Data Processing and Analysis

Rick Aster and Brian Borchers

September 27, 2013

Sampled Time Series

Numerical scientific data are commonly organized into series or matrices, i.e., sets of spatially or temporally ordered numbers that approximate a continuous time function (e.g., seismic signals, magnetic observatory data, temperature variations). In spatial applications, the data commonly consists of a two- or three-dimensional array of samples (e.g., gravity, magnetic, or structural surveys). These data may be irregularly sampled in space and/or time. Here, we will consider Fourier theory appropriate to the case where the data are sampled at regular intervals (or where irregularly sampled data has been interpolated or otherwise transferred to a regular array of numbers).

Beginning with a continuou function, multiplication by the (uniformly spaced delta function sequence) shah function, $\mathrm{III}(t)$, can be conceptualized as performing a *regular sampling* operation for a time series. By "regular", we mean that this operation selects out instantaneous functional values at equally-spaced intervals, $1/r$ (where r is the *sampling rate* or *sampling frequency*), and ignores continuous function information between the samples. In instrumentation practice, this type of operation is in practice performed by an *analog-to-digital converter (A to D)* or *digitizer*, and the sampled values are stored as series or arrays of numbers.

To examine what sampling does to the spectral characteristics of an arbitrary function, we evaluate the Fourier transform of $\mathrm{III}(t)$ and apply the frequency-domain counterpart of the convolution theorem. We will find $\mathcal{F}[\mathrm{III}(t)]$ by evaluating the Fourier transform of a function with a limit that converges to $\mathrm{III}(t)$. One such function is

$$\mathrm{III}(t) = \sum_{n=-\infty}^{\infty} \delta(t-n) = \lim_{\tau\to 0} \frac{1}{\tau} e^{-\pi\tau^2 t^2} \sum_{n=-\infty}^{\infty} e^{-\pi(t-n)^2/\tau^2} \ . \qquad (1)$$

Note that (1) consists of a broad Gaussian envelope

$$e^{-\pi\tau^2 t^2} \qquad (2)$$

multiplied by a periodic component

$$\frac{1}{\tau} \sum_{n=-\infty}^{\infty} e^{-\pi(t-n)^2/\tau^2} \qquad (3)$$

You may already know that a smooth periodic function has a *Fourier series*, which is a line spectrum consisting of equally spaced delta functions (some of which may have zero amplitude so as to leave holes in the spectrum). The Fourier series for (3) is

$$\frac{1}{\tau}\sum_{n=-\infty}^{\infty} e^{-\pi(t-n)^2/\tau^2} = \sum_{n=-\infty}^{\infty} e^{-\pi\tau^2 n^2} e^{\imath 2\pi nt} \tag{4}$$

so that

$$\mathrm{III}(t) = \lim_{\tau\to 0} e^{-\pi\tau^2 t^2} \sum_{n=-\infty}^{\infty} e^{-\pi\tau^2 n^2} e^{\imath 2\pi nt} \ . \tag{5}$$

Thus,

$$\mathcal{F}[\mathrm{III}(t)] = \lim_{\tau\to 0} \sum_{n=-\infty}^{\infty} e^{-\pi\tau^2 n^2} \mathcal{F}[e^{-\pi\tau^2 t^2} e^{\imath 2\pi nt}] \tag{6}$$

and applying the frequency-domain counterpart of the time shift theorem gives

$$\mathcal{F}[\mathrm{III}(t)] = \lim_{\tau\to 0} \sum_{n=-\infty}^{\infty} e^{-\pi\tau^2 n^2} \mathcal{F}[e^{-\pi\tau^2 t^2}]|_{f=f-n} \ . \tag{7}$$

The Fourier transform of a Gaussian function is

$$\mathcal{F}[e^{-\alpha\pi t^2}] = \int_{-\infty}^{\infty} e^{-\alpha\pi t^2 - 2\pi\imath ft}\ dt \tag{8}$$

$$= e^{-\pi f^2/\alpha} \int_{-\infty}^{\infty} e^{-\pi(\alpha t^2 + 2\imath ft - f^2/\alpha)}\ dt = e^{-\pi f^2/\alpha} \int_{-\infty}^{\infty} e^{-\pi(\alpha^{1/2} t + \imath f/\alpha^{1/2})^2}\ dt \ . \tag{9}$$

Substituting $\xi = \alpha^{1/2} t + \imath f/\alpha^{1/2}$ gives

$$= \frac{1}{\alpha^{1/2}} e^{-\pi f^2/\alpha} \int_{-\infty}^{\infty} e^{-\pi\xi^2} d\xi = \frac{1}{\alpha^{1/2}} e^{-\pi f^2/\alpha} \ . \tag{10}$$

So the Fourier transform of a Gaussian is just another Gaussian! Thus, we have

$$\mathcal{F}[\mathrm{III}(t)] = \lim_{\tau\to 0} \frac{1}{\tau} \sum_{n=-\infty}^{\infty} e^{-\pi\tau^2 n^2} e^{-\pi(f-n)^2/\tau^2} \ . \tag{11}$$

Now we take the limit as $\tau \to 0$ and see that (11) converges to the same limit as (1); the shah is, like the Gaussian, its own Fourier transform

