, Undergraduate Texts in Mathematics, Springer, New York, 2018.

[**12**] Saff, E.B., and Snider, A.D., *Fundamentals of Complex Analysis*, Third Ed., Prentice–Hall, Inc., Upper Saddle River, N.J., 2003.

[**13**] Strang, G., and Fix, G.J., *An Analysis of the Finite Element Method*, Prentice–Hall, Inc., Englewood Cliffs, N.J., 1973.

[**14**] Walter, G.G., and Shen, X., *Wavelets and Other Orthogonal Systems*, 2nd ed., Chapman & Hall/CRC, Boca Raton, Fl, 2001.

