The Optimal Fourier Transform (OFT)

David M. W. Evans[1], 14 September 2013, last updated 18 October 2015

Abstract

The optimal Fourier transform (OFT) is a new development in Fourier analysis, with greater sensitivity and frequency resolution than the traditional discrete Fourier transform (DFT). It takes much longer to compute than the DFT, but offers benefits in analyzing noisy datasets. In particular, the OFT is better than the DFT at estimating the exact frequencies of sinusoids in a time series.

Like the DFT, the OFT estimates the spectrum of a time series, describing its sinusoids with a series of coefficients of cosines and sines. Unlike the DFT, it can analyze irregular time series (data points not equally spaced), it considers all frequencies (rather than just a small set of equally-spaced frequencies like the DFT), it orders the spectral sinusoids by amplitude (so the lesser ones, more likely to be describing noise, can be discarded or never computed), it typically describes the spectrum in far fewer sinusoids than a DFT (it stops when the sum of the spectral sinusoids is close enough to the original time series), but it is not invertible (the original time series cannot be exactly recovered from the OFT of the time series).

This paper includes examples of the OFT doing things that the DFT cannot do.

This paper also introduces the manual Fourier transform (MFT), which analyzes a time series into a spectrum of sinusoids at a given set of frequencies. The DFT is a special case of an MFT. The MFT is one of the key ingredients in the OFT. In turn the building blocks of the MFT are the four suprod functions, which are also introduced here.

The OFT, MFT, and suprods are original, as far as we know. They are unlikely to be completely original because the ideas are obvious, but we cannot find similar work elsewhere.

Administration

Cite as: Evans, David M.W., "*The Optimal Fourier Transform (OFT)*", sciencespeak.com, 2013.

Download: pdf

The spreadsheet *climate.xlsm* by the same author contains an implementation of the OFT and MFT, all the examples in this document, and applies the Fourier analysis described in this document to climate datasets. For context, see the notch-delay solar theory website.

[1] david.evans@sciencespeak.com, http://sciencespeak.com/

If reading this as a pdf, consider navigating the document by headings—click on the “Bookmarks” icon in the small column on the left of the pdf window.

If reading this in Microsoft Word, consider navigating the document by headings in the Navigation Pane—on the “View” tab, “Show” section, click “Navigation Pane”. Best at 120%.

1 Introduction

The traditional tool for estimating the spectrum of a time series is the discrete Fourier transform (DFT), but the DFT only looks for sinusoids at certain frequencies. If the time series consists of a single sinusoid at some other frequency, then the DFT will produce a spectrum of many sinusoids all at the wrong frequency (whose sum is the contained sinusoid, but that doesn’t alert you to the fact that the time series is a single sinusoid, let alone tell you its frequency). What if we want to detect the frequencies of sinusoids in the time series as precisely as possible?

The optimal Fourier transform (OFT) is a newly developed version of the Fourier transform that considers sinusoids at all frequencies (from zero to the Nyquist limit). It takes much longer to compute than the DFT, because it uses multi-variable function minimization to fit sums of sinusoids at variable frequencies to the time series.

In this document we briefly discuss the continuous Fourier transform and the discrete Fourier transform, applied to real-valued functions, in order to state our definitions and to explain those versions of the Fourier transform in the same notation as we use for the OFT.

Then we get to the new developments. We develop the four suprod functions, then the manual Fourier transform (MFT), then the OFT—because the suprods are the building blocks of the manual Fourier transform, which in turn is one of the basic building blocks of the OFT. Then we look at some examples comparing the OFT to the DFT, with the OFT doing things the DFT cannot. Finally we show how the OFT and MFT can be applied to irregular time series (those whose data points are not equally spaced in time).

The special functions I (indicator) and pha (phase) are used in this document; they are defined in Appendix A.

2 Sinusoids

In Fourier analysis, a function is expressed as a sum of sinusoids. In Fourier synthesis, which is the inverse of Fourier analysis, a bunch of sinusoids are added to form a function.

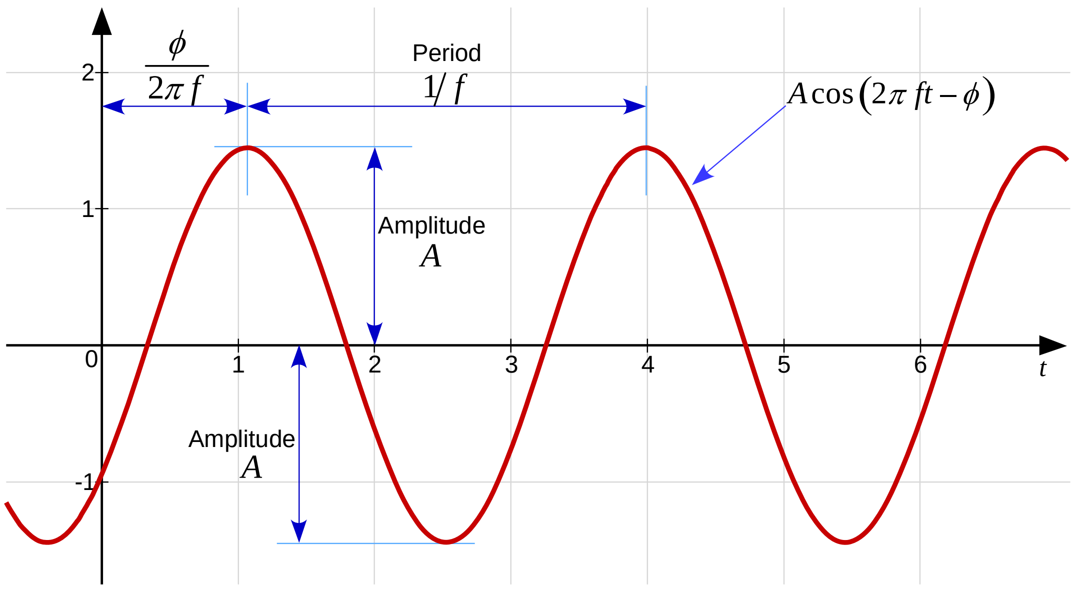


Figure 1: The sinusoid in *t* at frequency *f*, with amplitude *A* and phase *φ*.