$$\mathcal{F}[\mathrm{III}(t)] = \mathrm{III}(f) \ . \tag{12}$$

Sampling and Aliasing. Consider a sampled time function

$$\psi(t) = \phi(t) \cdot r\mathrm{III}(rt) \tag{13}$$

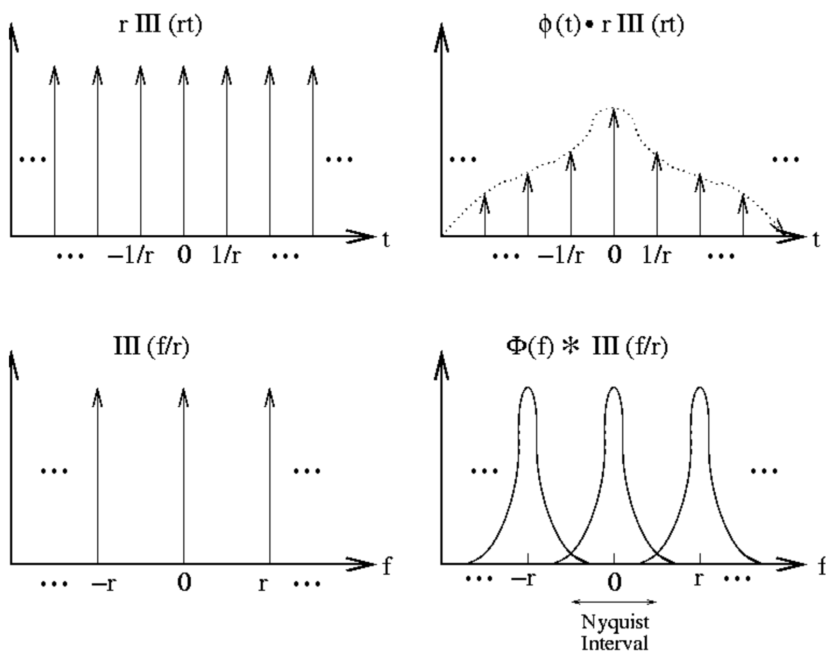

Figure 1: The Shah function and its Fourier Transform; Fourier Transform of a Sampled Function (slightly aliased)

which is a regularly spaced (intervals of r^{-1}) sequence of delta functions in time with areas given by the values of $\phi(t)$ at those times. Using the convolution and scaling theorems, we can see that (12) gives (Figure 1)

$$\Psi(f) = \Phi(f) * \text{III}(\frac{f}{r}) \ . \tag{14}$$

Sampling thus simply replicates the Fourier transform of $\phi(t)$, $\Phi(f)$, along the frequency axis at $\pm nr$. These copies are referred to as *aliases.* If $\Phi(f)$ is *band-limited* to having its energy in the frequency interval $(-f_{max}, f_{max})$ and if $f_{max} \leq r/2$, then these aliases will not overlap. This is a crucial observation; it implies that $\phi(t)$ is fully recoverable from the sampled series via an inverse Fourier transform across one of the aliases

$$\phi(t) = F^{-1}[\Psi(f)\Pi(f/r)] \tag{15}$$

or (using the convolution theorem) as the convolution

$$\phi(t) = \psi(t) * r \ \text{sinc}(rt) \ . \tag{16}$$

This remarkable result, that a continuous band-limited function can be fully recovered from time series sampled at a rate of $r > 2f_{max}$ (so that the aliases don't overlap!), leads to the definition of the *Nyquist frequency*

$$f_N = 2f_{max} \tag{17}$$

the minimum frequency at which we must sample for information to be recovered without corruption from a sampled time series. Thus, if we wish to sample a signal that has appreciable power up to 100 Hz, we must sample using a rate of at least $f_N = 200$ Hz. One way of intuitively appreciating the Nyquist frequency concept is that it takes slightly more than two samples per period to accurately characterize a sinusoid.

As can be seen from (14) that, if the sampling rate r is less than $2f_{max} = f_N$, (as in Figure 1), then the sampled times series aliases will overlap and corrupt each other, a condition called *undersampling.* Applying (16) to try and recover $\phi(t)$ in this case will produce a distorted recovered function. This undersampling distortion is called *aliasing*, and such a time series is referred to as being *aliased.* If we aren't interested in the higher frequency content in a signal, we can eliminate aliasing problems by removing the higher-frequencies from the data (using low-pass filtering) prior to sampling so that the signal contains a negligible amount of energy at frequencies near and above $f_N/2$. This type of presampling, low-pass filter is called an *antialias filter.* In data acquisition systems, antialiasing is sometimes practically accomplished by drastically oversampling the data at the analog input and then filtering and decimating the signal digitally to produce an unaliased signal at a lower, desired sampling rate. This eliminates the need for variable analog antialiasing electronics for the lower sampling rates.

It is important to understand in detail what happens if we undersample data. First, note that we never satisfy (17) exactly, because all real data sets are time or space limited and thus can never be truly band-limited to $\pm r/2$ (fortunately, we can get close in this regard in practical cases). One way to see this is to note that ”perfect” low-pass filtering is unobtainable, as the impulse response of a perfect low-pass filter (one with a frequency response of $\Pi(f/f_{max})$) is the acausal sinc function, which has non-zero values from $t = -\infty$ to $t = \infty$. Consider the distorted spectrum, $\Phi_a(f)$, resulting from the influence of the two nearest frequency-domain aliases, which are centered at $f = \pm r$ (Figure 1)

$$\Phi_a(f) = \Phi(f) + \Phi(f - r) + \Phi(f + r) \ . \tag{18}$$

If $\phi(t)$ is real, then $\Phi(f)$ is Hermitian, so that

$$\Phi_a(f) = \Phi(f) + \Phi^*(r - f) + \Phi^*(r + f) \ . \tag{19}$$

The contribution to the aliased signal from the second two terms is just what one would get by adding complex-conjugated versions of the spectrum which have been “folded” in the frequency domain at $f = \pm r/2$. Note that the actual character of corruption of the original signal depends on the specific characteristics of $\Phi(f)$. The greatest distortion will occur if there is sufficient high-frequency energy above $f = r/2$ so that even the lower frequency components of $\Phi_a(f)$ (19) will be significantly different than those of $\Phi(f)$. A time domain sign of danger in a sampled data set would be the occurrence of lots of terms with alternating signs, as this is an indication that there is significant energy at or above $f = r/2$.

