



THE FAST FOURIER TRANSFORM FOR EXPERIMENTALISTS, PART I: CONCEPTS

By Denis Donnelly and Bert Rust

THE DISCRETE FOURIER TRANSFORM (DFT) PROVIDES A MEANS FOR TRANSFORMING DATA SAMPLED IN THE TIME DOMAIN TO AN EXPRESSION OF THIS DATA IN THE FREQUENCY

domain. The inverse transform reverses the process, converting frequency data into time-domain data. Such transformations can be applied in a wide variety of fields, from geophysics to astronomy, from the analysis of sound signals to CO₂ concentrations in the atmosphere. Over the course of three articles, our goal is to provide a convenient summary that the experimental practitioner will find useful. In the first two parts of this article, we'll discuss concepts associated with the fast Fourier transform (FFT), an implementation of the DFT. In the third part, we'll analyze two applications: a bat chirp and atmospheric sea-level pressure differences in the Pacific Ocean.

The FFT provides an efficient algorithm for implementing the DFT and, as such, we'll focus on it. This transform is easily executed; indeed, almost every available mathematical software package includes it as a built-in function. Some books are devoted solely to the FFT,¹⁻³ while others on signal processing,⁴⁻⁶ time series,^{7,8} or numerical methods^{9,10} include major sections on Fourier analysis and the FFT. We draw together here some of the basic elements that users need to apply and interpret the FFT and its inverse (IFFT). We will avoid descriptions of the Fourier matrix, which lies at the heart of the DFT process,¹¹ and the parsing of the Cooley-Tukey algorithm¹² (or any of several other comparable algorithms), which provides a means for transforming the discrete into the fast Fourier transform.

The Cooley-Tukey algorithm makes the FFT extremely useful by reducing the number of computations from something on the order of n^2 to $n \log(n)$, which obviously provides an enormous reduction in computation time. It's so useful, in fact, that the FFT made *Computing in Science & Engineering's* list of the top 10 algorithms in an article that noted the algorithm is, "perhaps, the most ubiquitous algo-

rithm in use today."¹³ The interlaced decomposition method used in the Cooley-Tukey algorithm can be applied to other orthogonal transformations such as the Hadamard, Hartley, and Haar. However, in this article, we concentrate on the FFT's application and interpretation.

Fundamental Elements

As a rule, data to be transformed consists of N uniformly spaced points $x_j = x(t_j)$, where $N = 2^n$ with n an integer, and $t_j = j \cdot \Delta t$ where j ranges from 0 to $N - 1$. (Some FFT implementations don't require that N be a power of 2. This number of points is, however, optimal for the algorithm's execution speed.) Even though any given data set is unlikely to have the number of its data points precisely equal to 2^n , zero padding (which we describe in more detail in the next section) provides a means to achieve this number of samples without losing information. As an additional restriction, we limit our discussions to real valued time series as most data streams are real. When the time-domain data are real, the values of the amplitude or power spectra at any negative frequency are the same as those at the corresponding positive frequency. Thus, if the time series is real, one half of the 2^n frequencies contain all the frequency information. In typical representations, the frequency domain contains $N/2 + 1$ samples.

The FFT's kernel is a sum of complex exponentials. Associated with this process are conventions for normalization, sign, and range. Here, we present what we consider to be good practice, but our choices are not universal. Users should always check the conventions of their particular software choice so they can properly interpret the computed transforms and related spectra.

Equation 1 shows some simple relationships between parameters such as Δt , the sampling time interval; Δf , the spacing in the frequency domain; N , the number of samples in the time domain; and f_j , the Fourier frequencies. The number of samples per cycle (spc) for a particular frequency component with period T in the time domain and (in some cases) the total number of cycles (nc) in the data record for a particular frequency component are two other pieces of information that are useful because they remind us of the adequacy of the sampling rate or the data sample. Some re-

lations between these parameters are

$$\Delta f = \frac{1}{N \cdot \Delta t} \text{ and } f_j = j \cdot \Delta f,$$

where $j = 0, \dots, N/2$

$$\text{spc} = \frac{T}{\Delta t}, \text{nc} = \frac{N}{\text{spc}} = \frac{1}{T \cdot \Delta f} = \frac{f}{\Delta f}. \quad (1)$$

The period T represents only one frequency, but, as we discuss later, there must be more than 2 spc for the highest frequency component of the sampled signal. This bandwidth-limiting frequency is called the *Nyquist frequency* and is equal to half the sampling frequency. The spacing in the frequency domain Δf is the inverse of the total time sampled, so time and frequency resolution can't both be simultaneously improved. Thus, the maximum frequency represented is $\Delta f \cdot N/2 = 1/(2 \cdot \Delta t)$, or the Nyquist frequency.

We can express the transform in several ways. A commonly used form is the following (with $i = \sqrt{-1}$):

$$X_k = \sum_{j=0}^{N-1} x_j \exp\left(-2\pi i \frac{j}{N} k\right), k = -N/2, \dots, -1, 0, 1, \dots, N/2 - 1, \quad (2)$$

where x_j represents the time-domain data and X_k their representation in the frequency domain.

We express the IFFT as

$$x_j = \frac{1}{N} \sum_{k=-N/2}^{N/2-1} X_k \cdot \exp\left(2\pi i \frac{k}{N} j\right), j = 0, 1, \dots, N-1. \quad (3)$$

The FFT replicates periodically on the frequency axis with a period of $1/\Delta t$; consequently, $X(f_{N/2}) = X(f_{-N/2})$ so that the transform is defined at both ends of the closed interval from $-1/(2\Delta t)$ to $+1/(2\Delta t)$. This interval is sometimes called the *Nyquist band*.