The archetypal sinusoid in time t and at frequency f (in cycles per unit of time), with amplitude A and phase ϕ (in radians), is the function

$$t \mapsto A\cos(2\pi ft-\phi) = A\cos(\phi)\cos(2\pi ft) + A\sin(\phi)\sin(2\pi ft), \tag{1}$$

which is defined on all real numbers ($t \in \mathbb{R}$).

3 The Fourier Transform

The Fourier transform is a tool for analyzing a function of a continuous real variable (such as time) into a sum of sinusoids, called the spectrum of the function. We will define it and examine just a couple of its properties before moving on to time series and discrete transforms.

3.1 Fourier Transforms of Complex-Valued Functions

Let g be a function defined on all real numbers (such as for all time). Let g be complex-valued (because complex numbers are an accounting tool for representing sinusoids, this is somewhat unmotivated and even nonsensical, but it is traditional). Let $g(t)$ and $g(-t)$ remain finite as t becomes infinite. Let g not be "extremely" discontinuous (or the integrals here do not converge; this is generally not an issue with "real-world" functions). Let the Fourier transform of g be the complex-valued function F. Let the argument of F vary over all the real numbers and be called the frequency f.

Synthesis:

$$g(t) = \int_{-\infty}^{\infty} F(f)\exp(i2\pi ft)\,df \quad \text{for } t \in \mathbb{R}. \tag{2}$$

Analysis:

$$F(f) = \int_{-\infty}^{\infty} g(t)\exp(-i2\pi ft)\,dt \quad \text{for } f \in \mathbb{R}. \tag{3}$$

We write the real and imaginary parts of F as F_{real} and F_{img} (which are real-valued):

$$F(f) = F_{\text{real}}(f) + iF_{\text{img}}(f). \tag{4}$$

The relationship between g and its (complex-valued) complex Fourier transform F can be expressed by the complex Fourier transform operator $\mathbf{F}$:

$$\mathbf{F}\{g\} = f \mapsto F(f) \quad \text{or} \quad \mathbf{F}\{g\}(f) = F_{\text{real}}(f) + iF_{\text{img}}(f). \tag{5}$$

The Fourier transform synthesizes g as a sum of complex exponentials, typically

$$\exp(\pm i 2\pi ft) = \cos(2\pi ft) \pm i \sin(2\pi ft), \tag{6}$$

where i is the square root of -1 (complex numbers are extremely useful for representing sinusoids in the context of linear invariant systems; don't take the square root of -1 literally). Thus, after applying the complex multiplication in its integrand, the synthesis integral synthesizes g as a sum of sinusoids. The units of frequency are cycles per unit of time; for example, if t is measured in years then f is measured in cycles per year (cycles are dimensionless).

The synthesis employs one sinusoid at each frequency f (though see Fig. 1: a sinusoid with a negative frequency has the same period as a sinusoid with the absolute value of that frequency, which is ambiguous).

We can calculate F from g (by analysis, or the forward transform) and g from F (by synthesis, or the inverse transform), so the information in the function can be fully represented either as g (in which case we say it is in the time domain, if g is a function of time) or as F (in the frequency domain). The Fourier transform is thus invertible.

We haven't proved that, given the definition of the Fourier transform in the analysis Eq. (3), the Fourier transform synthesis in Eq. (2) is correct. There is an intricate mathematical proof, reasonably well-known, that we won't reproduce here.

3.2 Fourier Transforms of Real-Valued Functions

Almost all functions of interest are real-valued, and the Fourier transform becomes simpler when g is real-valued. Everything above about complex-valued functions still applies, because a real-valued function is just a complex-valued function whose imaginary part is zero.

If g is real-valued then its Fourier transform is complex-valued, but by Eq. (3)

$$\left.\begin{aligned} F_{\text{real}}(-f) &= F_{\text{real}}(f) \\ F_{\text{img}}(-f) &= -F_{\text{img}}(f) \end{aligned}\right\} \quad f \geq 0, \tag{7}$$

so the values of the Fourier transform at negative frequencies are redundant.

It is easier to work with Fourier transforms of real-valued functions by focusing on their cosine and sine parts, denoted by B_{C} and B_{S} respectively. (The "B" is for Professor Ronald Bracewell, late of Electrical Engineering at Stanford University, who played a large part in the modern revival of the Fourier transform, applied it in radio astronomy and image reconstruction, and wrote an influential text on Fourier transforms in 1978.) Further, we need only consider non-negative frequencies, because the values of the Fourier transform at negative frequencies give you no extra information about the spectrum of a real-valued function. These two policies remove the analysis of imaginary functions and redundant (aka aliased) frequencies from the picture, allowing us to focus just on the essentials without stumbling over irrelevant symmetries and unnecessary complications. Finally, the eta function η (η is the Greek letter "eta") is useful for taking care of the inevitable factors of two:

$$2^{\eta} = 2^{\eta(f)} = \begin{cases} 1 & \text{if } f = 0 \\ 2 & \text{if } f > 0. \end{cases} \tag{8}$$

η is simply the number of normal (that is, non-edge) frequencies in a context—here, because there is only one frequency variable, the only edge frequency is zero and η is either one or zero. We usually omit its frequency argument as understood and write "η" rather than "$\eta(f)$" in formulae. Now we can define the real Fourier transform (or Bracewell transform) of a real-valued function g.