As an example of aliasing which could occur in practice, consider an undersampled voltage that is contaminated by an $f_0 = 60$ Hz AC sinusoidal noise

$$n(t) = A\cos(2\pi \cdot 60t) \ . \tag{20}$$

To prevent aliasing of $n(t)$, we would have to sample at a rate greater than $r \geq f_N = 2f_0 = 120$ Hz. If we instead sampled at a lesser rate, the delta function spectrum of the noise component

$$n_a(t) = n(t) \cdot r\mathrm{III}(rt) \tag{21}$$

would have, in the central alias bracketing $f = 0$, its frequencies mapped to $f = \pm(r - 60)$ Hz. As an extreme case, if we sampled at half of the Nyquist frequency, (60 Hz), the 60 Hz energy in $n(t)$ would be mapped to zero frequency – producing a zero frequency component in the retrieved function. We can see why this is by looking back in the time domain and noting that this corresponds to sampling a sinusoid once per period, so that all such samples will have identical value. The specific value would depend on the phase relationship between the sampling function and $n(t)$; if the samples are centered on zero time and $n(t)$ is a cosine, then we would recover a maximum zero-frequency signal of amplitude A. Aliasing thus puts true signal into different frequency ranges. This behavior occurs because, for signal frequencies higher than the Nyquist frequency, sampling and recovery is a nonlinear process.

Fourier Theory in Discrete Time. In analyzing sampled time series, it is more practical to work in discrete (rather than continuous) time or space. As previously mentioned, essentially all practical data analysis schemes are implemented on computers, which do not process functions per se, but instead operate on discrete ordered sets of numbers. A 1-dimensional ordered set of numbers is called a *sequence* , which we will typically represent in subscript notation

$$x_n(n \in \text{integers}) \ . \tag{22}$$

The discrete time equivalent of the delta and step functions are the *Kronecker delta*

$$\delta_{n-m} \equiv \delta_{n,m} = \begin{cases} 1 \ (n = m) \\ 0 \ (n \neq m) \end{cases} \tag{23}$$

and its associated discrete step function

$$\mathrm{H_{n-m}} = \begin{cases} 1 \ (n \geq m) \\ 0 \ (n < m) \end{cases} \tag{24}$$

In the discrete time domain, summation supplants integration, so that the delta/step relationship integral relationship in continuous time becomes

$$H_{n-l} = \sum_{k=-\infty}^{n} \delta_{k-l} \ . \tag{25}$$

Analogously, convolution in the discrete world (e.g., in MATLAB) is a summation operation

$$x_n * y_n = \sum_{k=-\infty}^{\infty} x_k y_{n-k} \tag{26}$$

where the y index is reversed in the summation index, k, which fills in for its continuous counterpart, τ.

To investigate how Fourier concepts apply to sequences, consider the response of a linear discrete-time system (with an infinite-length impulse response sequence x_n) to a unit-amplitude, complex sinusoidal signal, s_n:

$$g_n = \sum_{k=-\infty}^{\infty} x_k s_{n-k} = \sum_{k=-\infty}^{\infty} x_k e^{2\pi\imath f(n-k)} \tag{27}$$

$$= e^{\imath 2\pi f n} \sum_{k=-\infty}^{\infty} x_k e^{-\imath 2\pi f k} \equiv X(f) e^{\imath 2\pi f n} \tag{28}$$

where $X(f)$ is the Fourier transform of x_n (keep in mind that x_n is a sequence, not a continuous function). We can unify the Fourier transform definitions for continuous and discrete functions using the sifting property of the delta function

$$X(f) \equiv \mathcal{F}[x_n] = \mathcal{F}[r\mathrm{III}(rt)x(t)] = r\mathcal{F}[\sum_{n=-\infty}^{\infty} \delta(rt-n)x(t)] \tag{29}$$

$$= r \int_{-\infty}^{\infty} \sum_{n=-\infty}^{\infty} \delta(rt - n) x(t) e^{-\imath 2\pi f t} dt = r \sum_{n=-\infty}^{\infty} x(n/r) e^{-\imath 2\pi f n/r} \ . \tag{30}$$

The spectrum of (30) is continuous and periodic in the frequency domain, (with a spectral period of r). This periodicity reflects the spectral aliasing effects of sampling discussed earlier. It is usually most convenient to take $r = 1$, in which case the spectrum is normalized with respect to the Nyquist frequency and we need only concern ourselves with a unit Nyquist interval $-1/2 \leq f \leq 1/2$ to capture all of the information in x_n (provided that we sample rapidly enough so that the spectral aliases are non-overlapping)

$$X(f) = \int_{-\infty}^{\infty} \sum_{n=-\infty}^{\infty} \delta(t - n) x(t) e^{-\imath 2\pi f t} dt = \sum_{n=-\infty}^{\infty} x(n) e^{-\imath 2\pi f n} \ . \tag{31}$$

The original sequence can be recovered using the inverse Fourier transform, where we restrict the range of integration to the Nyquist interval

$$x_n = \int_{-1/2}^{1/2} X(f) e^{\imath 2\pi f n} \ df \ . \tag{32}$$

The Discrete Fourier Transform. (31) and (32) form a transform pair, but not a very useful or symmetric one, as the time sequence is infinite and the spectrum is continuous. You might imagine (and you would be right), that $X(f)$, being band limited to the Nyquist interval, could be completely specified by some sequence in the frequency domain. In this case, we would have a transform pair where both the time and frequency domain representations are discrete, and that could be used in practical situations to analyze data.

To construct such a transform pair, consider a periodic sequence, x_n, where the period is N samples. For the moment, assume that the sequence is sampled at a sampling rate of $r = 1$– we will discuss other sampling rates later. Because of its periodicity, every component of x_n of the form $e^{2\pi ikn/N}$ must also be N-sample periodic. These periodic components must therefore have frequencies $f = k/N$, where k is some integer. Because our sequence is sampled at rate $r = 1$, frequencies outside of the range $0 \leq r \leq 1$ would be aliased. Thus it's unnecessary to include frequencies k/N for k outside of the range from 0 to $N - 1$.