Some FFT and IFFT implementations use different normalizations or sign conventions. For example, some implementations place the factor $1/N$ in the FFT conversion rather than with the IFFT. Some place $1/\sqrt{N}$ in both conversion processes, and some reverse the signs in the exponentials of the transforms; this sign change reverses the sign of the phase component. Moreover, some implementations take the range for k from $0, \dots, N/2$.

Because Equations 2 and 3 represent the frequency and time domains of the same signal, the energy in the two cases must be the same. Parseval's relation expresses this equality.

For real data, we can express the relation as

$$\sum_{i=0}^{N-1} x_i^2 = \frac{1}{N} \left(|X_0|^2 + 2 \cdot \sum_{j=1}^{N/2-1} |X_j|^2 + |X_{N/2}|^2 \right), \quad (4)$$

where $X = \text{fft}(x)$. The last term on the right-hand side is not usually separated from the sum as it is here; we do this because there should be only N terms to consider in both summations, not N in one and $N+1$ in the other. Recall that because we're dealing with real valued data, we can exploit a symmetry and present the frequency data only from 0 to $N/2$; this symmetry is the source of the factor of two associated with the summation. Unlike the other terms, the $+N/2$ frequency value isn't independent and was assigned, as noted earlier, to the value at $-N/2$. Should the $+N/2$ term be included in the sum, we would, in effect, double count the term, so we pull the $N/2$ term from the sum to avoid this. Of course, if N is large, this difference is likely to be minimal.

There are two common ways to display an FFT. One is the amplitude spectrum, which presents the magnitudes of the FFT's complex values as a function of frequency:

$$A_k = \frac{2}{N} |X_k|, k = -N/2, \dots, -1, 0, 1, \dots, N/2. \quad (5)$$

Given the symmetry of real time series, the standard presentation restricts the range of k to positive values: $k = 0, 1, \dots, N/2$. An equally common way to represent the transform is with a power spectrum (or periodogram), which is defined as

$$P_k = \frac{1}{N} \cdot |X_k|^2, k = 0, 1, \dots, N/2. \quad (6)$$

However, neither of these spectral representations is universal. For example, some conventions place a 1 in the numerator instead of a 2 for the amplitude spectrum. The periodogram is sometimes represented with a factor of 2 in the numerator instead of 1 or as the individual terms expressed in Parseval's relation (Equation 4).

In Figure 1, as an example of the FFT process, we show the amplitude spectrum of a single-frequency sine wave with two different sampling intervals. In one case, the interval Δt is chosen to make nc integral, and in the other, nonintegral. If nc is integral, f is necessarily a multiple of Δf , and one point of the transform is associated with the true frequency (see the circles in Figure 1a). However, in any FFT application, we're dealing with a finite-length time series. The process of restricting the data in the time domain (multiplying the data by one over the range where we wish to keep

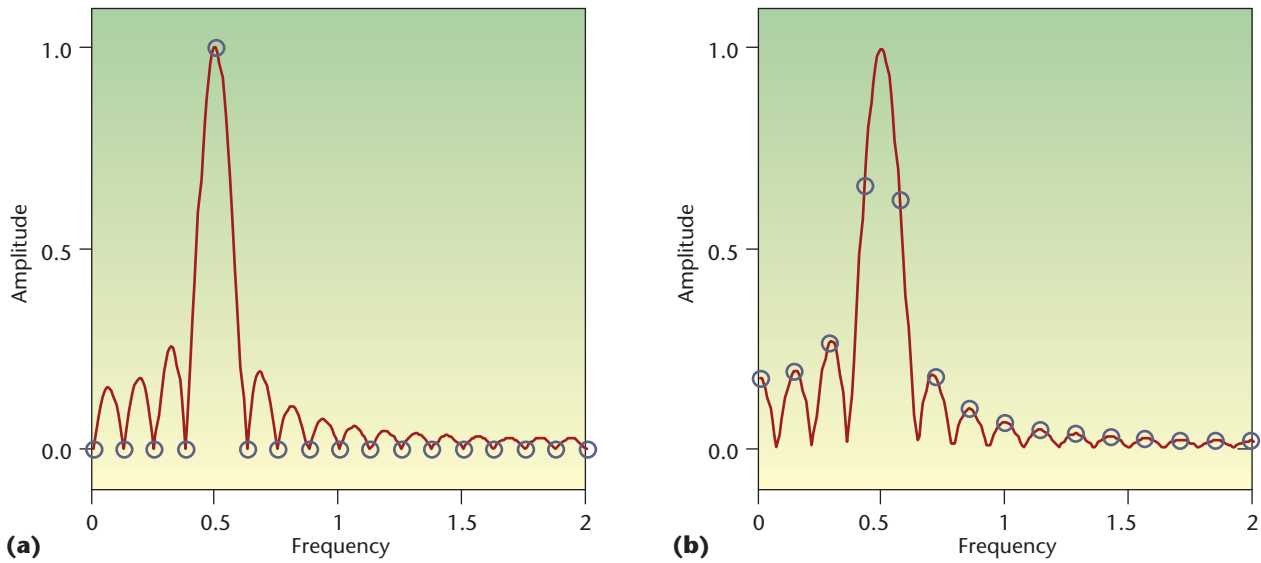


Figure 1. Amplitude spectra of a single-frequency sine wave. Two representations of a sine wave of frequency 0.5 are shown in each part of the figure. In each case, the circles are based on a time series where the number of sample points $N = 32$ but the time step is slightly different: (a) $N\Delta t = 8$, so $nc = 4$; (b) $N\Delta t = 7.04$, so $nc = 3.52$, where nc is the total number of cycles. The solid lines provide a view of these same spectra with zero padding. This form is closer to what would be expected from a continuous rather than a discrete Fourier transform. The zero-padded examples reveal detail that might not have been expected, given the appearance of the unpadded case.

the data and multiplying by zero elsewhere—an example of windowing, discussed later) introduces sidelobes in the frequency domain. These sidelobes are called leakage.