Synthesis:

$$g(t) = \int_0^{\infty} \left[B_{\mathrm{C}}(f)\cos(2\pi ft) + B_{\mathrm{S}}(f)\sin(2\pi ft) \right] df \quad \text{for } t \in \mathbb{R}. \tag{9}$$

Analysis:

$$\left. \begin{aligned} B_{\mathrm{C}}(f) &= 2^{\eta}\int_{-\infty}^{\infty} g(t)\cos(2\pi ft)\,dt \\ B_{\mathrm{S}}(f) &= 2^{\eta}\int_{-\infty}^{\infty} g(t)\sin(2\pi ft)\,dt \end{aligned} \right\} \quad \text{for } f \geq 0. \tag{10}$$

The cosine and sine components B_{C} and B_{S} of the real Fourier transform are often combined into a complex number, giving a single analysis equation:

$$B(f) = B_{\mathrm{C}}(f) + iB_{\mathrm{S}}(f) = 2^{\eta}\int_{-\infty}^{\infty} g(t)\exp(i2\pi ft)\,dt, \quad f \geq 0. \tag{11}$$

The synthesis equation (9) then becomes a dot product:

$$g(t) = \int_0^{\infty} B(f) \bullet \exp(i2\pi ft)\,df, \quad t \in \mathbb{R}. \tag{12}$$

The dot product expands as in Eq. (9) in rectangular coordinates, while in polar coordinates

$$Ae^{i\phi} \bullet \exp(i2\pi ft) = A\cos(2\pi ft - \phi), \quad A, \phi \in \mathbb{R}. \tag{13}$$

The relationship between g and its (complex-valued) real Fourier transform B can be expressed by the real Fourier transform operator **B**:

$$\mathbf{B}\{g\} = f \mapsto B(f), \quad \text{or} \quad \mathbf{B}\{g\}(f) = B(f) = 2^{\eta}\int_{-\infty}^{\infty} g(t)\exp(i2\pi ft)\,dt. \tag{14}$$

For a real-valued function, the relationship between its complex Fourier transform and its real Fourier transform is

$$\left.\begin{aligned} B_C(f) &= \ \ 2^\eta F_{\text{real}}(f) \\ B_S(f) &= -2^\eta F_{\text{img}}(f) \end{aligned}\right\} \quad \text{for } f \geq 0 \tag{15}$$

or

$$B(f) = 2^\eta F^*(f) \quad \text{for } \ f \geq 0 \tag{16}$$

where the asterisk indicates the complex conjugate. (The 2^η factor may be regarded as "folding" the negative part of the real number line over onto the positive part, for frequency. The complex conjugate is an arbitrary sign change in the frequency in Eq. (3).)

The synthesis explicitly expresses *g* as a sum of sinusoids, one at each frequency (see Fig. 1; a sinusoid with a positive frequency has an unambiguous period).

4 Time Series and Discrete Transforms

The Fourier transform finds the spectrum of a function of continuous time. In climate research, for example, the temperature and solar functions of interest here are indeed functions of continuous time. However our measurements of functions is intrinsically discrete—we have only measurements taken at intervals, that is, we have only samples of the continuous time functions. For example, our information about the temperature and solar functions comes in the form of time series.

4.1 Time Series

The length-*N* time series (aka discrete function) *g* is an ordered set of *N* data points:

$$\left(g[\tau],\ \ \tau = 0,1,\ldots,N-1\right).$$

τ (the Greek letter "tau") is the time index; it is discrete and dimensionless. We often use square brackets rather than parentheses for the argument of *g*, as a reminder that τ is best thought of as an index rather than just a normal argument. For example, the time series $(5,\ 2,\ 17,\ 50)$ has four data points, which are the values of the discrete function *g* whose four values are $g[0]=5$, $g[1]=2$, $g[2]=17$, and $g[3]=50$.

The connection with functions of continuous time is that the data points of *g* are samples from some continuous-time function *h*.

Each data point is associated with or represents a time period, and the time periods butt up against one another—their union is the continuous time represented by the time series, and they do not intersect. The time of a data point is presumed to be in the middle of the time period associated with that data point, and is presumed to be the average value of a continuous time function over that time period. The extent of the time series is the total amount of continuous time represented by the time series, that is, the time from the start of the time period associated with the first data point to the end of the time period associated with the last data point.

In a regular time series the time between data points is always the same, the data points occurring at times

$$t_0 + \tau \frac{E}{N} = t_0 + \frac{\tau}{f_S}, \quad \tau = 0,1,\ldots,N-1, \tag{17}$$

where t_0 is the time associated with the first data point of the time series, E is the extent of the time series (from $t_0 - d/2$ to $t_{N-1} + d/2$, where d is the time between consecutive samples), and f_S is the sampling frequency ($f_S > 0$).

"Sampling" means that

$$g[\tau] = h\left(t_0 + \tau \frac{E}{N}\right) = h\left(t_0 + \frac{\tau}{f_S}\right). \tag{18}$$

The time between consecutive samples is the sampling period

$$\frac{1}{f_S} = \frac{E}{N}. \tag{19}$$

The extent E of the time series is thus

$$E = \left(t_{N-1} + \frac{1}{2 f_S}\right) - \left(t_0 - \frac{1}{2 f_S}\right). \tag{20}$$

Continuing the previous example, h could be the function $h: x \mapsto x^2 + 1$, with sampling starting at $t_0 = -2$ and a sample taken every $T_S = 3$ units of time.