The sequence can thus be completely characterized across one of its periods via the expansion

$$x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k e^{2\pi ikn/N} = \text{IDFT}(X_k) \ . \tag{33}$$

The normalization factor $1/N$ is not strictly required, but is included at this point to conform with standard conventions. Equation (33) defines our inverse discrete Fourier transform. The corresponding forward transform is

$$X_k = \sum_{n=0}^{N-1} x_n e^{-2\pi ikn/N} = \text{DFT}(x_n) \ . \tag{34}$$

To verify this transform pair, we can begin with (33) and apply the forward transform to both sides of the equation

$$\sum_{n=0}^{N-1} x_n e^{-i2\pi nm/N} = \frac{1}{N} \sum_{n=0}^{N-1} \sum_{k=0}^{N-1} X_k e^{\imath 2\pi kn/N} e^{-\imath 2\pi nm/N} \ . \tag{35}$$

Interchanging the order of summation gives

$$\sum_{n=0}^{N-1} x_n e^{-i2\pi nm/N} = \frac{1}{N} \sum_{k=0}^{N-1} X_k \sum_{n=0}^{N-1} e^{\imath 2\pi n(k-m)/N} \ . \tag{36}$$

$$\sum_{n=0}^{N-1} x_n e^{-i2\pi nm/N} = \frac{1}{N} \sum_{k=0}^{N-1} X_k \sum_{n=0}^{N-1} \left(e^{\imath 2\pi (k-m)/N}\right)^n \ . \tag{37}$$

Now, consider the innermost sum

$$\sum_{n=0}^{N-1} \left(e^{\imath 2\pi (k-m)/N}\right)^n \ . \tag{38}$$

Recall that the sum of a finite geometric series is given by

$$1 + r + r^2 + \ldots + r^{N-1} = \frac{1 - r^N}{1 - r} \quad r \neq 1 \ . \tag{39}$$

When $r = 1$, the sum is simply N. When $k - m$ is a multiple of N, then

$$e^{\imath 2\pi (k-m)/N} = 1 \tag{40}$$

and

$$\sum_{n=0}^{N-1} \left(e^{\imath 2\pi (k-m)/N}\right)^n = \sum_{n=0}^{N-1} 1^n = N \ . \tag{41}$$

When (the integer) $k - m$ is not a multiple of N, $e^{\imath 2\pi (k-m)/N}$ is not equal to one, and

$$\sum_{n=0}^{N-1} \left(e^{\imath 2\pi (k-m)/N}\right)^n = \frac{1 - \left(e^{\imath 2\pi (k-m)/N}\right)^N}{1 - e^{\imath 2\pi (k-m)/N}} \ . \tag{42}$$

But

$$\left(e^{\imath 2\pi (k-m)/N}\right)^N = e^{\imath 2\pi (k-m)} = 1 \tag{43}$$

so,

$$\sum_{n=0}^{N-1} \left(e^{\imath 2\pi (k-m)/N}\right)^n = \frac{1 - 1}{1 - e^{\imath 2\pi (k-m)/N}} = 0 \ . \tag{44}$$

Thus

$$\sum_{n=0}^{N-1} \left(e^{\imath 2\pi (k-m)/N}\right)^n = \begin{cases} N & (k-m) \text{ is a multiple of } N \\ 0 & \text{otherwise} \end{cases} \tag{45}$$

We are only interested in integer values of k and m between 0 and $N-1$. Thus $k-m$ will only be a multiple of N when $k-m=0$, and

$$\sum_{n=0}^{N-1} \left(e^{\imath 2\pi(k-m)/N}\right)^n = N\delta_{k,m} \ . \tag{46}$$

Returning to our original sum, and using the above result,

$$\sum_{n=0}^{N-1} x_n e^{-i2\pi nm/N} = \frac{1}{N}\sum_{k=0}^{N-1} X_k N\delta_{k,m} \tag{47}$$

which reduces to

$$\sum_{n=0}^{N-1} x_n e^{-i2\pi nm/N} = X_m \ . \tag{48}$$

This derivation shows that $\mathrm{DFT}(\mathrm{IDFT}(X_k)) = X_k$. Similarly, it can now be easily confirmed that $\mathrm{IDFT}(\mathrm{DFT}(x_n)) = x_n$. An example DFT/IDFT pair is shown in Figure 3.

It is easy to see that the Discrete Fourier Transform of an N-periodic sequence also produces an N-periodic sequence

$$X_{m+N} = \sum_{n=0}^{N-1} x_n e^{-i2\pi n(m+N)/N} \tag{49}$$

or

$$X_{m+N} = e^{-i2\pi N/N} \sum_{n=0}^{N-1} x_n e^{-i2\pi nm/N} \ . \tag{50}$$

Since $e^{-i2\pi} = 1$,

$$X_{m+N} = \sum_{n=0}^{N-1} x_n e^{-i2\pi nm/N} = X_m. \tag{51}$$

The $k=0$ term of the DFT is just N times the average value of x_n, while the N-periodicity of the DFT implies that

$$X_{N-k} = \sum_{n=0}^{N-1} x_n e^{-\imath 2\pi n(N-k)/N} = \sum_{n=0}^{N-1} x_n e^{\imath 2\pi nk/N} = X_{-k} \ . \tag{52}$$

Thus, the N periodicity here implies that the upper portion of the (N even) DFT sequence, $N/2 \le k \le N-1$, contains negative frequency spectral information, corresponding to $-N/2 \le k \le -1$ (Figure 4), while the lower portion contains positive frequency spectral information. If we wish to display an N-point DFT spectral sequence centered on the zero-frequency component (as we are used to picturing continuous Fourier transforms) we must therefore plot the DFT for $-N/2 \le k \le (N/2)-1$ (or $-(N-1)/2 \le k \le (N-1)/2$ for N odd) rather than

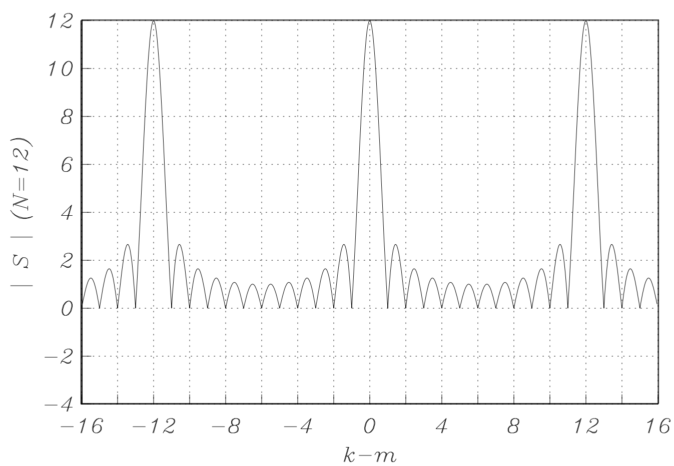

Figure 2: Amplitude of (38) as a function of $k - m$ for $N = 12$.