Even though there's leakage, because there's only one frequency associated with the transformed sine wave, we might expect to be able to estimate that frequency with a weighted average of all the points in the frequency domain. Such an average, however, wouldn't yield the correct frequency.

In general, the FFT process generates complex values in the frequency domain from the real values in the time domain. If we transform sine or cosine waves where we consider an integral number of cycles, the transform magnitudes are identical. However, in the frequency domain, a sine curve is represented only with imaginary values and a cosine curve only with real values. When the number of cycles is nonintegral or if there is a phase shift, then both real and imaginary parts appear in the transform of both the sine and cosine.

Zero Padding

Zero padding is a commonly used technique associated with FFTs. Two frequent uses are to make the number of data points in the time-domain sample a power of two and to improve interpolation in the transformed domain (for example, zero pad in the time domain, improve interpolation in the frequency domain).

Zero padding, as the name implies, means appending a string of zeros to the data. It doesn't make any difference if the zeros are appended at the end (the typical procedure), at

the beginning, or split between the beginning and end of the data set's time domain. One very common use of this process is to extend time-series data so that the number of samples becomes a power of two, making the conversion process more efficient or, with some software, simply possible. Because the spacing of data in the frequency domain is inversely proportional to the number of samples in the time domain, by increasing the number of samples—even if their values are zero—the resulting frequency spectrum will contain more data points for the same frequency range. Consequently, the zero-padded transform contains more data points than the unpadded; as a result, the overall process acts as a frequency interpolating function. The resulting, more detailed picture in the frequency space might indicate unexpected detail (see, for example, Figure 2). As the number of zeros increases, the FFT better represents the time series' continuous Fourier transform (CFT).

As we noted earlier, zero padding introduces more points into the same frequency range and provides interpolation between points associated with the unpadded case. When data points are more closely spaced, clearly, there's a possibility that unnoticed detail could be revealed (such as Figure 1a shows). In Figure 2, we see the effect of quadrupling the number of points for two different cases. The transforms of the zero-padded data contain the same information as the unpadded data, and every fourth point of the padded data matches the corresponding unpadded data point. The intermediate points provide interpolation.

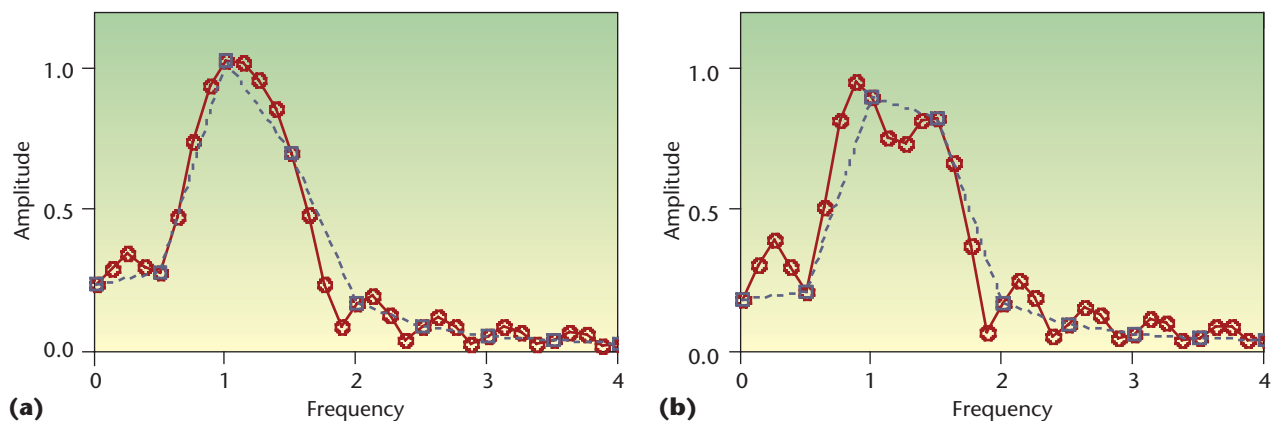


Figure 2. The effect of zero padding on the transform of a signal containing two different frequencies. We look at two cases: one in which the two frequencies are too close to be clearly resolved, and one in which resolution is possible. (a) Fast Fourier transforms (FFTs) of the sum of two sine waves of amplitude 1 and frequencies of 1 and 1.3 Hz; the frequencies aren't resolved, and (b) FFTs of the sum of two sine waves of amplitude 1 and frequencies of 1 and 1.35 Hz; the frequencies are resolved. The solid curves are transforms of zero-padded data and include four times as many samples as the transforms of the unpadded data (dotted curves). Because the zero-padded curve has four times as many data points as the unpadded case ($N = 32$), every fourth point of the zero-padded data is the same as the unpadded data. Zero-padded results provide better interpolation and more detail.

In Figure 2, we see an application of that interpolating ability when we consider a signal consisting of two closely lying frequencies. In Figure 2a, although the envelope is more clearly drawn, zero padding does not have the power to resolve the two frequencies associated with this case. In Figure 2b, the peaks are sufficiently separated so that the interpolation reveals the two peaks, whereas the unpadded data seemingly did not. This example reminds us that a graphical representation connecting adjacent data points with straight lines can be misleading.

Zero padding can also be performed in the frequency domain. The inverse transform results in an increase in the number of data points in the time domain, which could be useful in interpolating between samples (see Figure 3). Zero padding is also used in association with convolution or correlation and with filter kernels, which we discuss later in this article.