In an irregular time series the time between adjacent data points are not all the same. All we can say is that the data points are at times

$$t_0, t_1, \ldots, t_{N-1}$$

where the times are ordered: $t_i < t_j$ whenever $i < j$, for $i, j \in \{0,1,\ldots,N-1\}$. The connection with functions of continuous time is that the data points of g can be samples from some continuous-time function h:

$$g[\tau] = h(t_\tau), \quad \tau = 0,1,\ldots,N-1.$$

The extent of the time series is still the crucial parameter for connecting the time index with the continuous time variable: this length of time should include a little time before the first data point and after the last. In the absence of further information, assume the distance between contiguous data points are about equal and so the extent of the time series is

$$E \approx \frac{N}{N-1}(t_{N-1} - t_0). \tag{21}$$

4.2 Sampled Sinusoids

Sampled sinusoids or sinusoidal time series are the time series formed by sampling continuous-time sinusoids. The length-*N* regular sampled sinusoid at frequency *f* (in cycles per unit of *t*), amplitude *A*, and phase ϕ, sampled at time t_0 and then regularly over extent *E* (or at sampling frequency f_S), is the time series

$$\left(A\cos(2\pi ft - \phi),\ t = t_0 + \tau\frac{E}{N} = t_0 + \frac{\tau}{f_S},\ \tau = 0, 1, \ldots, N-1 \right). \tag{22}$$

The irregular sampled sinusoid with the same parameters is

$$\left(A\cos(2\pi ft - \phi),\ t = t_0, t_1, \ldots, t_{N-1} \right).$$

4.3 Discrete Transforms

A discrete transform is the mathematical tool for estimating the spectrum of a sampled continuous time function, or equivalently, of computing the spectrum of a time series. The name "discrete" arises because a time series is often called a "discrete function", a mapping from a finite series to the real numbers.

A discrete transform expresses a time series as a sum of sampled sinusoids. A transform always comes in two parts, the analysis equation that tells how to calculate the transform coefficients, and the synthesis equation that tells how to form the time series from the coefficients. It is the synthesis equation of a discrete transform that (literally) expresses a time series as a sum of sampled sinusoids.

Crucially, a discrete transform of a time series is also an estimate of the spectrum of the continuous-time function from which the time series is sampled, that is, of the set of sinusoids whose sum approximates the continuous-time function. We say these sinusoids are "in the time series" and "in the continuous-time function".

4.4 Power

Analyses in the frequency domain often use power rather than amplitude. Power is a proxy for amplitude, because there is an invertible relationship between them and because a sinusoid with larger amplitude always has more power (except possibly near the edge frequencies—see below). Power is quicker to compute, the amplitude requiring the same computation as power but then a square root. In prior days of more limited computing speeds, this extra square root was a significant factor. Here we use amplitude because computing today is fast enough, because a sinusoid is more naturally characterized by its amplitude than by its power, and because it is more natural when computing the transfer functions of systems—whose amplitudes are simply the amplitude of the sinusoid in the output function divided by the amplitude of sinusoid in the input function at the same frequency.

5 Overview of Types of Discrete Transforms

There are several useful types of discrete transforms for analyzing time series. Each has different applicability, assumptions, strengths and weaknesses. The discrete transforms used here are all “Fourier transforms”, in the sense that they express a time series as sums of cosines and sines (or complex exponentials) in close analogy to the Fourier transform of continuous time functions. They differ principally in which frequencies they employ and whether they are for regular or irregular time series.

5.1 DFT

The standard and conventional discrete transform is *the* discrete Fourier transform (DFT). The DFT expresses a regular time series as a sum of cosines and sines at frequencies that belong to a special set, which has the property that the sampled cosines and sines at these frequencies are all orthogonal to one another (that is, they are linearly independent, or cannot be expressed as the sums of each other). For a length-N regular time series there are about $N/2$ such frequencies, and they depend only on N and the extent or sampling frequency of the time series:

$$0, \frac{1}{E}, \frac{2}{E}, \ldots, \left\lfloor \frac{N}{2} \right\rfloor \frac{1}{E} = 0, \frac{1}{N} f_S, \frac{2}{N} f_S, \ldots, \frac{1}{N} \left\lfloor \frac{N}{2} \right\rfloor f_S . \tag{23}$$

This reveals an important limitation of the DFT: it is not so good at determining the spectrum of a time series that contains sinusoids at frequencies other than the predetermined frequencies. Essentially the DFT assumes that the *only* sinusoids in the time series are those at the predetermined frequencies, then proceeds to find the amplitudes and phases of sinusoids at those assumed frequencies that sum exactly to the time series.

For instance, if the time series is purely sinusoidal at frequency $1.5/E$ say, which is half way between the second and third of the DFTs predetermined frequencies, then the DFT will construct a sum consisting of a sinusoid at frequency 0 with a small amplitude, a sinusoid at frequency $1/E$ with a larger amplitude, a sinusoid at frequency $2/E$ with a similar amplitude, and sinusoids at $3/E$, $4/E$, and so on with ever decreasing amplitudes. Although the DFT will construct a sum of sinusoids that perfectly adds to the time series, the spectrum implied by that sum is misleading—a transform that told us the time series was the sum of a single sampled sinusoid at frequency $1.5/E$ would often be preferable.