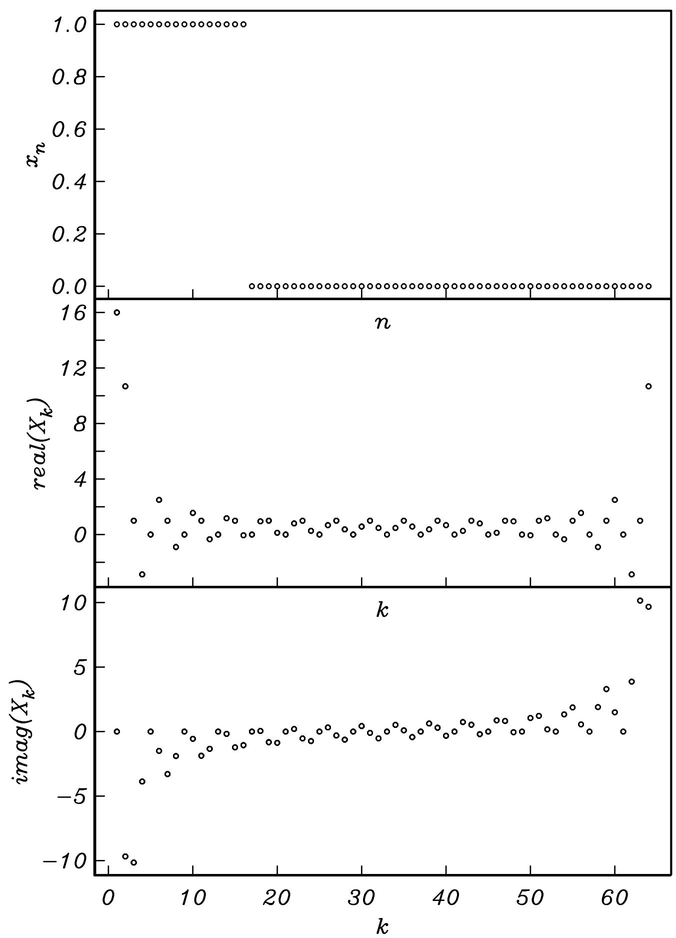

Figure 3: An example ($N = 64$) DFT.

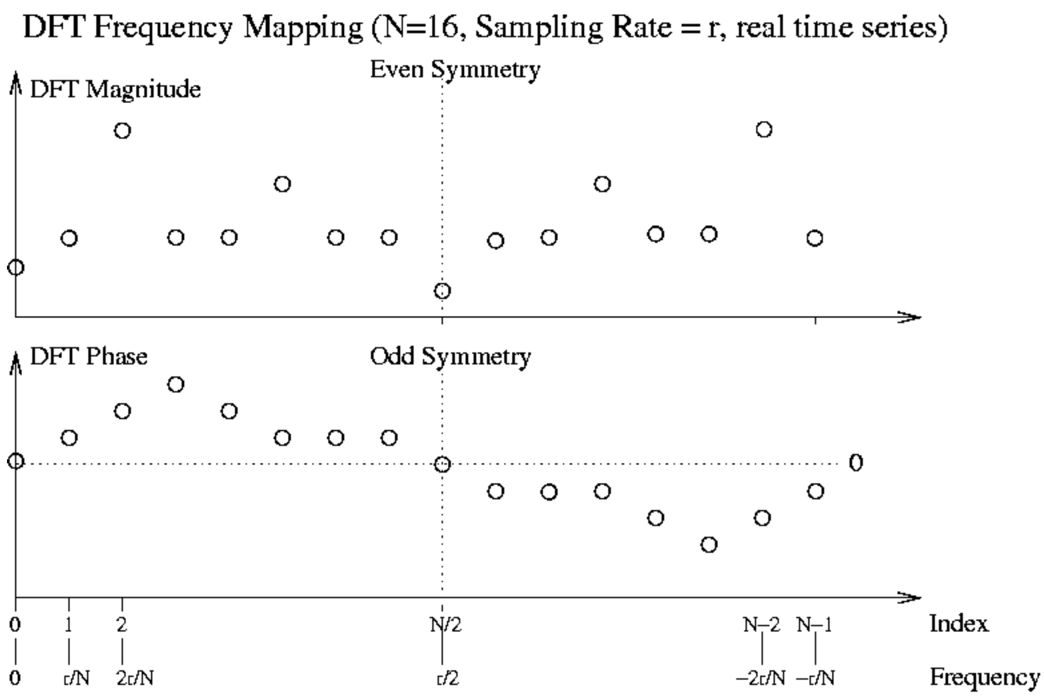

Figure 4: DFT frequency-index mapping.

$0 \le k \le N-1$, taking into account the above mapping. In MATLAB, there is an *fftshift* command that performs this rearrangement of the DFT coefficients.

Formulas for the DFT and its inverse can be written more compactly in terms of

$$w_N = e^{\imath 2\pi/N}. \tag{53}$$

The DFT can be written as

$$X_m = \sum_{n=0}^{N-1} x_n w_N^{-mn}. \tag{54}$$

The inverse DFT becomes

$$x_n = \frac{1}{N}\sum_{k=0}^{N-1} X_k w_N^{kn}. \tag{55}$$

The DFT and its inverse can also be written in matrix form as

$$\begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_{N-1} \end{bmatrix} = \frac{1}{N}\begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & w_N & w_N^2 & \cdots & w_N^{N-1} \\ 1 & w_N^2 & w_N^4 & \cdots & w_N^{2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & w_N^{N-1} & w_N^{2(N-1)} & \cdots & w_N^{(N-1)(N-1)} \end{bmatrix}\begin{bmatrix} X_0 \\ X_1 \\ X_2 \\ \vdots \\ X_{N-1} \end{bmatrix} \tag{56}$$

and

$$\begin{bmatrix} X_0 \\ X_1 \\ X_2 \\ \vdots \\ X_{N-1} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & w_N^{-1} & w_N^{-2} & \cdots & w_N^{-(N-1)} \\ 1 & w_N^{-2} & w_N^{-4} & \cdots & w_N^{-2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & w_N^{-(N-1)} & w_N^{-2(N-1)} & \cdots & w_N^{-(N-1)(N-1)} \end{bmatrix}\begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_{N-1} \end{bmatrix} \tag{57}$$

In the language of linear algebra, this shows that the DFT is a change of basis formula. It's also easy to show that the DFT basis is an orthogonal basis.