Aliasing

When performing an FFT, it's necessary to be aware of the frequency range composing the signal so that we sample the signal more than twice per cycle of the highest frequency associated with the signal. In practice, this might mean filtering the signals to block any signal components with a frequency above the Nyquist frequency ($2 \cdot \Delta t_{\text{sample}}^{-1}$) before performing a transform. If we don't restrict the signal in this way, higher frequencies will not be adequately sampled and will masquerade as lower-frequency signals. This effect is similar to what moviegoers experience when the onscreen wheels of a moving vehicle seemingly freeze or rotate in the wrong direction. The camera, which operates at the sampling rate of

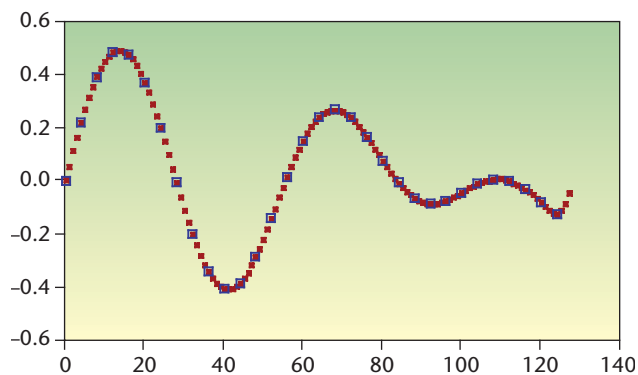


Figure 3. The effect of zero padding in the frequency domain on the time-domain data. The frequency data (the unpadded case in Figure 2a) was zero-padded to four times its original length. We show the original unpadded time-domain data (boxes) and the inverse fast Fourier transform of the zero-padded frequency data (dots). The padding process again acts as an interpolation function.

24 frames per second, only has a Nyquist limit of 12 Hz; any higher frequencies present will appear as lower frequencies.

Let's assume that we can readily observe a point on a wheel (not at the center) that's rotating but not translating. At a slow rotation rate, each successive frame of our film shows the observable point advancing from the previous frame. (The fraction of a complete rotation and the sampling rate are related; the number of samples per rotation is the inverse of the fraction of a rotation per sample.) As the rotation rate increases,

Table 1. Actual and apparent angles for 170° and 190° rotations.

Angle sequence for 170° step	Angle sequence for 190° step	Apparent angle sequence for 190° steps with rotation direction reversed.*
0	0	0
170	190	170
340	20	340
150	210	150
320	40	320
130	230	130

*Magnitudes of reverse angles are given by $360^\circ - \text{column 2}$.

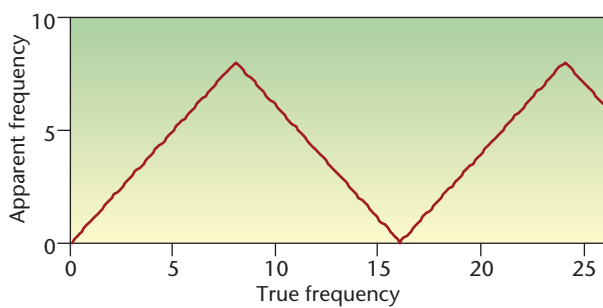


Figure 4. Apparent frequency as a function of the true frequency. Frequencies greater than the Nyquist frequency fold back into the allowed frequency range and appear as lower frequencies. In this example, where the Nyquist frequency is 8 Hz, an actual frequency of 9 Hz would appear as 7 Hz.

the angle between our observed point in successive frames increases. When the angle reaches 180 degrees, or two samples per rotation, the perceived rotation rate is at its maximum—the wheel is rotating at the Nyquist frequency.

When passing through the Nyquist limit, as the frequency goes from $f_{Ny} - \epsilon$ to $f_{Ny} + \epsilon$ (where $\epsilon \ll f_{Ny}$), the rotation direction appears to change from forward to reverse while the rotation rate remains the same. Further increases in the rotation rate make the wheel appear to continue rotating in a reversed direction but at a decreasing rate. When the actual rotation rate is twice the Nyquist frequency, the apparent rotation rate is zero and the sampling rate is just once per rotation. (Another example of one sample per rotation and an apparent zero rotation rate is to use a stroboscope to determine an object’s rotation rate. With one flash per rotation, the rotating object appears at rest and the flash rate and rotation rate are equal.) If the frequency of rotation continues to increase, the wheel will again appear to rotate in the original rotation direction.

To make this more concrete, consider two constant rotation rates, one of 170 degrees between successive frames/sam-

ples and one of 190 degrees. We observe only the current position in each frame, so as we compute a value sequence, we take them mod(360). If we compute values for the 170-degree case, we obtain 0, 170, 340, 150, 320, 130, and so on. If we compute values for the 190-degree case, we get 0, 190, 20, 210, 40, 230, and so on, but we wouldn’t see the 190-degree rotation. We don’t observe an increase greater than 180 degrees (for angles greater than that, the data is under-sampled). For the 190-degree case, we would see a 170-degree step, but with the rotation in the opposite direction.

To consider a reverse rotation, we subtract the forward rotation angle from 360. The result is the magnitude of the angle of rotation in the reverse direction. For example, a forward rotation angle of 350 degrees is equivalent to a 10-degree step in the reverse direction. So for our 190-degree case, the numbers become 0, $360 - 190 = 170$, $360 - 20 = 340$, $360 - 210 = 150$, and so on. Table 1 provides a summary. The magnitudes of these rotation angles are identical to the 170-degree data. Thus, we would see the 190-degree case as equivalent to the 170-degree case in terms of rotation rate, but with the rotation direction reversed. The graph in Figure 4 helps demonstrate this kind of behavior.

In the example shown in Figure 4, the Nyquist frequency is 8 Hz. Frequencies associated with the first leg of the sawtooth curve have more than two samples per cycle, and the apparent and actual frequencies are equal. Once the actual frequency exceeds the Nyquist frequency, the apparent frequency begins to decrease, with the negative slope corresponding to a reversed rotation direction. At 16 Hz, with one sample per rotation, the apparent frequency is zero. With further increases in the true frequency, the apparent frequency once again increases.