For a more concrete example, consider the PMOD dataset on total solar irradiance (TSI), the dataset of monthly TSI measurements by satellite from December 1978. Up to and including November 2012, the dataset has 408 data points and its extent is 34.0 years, so the periods corresponding to the frequencies used by its DFT are 34.0, 17.0, 11.33, 8.5, 6.8,…,0.17 years. The main solar cycle has a period of about 22 years (the effects of the solar cycles on Earth are often due to the *square* of the Sun’s magnetic field strength, which repeats about every 11 years but is a squared sinusoid rather than a sinusoid). Unfortunately the DFT is not using a frequency close to 22 years—if there were a strong 22 year sinusoid in the TSI, the DFT of the PMOD data would show it mainly as a strong 17 year component and a slightly less strong 34 year component. Wouldn’t it be good to be able to see directly what the PMOD data says about a sinusoid at 22 years?

5.2 MFT and OFT

It is often important to know as much about the true spectrum of a time series as possible. The DFT essentially introduces noise through its assumption of the frequencies of any contained sinusoids: constraining the frequencies of sinusoids in the spectrum is like adding noise because it prevents the signal from being estimated as accurately as possible. To avoid that we need to break that assumption, by considering other frequencies. If the time series is noisy, with weak signals compared to the noise level, that extra noise introduced by the DFT's assumption on frequencies is significant.

So, we need to use transforms that introduce as little noise as possible. Therefore we need transforms that do well at detecting the frequencies actually present in our time series, and express the time series out of sinusoids at such frequencies.

To that end, we introduce first the manual Fourier transform (MFT), which is basically the same as the DFT except that we must manually specify the frequencies of any sinusoids contained in the time series. From the time series and a set of frequencies, the MFT calculates the sum of sinusoids at those frequencies that best recreates the time series. Unlike the DFT, the frequencies are not restricted—they can be any frequencies at all. Nor do we have to give it any particular number of frequencies. However, if we give the MFT unrealistic frequencies or too few frequencies to work with, the result will be poor—the sum of the sinusoids will not match the time series well. So a measure of how well the sum matches the time series is an important part of the MFT results. The MFT allows us to experiment with different frequencies and see how good the fit is. The DFT is a special case of the MFT, where the specified frequencies are just those the DFT would use.

Secondly, we introduce the optimal Fourier transform (OFT). The OFT takes a time series and attempts to (a) find the frequencies of any contained sinusoids, then to (b) find the sinusoids at those frequencies. Conceptually, it uses a DFT to guess some initial frequencies, then uses multi-variable function minimization to choose a set of frequencies that minimize the mismatch between the time series and the MFT's sum of sinusoids at those frequencies. As with the MFT, an important part of the OFT result is how well the sum of sampled sinusoids it finds matches the time series.

Faster to Compute (↑)

FFT	Fast Fourier transform	Algorithm for computing the DFT more quickly.
DFT	Discrete Fourier transform	Predetermined frequencies only.
MFT	Manual Fourier transform	Any frequencies, frequencies manually specified.
OFT	Optimal Fourier transform	Any frequencies, frequencies found automatically.

Lower Noise (↓)

Table 1: Fourier transforms for regular time series, trading off speed for lower noise and greater frequency resolution.

A further limitation of the DFT is that it can only be applied to a regular time series. Temperature time series from proxy data in the far past are examples of irregular time series. So we introduce the irregular manual Fourier transform (iMFT) and the irregular optimal Fourier transform (iOFT) for dealing with irregular time series. Irregularity generally destroys the orthogonality of the sampled sinusoids employed by the DFT, so there is no iDFT. (At least formally—informally, just use an MFT with the frequencies that a DFT of a time series with the same number of data points and extent would use.)

The MFT and OFT are both defined and named here. Being fairly obvious ideas, neither are likely to be original, but I haven't been able to find anything too similar searching the Web. I expect that both have been developed in part or whole before, but have no knowledge of such (except that Tim Channon of Tallbloke's Talkshop may have developed something similar to the OFT with his minimal decomposition analysis). In any case, we need such tools for the analysis in the climate research mentioned in the *Administration* section above. Similarly for the irregular versions.

5.3 FFTs and Computation Speed

Historically a major theme in Fourier transforms has been computation speed. Transforms are computationally intensive. In applications like oil exploration where they were used extensively from the 1960s, when computers were *much* slower, a little extra noise was happily traded off for a lot more speed. However in the last decade computers have become fast enough that computation time is not critical for datasets of only a few thousand points, like our temperature and solar datasets. The MFT is at least an order of magnitude slower than the DFT, and the OFT is several orders of magnitude slower than the MFT.

A fast Fourier transform (FFT) is an algorithm for computing the DFT. Computing an FFT is considerably faster than computing the DFT directly as suggested by the DFT definition—an FFT of a length-N time series is $O(N \log N)$, while computing the DFT naïvely is $O(N^2)$. An FFT computes a DFT with divide and conquer approach, combining a small number of DFTs of sub-time-series together. However an FFT requires the factors of N to be small prime numbers, and the most common FFT programs require N to be a power of two. This restriction sometimes tempts people to arbitrarily change the data in the time series, such as by adding data points that are zero or to using "windowing", in order to apply an FFT. Software packages sometimes do this automatically. This raises the noise level in the dataset, making any signal harder to find.

The main temperature and TSI time series are rarely more a few thousand data points each. Naïvely computing the DFT of time series of these lengths on a modern computer is quick, much less than a second, even using a semi-interpreted language like VBA (the language in Microsoft Excel spreadsheets). An FFT is even quicker of course. An MFT usually takes less than a second, and an OFT rarely takes more than twenty minutes.