A summary of the discrete and continuous Fourier transform pairs defined here is given in Table 1. Here, (C , D) denote continuous or discrete and (P , A) denote periodic or aperiodic. Continuous periodic functions are assumed periodic on the unit interval and discrete periodic functions have period N.

There are many results for the DFT that are analogous to results for the continuous Fourier transform. For example, the time shift theorem for the DFT is

$$\text{DFT}[x_{n-n_0}] = \sum_{n=0}^{N-1} x_{n-n_0} e^{-\imath 2\pi kn/N} \tag{58}$$

$$\text{DFT}[x_{n-n_0}] = \sum_{l=-n_0}^{N-n_0-1} x_l e^{-\imath 2\pi k(l+n_0)/N} \tag{59}$$

$\phi(t)$	$\Phi(f)$	Transform	Forward Transform	Inverse Transform
C, A	C, A	Fourier Transform	$\Phi(f) = \int_{-\infty}^{\infty} \phi(t) e^{-\imath 2\pi f t} dt$	$\phi(t) = \int_{-\infty}^{\infty} \Phi(f) e^{\imath 2\pi f t} df$
C, P	D, A	Fourier Series	$\Phi_k = \int_{-1/2}^{1/2} \phi(t) e^{-\imath 2\pi k t} dt$	$\phi(t) = \sum_{k=-\infty}^{\infty} \Phi_k e^{\imath 2\pi k t}$
D, A	C, P	F.T. of a Sampled function	$\Phi(f) = \sum_{n=-\infty}^{\infty} \phi_n e^{-\imath 2\pi f n}$	$\phi_n = \int_{-1/2}^{1/2} \Phi(f) e^{\imath 2\pi f n} df$
D, P	D, P	DFT	$\Phi_k = \sum_{n=0}^{N-1} \phi_n e^{-\imath 2\pi k n/N}$	$\phi_n = \frac{1}{N} \sum_{k=0}^{N-1} \Phi_k e^{\imath 2\pi k n/N}$

Table 1: Four Discrete and Continuous Fourier Transform Pairs.

$$\text{DFT}[x_{n-n_0}] = e^{-\imath 2\pi k n_0/N} \sum_{l=-n_0}^{N-n_0-1} x_l e^{-\imath 2\pi k l/N} \tag{60}$$

because of the periodicity of x_l, we can shift the summation limits to obtain

$$\text{DFT}[x_{n-n_0}] = e^{-\imath 2\pi k n_0/N} \sum_{l=0}^{N-1} x_l e^{-\imath 2\pi k l/N} \tag{61}$$

or

$$\text{DFT}[x_{n-n_0}] = e^{-\imath 2\pi k n_0/N} X_k \ . \tag{62}$$

Parseval's theorem for the DFT is

$$\sum_{n=0}^{N-1} |x_n|^2 = \sum_{n=0}^{N-1} x_n x_n^* \tag{63}$$

$$\sum_{n=0}^{N-1} |x_n|^2 = \frac{1}{N^2} \sum_{n=0}^{N-1} \sum_{k=0}^{N-1} X_k e^{\imath 2\pi k n/N} \sum_{l=0}^{N-1} X_l^* e^{-\imath 2\pi l n/N} \tag{64}$$

$$\sum_{n=0}^{N-1} |x_n|^2 = \frac{1}{N^2} \sum_{n=0}^{N-1} \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} X_k X_l^* e^{\imath 2\pi n(k-l)/N} \ . \tag{65}$$

Evaluating the sum over n first, using (46) gives

$$\sum_{n=0}^{N-1} |x_n|^2 = \frac{1}{N^2} \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} X_k X_l^* N \delta_{k,l} \tag{66}$$

which gives

$$\sum_{n=0}^{N-1} |x_n|^2 = \frac{1}{N} \sum_{k=0}^{N-1} |X_k|^2 \ . \tag{67}$$

Many properties of the continuous Fourier transform also apply to the DFT, but we must be careful, as the DFT applies to a periodic sequence, and *not* to a finite series surrounded by an infinite number of zeros, as we might at first be tempted to conceptualize from our experience with continuous time series.

A very important application of the DFT is in implementing the discrete counterpart of the convolution theorem. Suppose we are given x_n and y_n, what series has the DFT $Z_k = X_k Y_k$?

$$z_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k Y_k e^{\imath 2\pi kn/N}. \tag{68}$$

$$z_n = \frac{1}{N} \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} \sum_{m=0}^{N-1} x_l e^{-\imath 2\pi lk/N} y_m e^{-\imath 2\pi mk/N} e^{\imath 2\pi kn/N}. \tag{69}$$

$$z_n = \frac{1}{N} \sum_{l=0}^{N-1} x_l \sum_{m=0}^{N-1} y_m \sum_{k=0}^{N-1} e^{\imath 2\pi k(n-m-l)/N}. \tag{70}$$

The innermost sum is zero whenever $n - m - l$ is not a multiple of N by (38), so we get

$$z_n = \text{IDFT}[X_k Y_k] = \sum_{l=0}^{N-1} x_l \; y_{n-l} \tag{71}$$

where it is understood that z_n is N-periodic, as are x_n and y_n.