If we take the FFT of three amplitude 1 cosine waves having frequencies of 3.5, 12.5, and 19.5 Hz and where we set $N = 16$ and $\Delta t = 1/N$ (so the Nyquist frequency is 8 Hz), we get identical FFTs, one of which is shown in Figure 5. The number of samples per cycle for these frequencies is 4.57, 1.28, and 0.82. Only the lowest frequency is adequately represented; the two higher-frequency cases have fewer than

two samples per cycle and consequently masquerade as lower frequencies, appearing in the allowed range between 0 Hz and the Nyquist frequency. For the example with the three different frequencies, we purposely selected the higher frequencies so that their FFTs would be identical to that of the lowest frequency. Referring to Figure 4, we note that the frequencies 12.5 and 19.5 Hz would appear on the second and third legs of the sawtooth curve. The apparent frequency of the 12.5-Hz line is $8 - (12.5 - 8)$; the apparent frequency of the 19.5-Hz line is $19.5 - 2 \cdot 8$. In general, the out-of-range frequency f_{true} would appear as f_{apparent} as given by

$$f_{\text{apparent}} = \left| f_{\text{true}} - k \cdot (2 \cdot f_{\text{Nyquist}}) \right| = \left| f_{\text{true}} - \frac{k}{\Delta t} \right|, \quad (7)$$

where $k = 1, 2, \dots$, and k is selected to bring f_{apparent} within the range $0 \dots f_{\text{Ny}}$.

In Figure 6, we see the actual curves that correspond to the three frequencies and the points where sampling occurs. If we performed an FFT followed by an IFFT for any one of the three curves (given the sampling specified), the algorithm would return the same result in each case, which, without other information, would be interpreted as the lowest-frequency case.

If the magnitudes of the Fourier coefficients approach zero (roughly as $1/f$) as the frequency approaches the Nyquist frequency (a zero between lobes would not qualify), then there is a good likelihood that aliasing has not occurred. If it isn't zero, we can consider the possibility that it has occurred. However, a nonzero value doesn't imply that aliasing has necessarily happened. The Fourier coefficients in Figure 5 don't go to zero even in the adequately sampled case. Zero padding of this example will show a great deal more detail, but the transform is still nonzero at the Nyquist frequency.

Relation to Fourier Series

There is a direct connection between the real and imaginary parts of the frequency information from an FFT and the coefficients in a Fourier series that would represent the corresponding time-domain signal. As we noted earlier, for the conditions stated, the transform of a single-frequency sine wave is imaginary, whereas the transform of a single-frequency cosine wave is real. So, in a Fourier series of the time-domain signal, we would expect the real parts of the frequency information to be associated with cosine series and the imaginary parts with sine series. This is, in fact, the case.

An equation for recreating the original signal as a Fourier

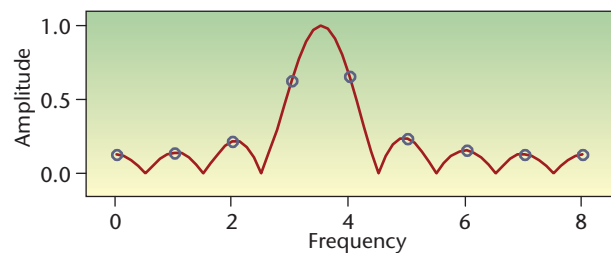


Figure 5. The FFT of a 3.5 Hz, amplitude one cosine wave where $N = 16$ and $\Delta t = 1/N$ (represented by circles). The FFTs of the frequencies 3.5 Hz, 12.5 Hz, and 19.5 Hz are identical for the case when the Nyquist frequency is 8 Hz. The solid curve shows the transform with zero padding.

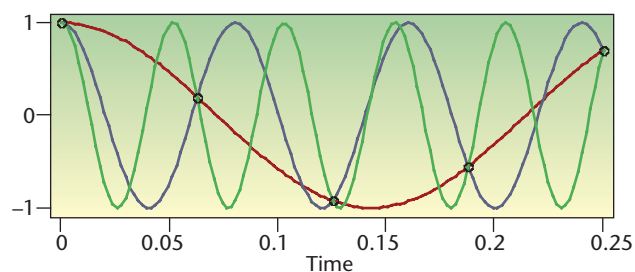


Figure 6. A view of the sampling of three cosine curves. Cosine curves with frequencies 3.5 Hz, 12.5 Hz, and 19.5 Hz are shown, with the marked points representing those at which sampling occurs ($\Delta t = 1/N$ and $N=16$). Only the lowest-frequency curve is adequately sampled, with more than two samples per cycle. In this case, the FFT for each curve would indicate a signal with a frequency of 3.5 Hz. For clarity, we show only the first five samples.

series from the frequency information is

$$S(t) = \left[\frac{a_0}{2} + \sum_{k=1}^{nt} \left[(a_k \cos(2\pi k \Delta f t)) + b_k \sin(2\pi k \Delta f t) \right] \right] \cdot \frac{2}{N}. \quad (8)$$

For the case $N = 2^n$, a_k represents the real part of the transformed signal, b_k the imaginary part, nt the number of terms to be included in the series (where $nt < N/2$), and Δf the spacing in the frequency domain.