6 The Discrete Fourier Transform (DFT)

The discrete Fourier transform (DFT) of a time series is an estimate of its spectrum, which is the set of sinusoids whose sum approximates any continuous-time function from which the time series is sampled.

The DFT assumes that the time series is the sum of sinusoids only at particular pre-determined frequencies, and can only be applied to regular time series. The function of continuous time implied by those sinusoids repeats itself with a period equal to the extent of the time series.

6.1 DFT Frequencies

A DFT expresses a length-N time series g as a weighted sum of length-N sampled sinusoids. The argument of the sinusoidal functions used by the DFT is always

$$\frac{2\pi\nu\tau}{N}, \ \tau = 0,1,\ldots,N-1, \ \nu = 0,1,\ldots,N-1, \tag{24}$$

where ν (the Greek letter "nu") is the frequency index, and is discrete and dimensionless. ν is the frequency counterpart of τ, best thought of as an index rather than a normal variable. For example, the length-4 cosine time series are

$$\left(1, \cos\left(\frac{\pi}{2}\nu\right), \cos(\pi\nu), \cos\left(\frac{3\pi}{2}\nu\right)\right), \quad \nu = 0,1,2,3,$$

namely

$$(1,1,1,1), (1,0,-1,0), (1,-1,1,-1), (1,0,-1,0)$$

(notice the time series for $\nu = 1$ and $\nu = 3$ are the same, an example of the redundancy or aliasing in the DFT of real-valued time series, more on that below).

The connection with the continuous-time function h from which g is sampled is that, for the frequency index ν, the argument of the sinusoids (namely $2\pi\nu\tau/N$) increases by 2π when τ increases by N/ν and thus t increases by E/ν or $N/\nu f_S$. Thus ν corresponds to a continuous-time frequency f that has period E/ν or $N/\nu f_s$, namely the frequency

$$f = \frac{\nu}{E} = \frac{\nu}{N} f_S. \tag{25}$$

Thus the argument of the sinusoids is

$$\frac{2\pi\nu\tau}{N} = \frac{2\pi\left(\frac{fN}{f_S}\right)[(t-t_0)f_S]}{N} = 2\pi f(t-t_0).$$

6.2 DFTs for Complex-Valued Time Series

Let the (complex) DFT of g be F, which is thus a complex-valued length-N time series.

Synthesis:

$$g[\tau] = \sum_{\nu=0}^{N-1} F[\nu]\exp\left(\frac{i2\pi\nu\tau}{N}\right) \ \text{ for } \tau = 0,1,\ldots,N-1. \tag{26}$$

Analysis:

$$F[\nu] = N^{-1}\sum_{\tau=0}^{N-1} g[\tau]\exp\left(-\frac{i2\pi\nu\tau}{N}\right) \quad \text{for } \nu = 0,1,\ldots,N-1. \tag{27}$$

We can calculate F from g (by analysis, or the "forward DFT") and g from F (by synthesis, or the "inverse DFT"), so the information in the time series can be fully represented either as g (in which case we say it is in the "time domain") or as F (in the "frequency domain").

6.3 DFTs for Real-Valued Time Series

The time series of interest here, temperature and solar signals such as total solar irradiance (TSI), are real-valued. There are significant simplifications to the DFT that apply when g is real-valued. Everything above still applies, because a complex-valued time series is also a real-valued time series, but in this section we go into the detail required to perform the necessary computations.

If g is real-valued then its DFT is still complex-valued, but let us explicitly consider its real and imaginary parts:

$$F[\nu] = F_{\text{real}}[\nu] + i\,F_{\text{img}}[\nu], \tag{28}$$

where F_{real} and F_{img} are real-valued. A little algebra with the complex DFT analysis Eq. (27) reveals that when g is real-valued

$$\begin{aligned} F_{\text{real}}[\nu] &= F_{\text{real}}[kN+\nu] = F_{\text{real}}[kN-\nu] \\ F_{\text{img}}[\nu] &= F_{\text{img}}[kN+\nu] = -F_{\text{img}}[kN-\nu] \end{aligned} \tag{29}$$

for any integer k. Consequently we ignore the values of the DFT at frequency indices outside the range $[0, N/2]$, because they are redundant or aliased. The Nyquist frequency, the highest frequency in a continuous-time function that can be unambiguously detected by regular sampling at a rate f_S, is $f = f_S/2$, which corresponds to $\nu = N/2$.

It is easier to work with Fourier transforms of real-valued functions by focusing on their cosine and sine parts, denoted by B_C and B_S respectively and defined as the multipliers of the cosine and sine sinusoids in the synthesis, and by focusing on non-negative frequencies (thereby removing imaginary numbers and redundant frequencies from the picture). Two useful constants are the maximum cosine frequency index ϑ_C ("nu-max-C") and the maximum sine frequency index ϑ_S ("nu-max-S"):

$$\begin{aligned} \vartheta_C &= \lfloor N/2 \rfloor \\ \vartheta_S &= \lfloor (N-1)/2 \rfloor \\ \vartheta_C + \vartheta_S &= N-1. \end{aligned} \tag{30}$$

The eta function is handy for taking care of factors of two:

$$2^{\eta} = 2^{\eta[\nu]} = \begin{cases} 2 & \text{if } \nu = 1,\ldots,\vartheta_S \\ 1 & \text{if } \nu = 0, N/2. \end{cases} \tag{31}$$