Because the functions that we manipulate with the DFT and IDFT are periodic, and because the length of a convolution will be greater than or equal to the maximum length of its two constituent series (and equal only in the case where one is a Kronecker delta function), it is possible to get perhaps unexpected effects when applying (71).

Suppose we convolve two aperiodic series x_n and y_m in the time domain to obtain the *serial product* (this is what the MATLAB *conv* function does). Further suppose that x and y have N and M contiguous non-zero terms, respectively. The convolution is then

$$z_n = x_n * y_n = \sum_{l=-\infty}^{\infty} x_l \; y_{n-l} \tag{72}$$

and will then have $N + M - 1$ significant terms, bracketed by zeros.

What happens if we use the discrete convolution theorem to convolve the two functions? Here it is important to again keep in mind that the convolution theorem for discrete series corresponds to a convolution of periodic sequences. We must therefore take the period (i.e., the DFT size, L) to be longer than $N + M - 1$, otherwise there will not be room to squeeze the $N + M - 1$-length convolution result into an L-periodic result. We must therefore be careful to pad sequences with a suitable numbers of zeros to accurately mirror (72) using DFT techniques.

If $L < N + M - 1$, we get generally undesirable *wraparound* effects and the result will be different from the serial product, especially in its tails. Because of

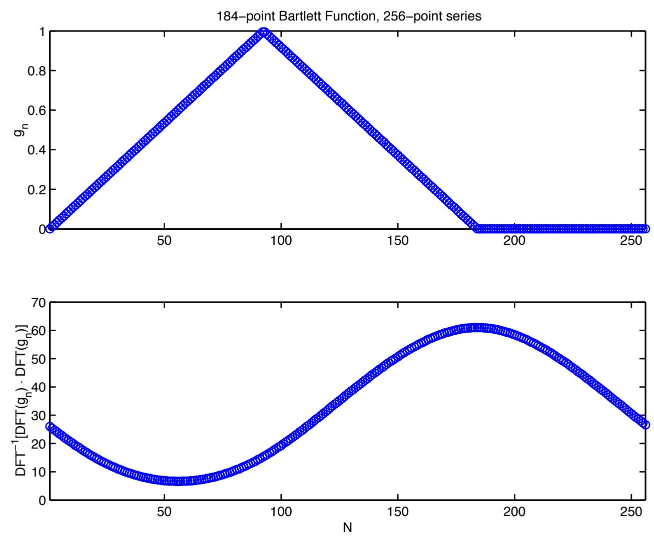

Figure 5: Wraparound in an $N = 256$-point Circular Convolution.

this wraparound, (71) strictly applies to what is referred to as *cyclic*, or *circular* convolution (Figure 5). One way to avoid wraparound is to pad functions with zeros (e.g., Figure 6).

Why bother to use (71) rather than (72) to evaluate convolutions? A major incentive arises because of a set of computer algorithms which first emerged in the mid 1960's (e.g., Cooley and Tukey, "An Algorithm for the Machine Computation of Complex Fourier Series", *Math. Comput.* , **19** , April, 1965). These *Fast Fourier Transform* or *FFT* algorithms evaluate the DFT, but in a much faster manner than the straightforward application of (48). Because large DFT's can be efficiently calculated using the FFT algorithm, it is much more efficient to evaluate a convolution by computing two DFT's, multiplying them, and then taking the inverse transform of the result, rather than by evaluating the serial product.

We will derive an FFT algorithm for the special case in which $N = 2^p$ is a power of 2. Similar ideas are used in algorithms that can work with more general values of N. Given a fixed input sequence x_n, Let

$$p(z) = x_0 + x_1 z + \ldots + x_{N-1} z^{N-1} \ . \tag{73}$$

Then the DFT of x is

$$X = \begin{bmatrix} p(w_N^0) \\ p(w_N^{-1}) \\ p(w_N^{-2}) \\ \vdots \\ p(w_N^{-(N-1)}) \end{bmatrix} \tag{74}$$

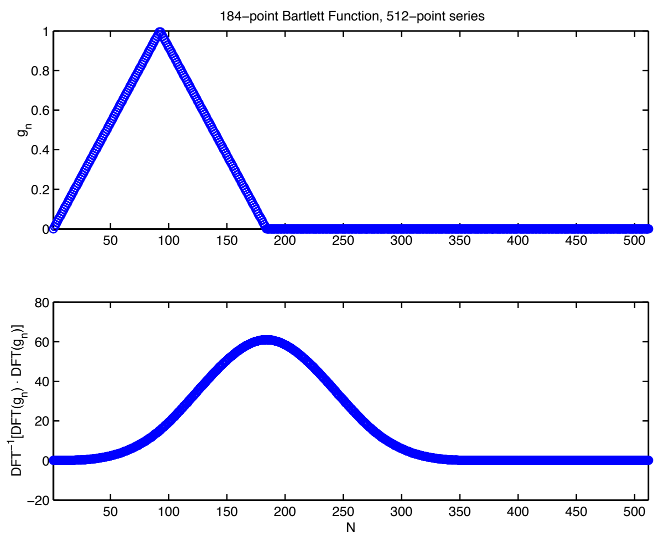

Figure 6: Same convolution as Figure 5, except with 256-point zero padding to eliminate wrap-around and thus emulate a noncircular convolution.

where $w_N = e^{\imath 2\pi/N}$.

We could use *Horner's rule* to evaluate the polynomial $p(z)$, which gives

$$p(z) = ((\cdots(x_{N-1}z + x_{N-2})z + x_{N-3})z + \ldots + x_1)z + x_0 \ . \tag{75}$$

Evaluating $p(z)$ in this way requires $N-1$ complex multiplications and $N-1$ complex additions. Computing the entire vector X takes N evaluations of $p(z)$, so computing the DFT in this manner takes $2N^2 - 2N = O(N^2)$ operations.