An alternate form in terms of magnitude and phase is also possible. Given that

$$\varphi_j = \tan^{-1} \left(\frac{\text{Im}(b_j)}{\text{Re}(b_j)} \right), \quad (9)$$

where $b_k = a_k + ib_k$ and the H_j are the magnitudes of b_j , the series is given by

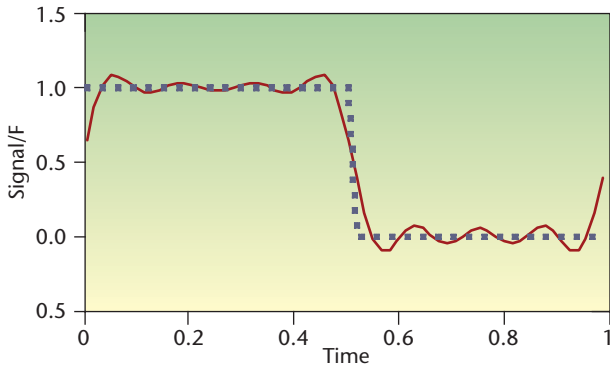


Figure 7. A comparison of the original time-domain signal and its partial reconstruction as a Fourier series. The original signal (dotted curve) and the first 10 terms of a Fourier series (solid curve) computed using coefficients from the original signal’s FFT.

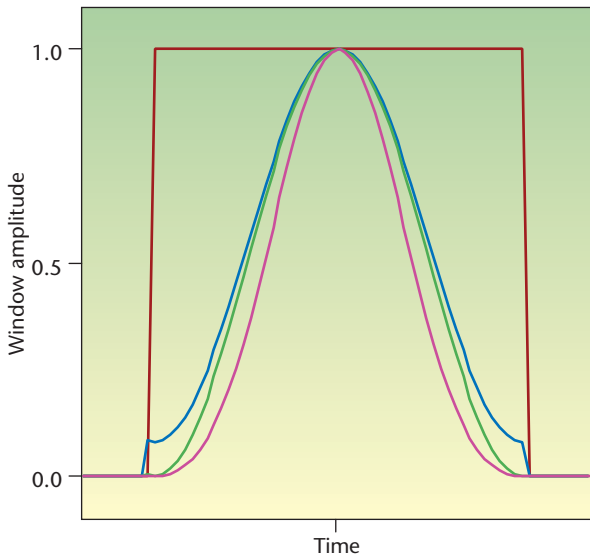


Figure 8. The shapes of four different windows. From the side, we see a rectangular (red), Hamming (blue), Hann (green), and Blackman (magenta), respectively. We’ll apply three of these windows to a sine wave sequence in Figure 9.

$$S(t) = \left[\frac{a_0}{2} + \sum_{k=1}^{nt} (H_k \cos(2\pi k \Delta f t - \varphi_k)) \right] \cdot \frac{2}{N} \quad (10)$$

In Figure 7, we see the square wave signal (one cycle of a square wave that ranges between 0 and 1 with equal times high and low) to be transformed as well as the signal constructed from the first 10 terms of a Fourier series using the coefficients from the FFT as per Equation 8. We would obtain an identical waveform if we took the IFFT of a truncation of the original FFT, where all the FFT’s coefficients

with an index greater than the number of desired terms (here, $nt = 10$) are set to zero.

Windows

Windows are useful for extracting and/or smoothing data. A window is typically a positive, smooth symmetric function that has a value of one at its maximum and approaches zero at the extremes. (A window might have a discontinuity in its first derivative, giving it an inverted V shape—such a window is sometimes referred to as a “tent”—or two discontinuities for a rectangular or trapezoidal shape.) We apply windows by multiplying time-domain data by the window function. Of course, whenever a window is applied, it alters at least some of the data.

Smoothing windows, for example, reduce the amplitude of the time-domain data at both the beginning and the end of the windowed data set. One effect of this smoothing is to reduce leakage in the frequency domain. In Figure 8, we show comparative plots of four frequently used windows. We show the effect of applying three of those windows to a sine wave sequence in Figure 9.

Let’s look at the expressions for four common windows:

$$\begin{aligned} \text{Rectangular:} \quad & \text{recw}_i = \begin{cases} 1 & (\text{inside}) \\ 0 & (\text{outside}) \end{cases} \\ \text{Hamming:} \quad & \text{hamw}_i = 0.54 - 0.46 \cdot \cos(2 \cdot \pi \cdot i/N) \\ \text{Hann:} \quad & \text{hanw}_i = 0.5 - 0.5 \cdot \cos(2 \cdot \pi \cdot i/N) \\ \text{Blackman:} \quad & \text{blkw}_i = 0.42 - 0.5 \cdot \cos(2 \cdot \pi \cdot i/N) + 0.08 \cdot \cos(4 \cdot \pi \cdot i/N). \end{aligned} \quad (11)$$

The Hamming and Hann windows differ in only one parameter: if the corresponding coefficients are written $\alpha - (1 - \alpha)$, then α is 0.54 for the Hamming window and 0.5 for the Hann. The fact that a slight change in the parameter value gives rise to two different windows hints at the sensitivity of the windowing process to the value of α . If α decreases from 0.5, the side lobes increase significantly in amplitude. As α increases from 0.5 to 0.54, the relative sizes of the side lobes change. The first set of the Hann side lobes tend to be significantly larger than those of the Hamming case, but subsequent Hann side lobes decrease rapidly in magnitude and become significantly smaller than the Hamming side lobes. As a general appearance, the Hamming window doesn’t quite go to zero at the window’s endpoints whereas the Rectangular, Hann, and Blackman windows do. Several other windows also exist, including Bartlett (tent function), Welch (parabolic), Parzen (piece-wise cubic), Lanczos (central lobe

of a sine function), Gaussian, and Kaiser (which uses a modified Bessel function).

Each of these windows has particular characteristics. Two particularly useful points of comparison in the frequency space are the full width at half maximum of the central peak and the relative magnitude of central peak to that of the side lobes. An unwindowed signal's FFT has the narrowest central peak, but it also has considerable leakage that decays slowly. The curves for the Hamming and Blackman cases show wider central peaks but significantly smaller side lobes. The Blackman window has the largest peak height to first sidelobe height ratio.