η (the Greek letter "eta") is the number of non-edge (normal) frequencies. Note that $\nu = N/2$ only occurs if N is even. The edge frequencies ("edge frequency indexes", if the obvious shortcut is not used) are those at the edges of the allowed frequency range, namely

$$\nu = 0 \quad \text{and, if and only if } N \text{ is even,} \quad \nu = \vartheta_C = \frac{N}{2}.$$

The edge frequency behavior is often different from the behavior at non-edge frequencies, so it is an important distinction. Often the difference is captured by the eta function. We can now define the real DFT (or Bracewell discrete transform, the BFT) B of g:

Synthesis:

$$g[\tau] = \sum_{\nu=0}^{\vartheta_C} \left\{ B_C[\nu]\cos\left(\frac{2\pi\nu\tau}{N}\right) + B_S[\nu]\sin\left(\frac{2\pi\nu\tau}{N}\right) \right\} \quad \text{for } \tau = 0,1,\ldots,N-1. \tag{32}$$

Analysis:

$$\left.\begin{aligned} B_C[\nu] &= 2^{\eta} N^{-1} \sum_{\tau=0}^{N-1} g[\tau]\cos\left(\frac{2\pi\nu\tau}{N}\right) \\ B_S[\nu] &= 2^{\eta} N^{-1} \sum_{\tau=0}^{N-1} g[\tau]\sin\left(\frac{2\pi\nu\tau}{N}\right) \end{aligned}\right\} \quad \text{for } \nu = 0,1,\ldots,\vartheta_C. \tag{33}$$

Spectrum of g:

$$\left\{ B_C[\nu]\cos\left(\frac{2\pi\nu\tau}{N}\right) + B_S[\nu]\sin\left(\frac{2\pi\nu\tau}{N}\right),\ \nu = 0,1,\ldots,\vartheta_C \right\} \tag{34}$$

Amplitude spectrum of g:

$$\text{amp}[\nu] = \left|B[\nu]\right| = \sqrt{B_C^2[\nu] + B_S^2[\nu]} \quad \text{for } \nu = 0,1,\ldots,\vartheta_C \tag{35}$$

Phase spectrum of g:

$$\text{phase}[\nu] = \text{pha}\left(B_C[\nu], B_S[\nu]\right) \quad \text{for } \nu = 0,1,\ldots,\vartheta_C \tag{36}$$

Relationship between the complex DFT and real DFT:

$$\left.\begin{aligned} F_{\text{real}}[\nu] &= \ \ 2^{-\eta} B_C[\nu] \\ F_{\text{img}}[\nu] &= -2^{-\eta} B_S[\nu] \end{aligned}\right\} \quad \text{for } \nu = 0,1,\ldots,\vartheta_C. \tag{37}$$

Notice that:

- $B_C[\nu]$ is generally non-zero for $\nu = 0,1,\ldots,\vartheta_C$, so there are $1+\vartheta_C$ cosine parts.
- $B_S[\nu]$ is generally non-zero for $\nu = 1,\ldots,\vartheta_S$, so there are ϑ_S sine parts.
- There are N independent real numbers in the DFT that are not redundant or identically zero ($1+\vartheta_C+\vartheta_S = N$), the same number as in the time series g. The number of degrees of freedom, or information, is preserved when moving between g and F.

- At the edge frequencies, the value of $B_S[\nu]$ is zero by the analysis equation above, and by convention it is zero. But because it simply does not appear in the synthesis equation, it could be anything at all. This is noteworthy because this convention can sometimes create a discontinuity in an otherwise smoothly varying estimate of $B_S[\nu]$ as ν varies, in contexts where ν can be non-integral.
- The real DFT expresses the time series as a sum of sinusoids at the frequencies

$$\nu = 0, 1, \ldots, \vartheta_C, \quad \text{or } f = 0, \frac{1}{E}, \ldots, \frac{\vartheta_C}{E}, \quad \text{or } f = 0, \frac{1}{N} f_S, \ldots, \frac{\vartheta_C}{N} f_S$$

(see the synthesis equation)., Only these pre-determined frequencies are used by the DFT; they are determined just by *N* and either *E* or f_S. These frequencies are special: they are uniquely the frequencies from zero to the Nyquist frequency (inclusive) that collectively have *N* cosines and sines (that aren't identically zero) that are orthogonal under time summation (next section), or equivalently, that are linearly independent when sampled at the same times as the continuous time function *h* was sampled. However, using only these frequencies forces the DFT to represent sinusoids at other frequencies as sums of these frequencies, which can be misleading at times.

6.4 Sinusoidal Orthogonality for Regular Time Series

In order to prove that the real DFT is invertible (next section), we need to know about the orthogonality of the sinusoids used by the DFT. For $\nu, \mu \in \{0, 1, \ldots, \vartheta_C\}$,

$$\sum_{\tau=0}^{N-1} \cos\left(\frac{2\pi\nu\tau}{N} - \phi\right) \cos\left(\frac{2\pi\mu\tau}{N} - \theta\right) = 2^{-\eta} N I_{\nu=\mu} \begin{cases} \cos 2\pi(\phi - \theta) & \nu = 1, \ldots, \vartheta_S \\ \cos(2\pi\phi)\cos(2\pi\theta) & \nu = 0, N/2. \end{cases} \quad (38)$$

Special cases:

$$\sum_{\tau=0}^{N-1} \cos\left(\frac{2\pi\nu\tau}{N}\right) \cos\left(\frac{2\pi\mu\tau}{N}\right) = 2^{-\eta} N I_{\nu=\mu}$$
$$\sum_{\tau=0}^{N-1} \cos\left(\frac{2\pi\nu\tau}{N}\right) \sin\left(\frac{2\pi\mu\tau}{N}\right) = 0$$
$$\sum_{\tau=0}^{N-1} \sin\left(\frac{2\pi\nu\tau}{N}\right) \sin\left(\frac{2\pi\mu\tau}{N}\right) = 2^{-\eta} N I_{\nu=\mu} I_{1\le\mu\le\vartheta_S}.$$