The FFT algorithm takes advantage of the fact that we are evaluating $p(z)$ only at powers of w_N^{-1}. We begin by breaking apart the even and odd powers in $p(z)$. Let

$$p_{\text{even}}(z) = \sum_{n=0 \ \ n \text{ even}}^{N-1} x_n z^{\frac{n}{2}} \tag{76}$$

and

$$p_{\text{odd}}(z) = \sum_{n=0 \ \ n \text{ odd}}^{N-1} x_n z^{\frac{n-1}{2}} \tag{77}$$

Then

$$p(z) = p_{\text{even}}(z^2) + z p_{\text{odd}}(z^2) \ . \tag{78}$$

For example, if $N = 8$, then

$$p(z) = x_0 + x_1 z + \ldots + x_7 z^7, \tag{79}$$

$$p_{\text{even}}(z) = x_0 + x_2 z + x_4 z^2 + x_6 z^3, \tag{80}$$

and

$$p_{\text{odd}}(z) = x_1 + x_3 z + x_5 z^2 + x_7 z^3. \tag{81}$$

Then

$$p_{\text{even}}(z^2) + z p_{\text{odd}}(z^2) = x_0 + x_1 z + \ldots + x_7 z^7 \ . \tag{82}$$

By using this decomposition of $p(z)$, we only need to evaluate z at even powers of w_N. Because of the periodicity of the powers of w_N, there are only $N/2$ points at which we have to evaluate the polynomial. For example, when $N = 8$, we need to evaluate $p(z)$ for

$$z = w_N^0, w_N^{-1}, \ldots, w_N^{-7}, \tag{83}$$

after removing multiples of $e^{i2\pi}$, the squares of these eight numbers are

$$z^2 = w_N^0, w_N^{-2}, w_N^{-4}, w_N^{-6}, w_N^0, w_N^{-2}, w_N^{-4}, w_N^{-6} \ . \tag{84}$$

Note that the even powers of w_N repeat twice. Thus we only need to evaluate p_{even} and p_{odd} at 4 points.

Let T_N be the number of arithmetic operations needed to evaluate the N point DFT. In other words, T_N is the number of arithmetic operations needed to evaluate an Nth degree polynomial $p(z)$ at w_N^0, w_N^{-1}, …, $w_N^{-(N-1)}$. By using our formula, we can reduce this to 2 evaluations of polynomials of degree $N/2$ at $N/2$ points plus N multiplications and N additions. Thus

$$T_N = 2T_{N/2} + 2N \ . \tag{85}$$

We won't find an explicit solution to this recurrence relation. However, we can easily compute a table of values of T_N for small values of N that are powers of 2. Table 2 shows operations counts for the naive algorithm and our FFT algorithm. Clearly, the FFT becomes much more efficient as N gets larger. In fact, it can be shown that the growth of T_N is $O(N \log N)$, while the growth of H_N is $O(N^2)$. For long signals with thousands or millions of samples, the FFT is vastly more efficient than the naive algorithm.

Computation of the convolution of two sequences of length N takes $O(N^2)$ time by direct evaluation of the convolution formula. If we use the convolution theorem for the DFT, then we can do the job by zero padding the sequences to length $2N$, computing two FFT's of length $2N$, performing $2N$ multiplications, and then doing an inverse FFT of length $2N$. Since

$$T_{2N} = 2T_N + 4N \leq 6T_N, \tag{86}$$

T_{2N} is $O(N \log N)$. Thus the entire FFT convolution process takes $O(N \log N)$ operations.

If we do not have large aliasing effects, so that the sampled sequence, x_n, adequately characterizes some near-band-limited continuous function in the real world, $\phi(t)$, then the DFT of the sequence x_n is just the spectrum of $\phi(t)$,

N	H_N	T_N
2	4	4
4	24	16
8	112	48
16	224	128
32	960	320
64	3972	768

Table 2: Operation counts for DFT by the naive algorithm (H_N) and FFT algorithm (T_N.)

sampled at the N equally-spaced frequency points. As we go to finer and finer sampling, we expect our calculated spectrum to approach the true spectrum, $\Phi(f)$. One way to help see that this is true (in a somewhat nonrigorous way) by considering the DFT when N becomes large.

It's instructive to investigate the convergence of the DFT to the Fourier Transform. Consider a discrete function defined by the N-point sequence, x_n. Taking the N-point DFT, where we'll take N to be odd, $N = 2M + 1$, we get

$$X_k = \sum_{n=-M}^{M} x_n e^{-\imath 2\pi kn/N} \tag{87}$$

Heuristically in the limit as N approaches infinity (finer and finer sampling), n remains discrete, but the function becomes aperiodic (we might conceptualize that it occupies the entire number line) and k thus become continuous (Table 4.1). The Fourier transform is thus

$$X(f) \equiv \sum_{n=-M}^{M} x_n e^{-\imath 2\pi fn} \ . \tag{88}$$

As a special case, consider the discrete rectangle function

$$\Pi_n = \begin{cases} 1 \text{ for } |n| \leq M \\ 0 \text{ for } |n| > M \end{cases} \tag{89}$$

Taking the Fourier Transform of (89), where $N = 2M + 1$, we get

$$\Pi(f) = \sum_{n=-M}^{M} e^{-\imath 2\pi fn} = e^{\imath 2\pi fM} \sum_{n=0}^{2M} e^{-\imath 2\pi fn} \tag{90}$$

$$= e^{\imath 2\pi fM} \frac{1 - e^{-\imath 2\pi(2M+1)f}}{1 - e^{-\imath 2\pi f}} = \frac{\sin(N\pi f)}{\sin(\pi f)} \equiv D(f) \tag{91}$$

Expressions of the form of (91) are a discrete, periodic analogue to the sinc function, occur frequently in discrete Fourier theory (e.g., in the kernel of the multitaper eigenfunction equation we noted in discussing power spectra. Such functions are commonly referred to as *Dirichlet kernels*. When N is large, the numerator of (91) oscillates much more rapidly than the denominator. Making the substitution $y = Nf$, (91) indeed then approaches the sinc function:

$$\lim_{N\to\infty} D(f) = \lim_{N\to\infty} \frac{\sin(\pi y)}{\sin(\pi y/N)} = \frac{\sin(\pi y)}{\pi y/N} = N \text{ sinc } y \ . \tag{92}$$