There is no final summary statement that says you should use window x in all cases—circumstances decide that. In the bat-chirp analysis we'll examine in part two of this series, we'll use an isosceles trapezoidal window. Such a window isn't generally recommended, but for the bat-chirp case, it's the best choice. (A split cosine bell curve, a Hann window shape for the beginning and end of the curve with a magnitude of one in the interior, would give essentially the same results.)

As an example of windowing's effect on the transform, we apply a Blackman window to the time-domain data associated with Figure 1b. Two effects of applying this window, as Figure 10 shows, are that the leakage is greatly reduced and that the central peak is broadened. Obtaining the needed detail to observe these features requires zero padding.

In part two of this series, we'll discuss auto-regression spectral analysis and the maximum entropy method, convolution, and filtering. In the third and final installment, we'll present some applications, including the analysis of a bat chirp and atmospheric sea-level pressure variations in the Pacific Ocean.

Whether there is an interest in CO₂ concentrations in the atmosphere, ozone levels, sunspot numbers, variable star magnitudes, the price of pork, or financial markets, or if the interest is in filtering, correlations, or convolutions, Fourier transforms provide a very powerful and, for many, an essential algorithmic tool.

References

1. R.N. Bracewell, *The Fourier Transform and Its Applications*, McGraw-Hill, 1965.
2. E.O. Brigham, *The Fast Fourier Transform and Its Applications*, Prentice-Hall, 1988.
3. J.F. James, *A Student's Guide to Fourier Transforms*, Cambridge Univ. Press, 1995.
4. C.T. Chen, *Digital Signal Processing*, Oxford Univ. Press, 2001.
5. S.L. Marple Jr., *Digital Spectral Analysis with Applications*, Prentice Hall, 1987.
6. S. Smith, *Digital Signal Processing*, Newnes, 2003.
7. P. Bloomfield, *Fourier Analysis of Time Series*, John Wiley & Sons, 2000.
8. P. Hertz and E.D. Feigelson, "A Sample of Astronomical Time Series," *Applications of Time Series Analysis in Astronomy and Meteorology*, T. Subba Rao, M.B. Priestley, and O. Lessi, eds., Chapman & Hall, 1979, pp. 340–356.
9. W.H. Press et al., *Numerical Recipes in Fortran*, Cambridge Univ. Press, 1992.
10. L.N. Trefethen, *Spectral Methods in Matlab*, SIAM Press, 2000.

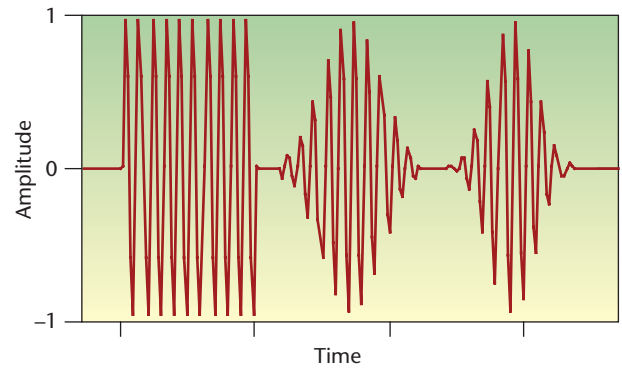


Figure 9. A comparison of the effects (from left to right) of a rectangular, a Hamming, and a Blackman window on a sine wave sequence. For convenience of display, we compute the three examples separately, shift the second and third in time, and sum the set, with the effect that the three examples appear sequentially in time; because each example is zero outside its window zone, the results do not interfere. The three windows have the same width, but as Figure 8 shows, the Blackman window increases in magnitude more slowly than the others, and we can observe the effect on the sine wave signal. The difference between Hamming and Blackman windowing is also evident.

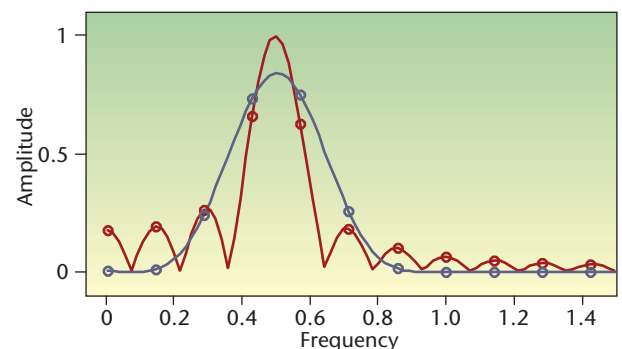


Figure 10. The effects of windowing as seen in the transform space. The FFT of the 3.52-cycle example in Figure 1 and the result of multiplying time-domain data and a Blackman window before taking the FFT are shown without zero padding (circles) and with zero padding (solid curves). The windowed form reduces leakage but has a broader central lobe.

11. C.D. Meyer, *Matrix Analysis and Applied Linear Algebra*, SIAM Press, 2000.
12. J.W. Cooley and J.W. Tukey, "An Algorithm for the Machine Calculation of Complex Fourier Series," *Mathematics of Computation*, vol. 19, no. 90, 1965, pp. 297-301.
13. D.N. Rockmore, "The FFT: An Algorithm the Whole Family Can Use," *Computing in Science & Eng.*, vol. 2, no. 1, 2000, pp. 60-64.

Denis Donnelly is a professor of physics at Siena College. His research interests include computer modeling and electronics. Donnelly received a PhD in physics from the University of Michigan. He is a member of the American Physical Society, the American Association of Physics Teachers, and the American Association for the Advancement of Science. Contact him at donnelly@siena.edu.

Bert Rust is a mathematician at the National Institute for Standards and Technology. His research interests include ill-posed problems, time-series modeling, nonlinear regression, and observational cosmology. Rust received a PhD in astronomy from the University of Illinois. He is a member of SIAM and the American Astronomical Society. Contact him at bwr@nist.gov.

Submissions: Send one PDF copy of articles and/or proposals to Norman Chonacky, Editor in Chief, cise-editor@aip.org. Submissions should not exceed 6,000 words and 15 references. All submissions are subject to editing for clarity, style, and space.

Editorial: Unless otherwise stated, bylined articles and departments, as well as product and service descriptions, reflect the author's or firm's opinion. Inclusion in *CISE* does not necessarily constitute endorsement by the IEEE, the AIP, or the IEEE Computer Society.

Circulation: *Computing in Science & Engineering* (ISSN 1521-9615) is published bimonthly by the AIP and the IEEE Computer Society. IEEE Headquarters, Three Park Ave., 17th Floor, New York, NY 10016-5997; IEEE Computer Society Publications Office, 10662 Los Vaqueros Circle, PO Box 3014, Los Alamitos, CA 90720-1314, phone +1 714 821 8380; IEEE Computer Society Headquarters, 1730 Massachusetts Ave. NW, Washington, DC 20036-1903; AIP Circulation and Fulfillment Department, 1NO1, 2 Huntington Quadrangle, Melville, NY 11747-4502. Annual subscription rates for 2005: \$42 for Computer Society members (print only) and \$42 for AIP society members (print plus online). For more information on other subscription prices, see www.computer.org/subscribe/ or https://www.aip.org/forms/journal_catalog/order_form_fs.html. Computer Society back issues cost \$20 for members, \$96 for nonmembers; AIP back issues cost \$22 for members.

Postmaster: Send undelivered copies and address changes to *Computing in Science & Engineering*, 445 Hoes Ln., Piscataway, NJ 08855. Periodicals postage paid at New York, NY, and at additional mailing offices. Canadian GST #125634188. Canada Post Corporation (Canadian distribution) publications mail agreement number 40013885. Return undeliverable Canadian addresses to PO Box 122, Niagara Falls, ON L2E 6S8 Canada. Printed in the USA.

Copyright & reprint permission: Abstracting is permitted with credit to the source. Libraries are permitted to photocopy beyond the limits of US copyright law for private use of patrons those articles that carry a code at the bottom of the first page, provided the per-copy fee indicated in the code is paid through the Copyright Clearance Center, 222 Rosewood Dr., Danvers, MA 01923. For other copying, reprint, or republication permission, write to Copyright and Permissions Dept., IEEE Publications Administration, 445 Hoes Ln., PO Box 1331, Piscataway, NJ 08855-1331. Copyright © 2005 by the Institute of Electrical and Electronics Engineers Inc. All rights reserved.

ADVERTISER / PRODUCT INDEX MARCH/APRIL 2005

Advertiser	Page Number
DE Shaw & Company	7
John Wiley	Cover 2
Nanotech 2005	Cover 4
North Carolina Central University	4

Boldface denotes advertisements in this issue.

Advertising Personnel

Marion Delaney IEEE Media, Advertising Director Phone: +1 212 419 7766 Fax: +1 212 419 7589 Email: md.ieeemedia@ieee.org	Marian Anderson Advertising Coordinator Phone: +1 714 821 8380 Fax: +1 714 821 4010 Email: manderson@computer.org	Sandy Brown IEEE Computer Society, Business Development Manager Phone: +1 714 821 8380 Fax: +1 714 821 4010 Email: sb.ieeemedia@ieee.org
---	---	--

Advertising Sales Representatives

Mid Atlantic (product/recruitment) Dawn Becker Phone: +1 732 772 0160 Fax: +1 732 772 0161 Email: db.ieeemedia@ieee.org	Midwest (product) Dave Jones Phone: +1 708 442 5633 Fax: +1 708 442 7620 Email: dj.ieeemedia@ieee.org	Northwest (product) Peter D. Scott Phone: +1 415 421-7950 Fax: +1 415 398-4156 Email: peterd@pscottassoc.com
New England (product) Jody Estabrook Phone: +1 978 244 0192 Fax: +1 978 244 0103 Email: je.ieeemedia@ieee.org	Will Hamilton Phone: +1 269 381 2156 Fax: +1 269 381 2556 Email: wh.ieeemedia@ieee.org	Southern CA (product) Marshall Rubin Phone: +1 818 888 2407 Fax: +1 818 888 4907 Email: mr.ieeemedia@ieee.org
New England (recruitment) Robert Zwick Phone: +1 212 419 7765 Fax: +1 212 419 7570 Email: r.zwick@ieee.org	Joe DiNardo Phone: +1 440 248 2456 Fax: +1 440 248 2594 Email: jd.ieeemedia@ieee.org	Northwest/Southern CA (recruitment) Tim Matteson Phone: +1 310 836 4064 Fax: +1 310 836 4067 Email: tm.ieeemedia@ieee.org
Connecticut (product) Stan Greenfield Phone: +1 203 938 2418 Fax: +1 203 938 3211 Email: greenco@optonline.net	Southeast (product) Bob Doran Phone: +1 770 587 9421 Fax: +1 770 587 9501 Email: bd.ieeemedia@ieee.org	Japan (product/recruitment) Tim Matteson Phone: +1 310 836 4064 Fax: +1 310 836 4067 Email: tm.ieeemedia@ieee.org
Midwest/Southwest (recruitment) Darcy Giovingo Phone: +1 847 498-4520 Fax: +1 847 498-5911 Email: dg.ieeemedia@ieee.org	Southeast (recruitment) Thomas M. Flynn Phone: +1 770 645 2944 Fax: +1 770 993 4423 Email: flynttom@mindspring.com	Europe (product/recruitment) Hilary Turnbull Phone: +44 1875 825700 Fax: +44 1875 825701 Email: impress@impressmedia.com
	Southwest (product) Josh Mayer Phone: +1 972 423 5507 Fax: +1 972 423 6858 Email: josh.mayer@wagen eckassociates.com	