Proof: First note that

$$\sum_{\tau=0}^{N-1} \cos\left(\frac{2\pi\lambda\tau}{N} - \phi\right) = \cos(\phi) \sum_{\tau=0}^{N-1} \cos\left(\frac{2\pi\lambda\tau}{N}\right) + \sin(\phi) \sum_{\tau=0}^{N-1} \sin\left(\frac{2\pi\lambda\tau}{N}\right) = N\cos(\phi) I_{\lambda=0,\pm N,\pm 2N,\ldots}$$

To see this, consider a unit circle on an *x-y* plane. Let *M* equal *N* divided by the greatest common divisor of *N* and λ. Draw *M* equally spaced points around the circle, including a point at the intersection of the circle with the positive *x* axis. The angles $2\pi\lambda\tau/N$ are the angles the points make with the origin and the positive *x* axis; cosines are projections onto the *x* axis, and sines are projections onto the *y* axis. By symmetry the sums of cosines and sines are zero, except when $M = 1$. Now to the statement to be proved:

$$\begin{aligned}\text{LHS} &= \frac{1}{2}\sum_{\tau=0}^{N-1}\cos\left(\frac{2\pi(\nu+\mu)\tau}{N}-(\phi+\theta)\right)+\frac{1}{2}\sum_{\tau=0}^{N-1}\cos\left(\frac{2\pi(\nu-\mu)\tau}{N}-(\phi-\theta)\right)\\ &= \frac{N}{2}\cos(\phi+\theta)\left[I_{\nu=\mu=0}+I_{\nu=\mu=N/2}\right]+\frac{N}{2}\cos(\phi-\theta)I_{\nu=\mu}\\ &= N I_{\nu=\mu}\begin{cases}\frac{1}{2}\text{co}(\phi-\theta) & \nu=1,\ldots,\vartheta_S\\ \frac{1}{2}\text{co}(\phi+\theta)+\frac{1}{2}\text{co}(\phi-\theta) & \nu=0,N/2\end{cases}\\ &= \text{RHS}. \qquad \square\end{aligned}$$

6.5 DFT Invertibility

Consider a sinusoidal time series g with amplitude A and phase ϕ at any of the frequencies used by the real DFT:

$$\begin{aligned} g[\tau] &= A\cos\left(\frac{2\pi\mu\tau}{N}-\phi\right)\\ &= A\cos(\phi)\cos\left(\frac{2\pi\mu\tau}{N}\right)+A\sin(\phi)\sin\left(\frac{2\pi\mu\tau}{N}\right), \quad \mu\in\{0,1,\ldots,\vartheta_C\}. \end{aligned} \tag{39}$$

Comparison with the synthesis equation shows that its cosine and sine parts are

$$\left.\begin{aligned} B_C[\nu] &= A\cos(\phi)I_{\nu=\mu}\\ B_S[\nu] &= A\sin(\phi)I_{\nu=\mu}\end{aligned}\right\} \text{ for } \nu=0,1,\ldots,\vartheta_C. \tag{40}$$

Two related quantities that are natural to compute are the cosine and sine averages of g:

$$\left.\begin{aligned} C_{\text{avg}}[\nu] &= N^{-1}\sum_{\tau=0}^{N-1} g[\tau]\cos\left(\frac{2\pi\nu\tau}{N}\right)\\ S_{\text{avg}}[\nu] &= N^{-1}\sum_{\tau=0}^{N-1} g[\tau]\cos\left(\frac{2\pi\nu\tau}{N}\right)\end{aligned}\right\} \text{ for } \nu=0,1,\ldots,\vartheta_C. \tag{41}$$

The relationships between the parts and averages, such that the DFT synthesis equation is correct (and the DFT is thus invertible), are found by substituting for g and expanding, then applying the special cases of sinusoidal orthogonality (see last section):

$$\left.\begin{aligned} C_{\text{avg}}[\nu] &= N^{-1}\sum_{\tau=0}^{N-1} g[\tau]\cos\left(\frac{2\pi\nu\tau}{N}\right) = A\cos(\phi)2^{-\eta}I_{\nu=\mu} = 2^{-\eta}B_C[\nu]\\ S_{\text{avg}}[\nu] &= N^{-1}\sum_{\tau=0}^{N-1} g[\tau]\sin\left(\frac{2\pi\nu\tau}{N}\right) = A\sin(\phi)2^{-\eta}I_{\nu=\mu}I_{1\le\mu\le\vartheta_S} = 2^{-\eta}B_S[\nu]\end{aligned}\right\} \nu=0,1,\ldots,\vartheta_C. \tag{42}$$

These give the formulae for the cosine and sine parts in the DFT analysis equations. This proves the DFT analysis and synthesis formulae above are correct, and that the DFT is invertible, for real-valued time series.

6.6 Some Observations about the DFT

There is an infinity of possible sums of sinusoidal time series that add to a given time series. The DFT expresses the time series as the weighted sum of all the mutually linearly independent sinusoidal time series whose frequencies range from 0 to the Nyquist frequency, of which there are exactly N. This sum is unique.

The DFT expresses a time series as a sum of orthogonal sinusoidal time series, which has the advantage that the DFT is invertible—we can compute the time series from its DFT and vice versa, and move between time and frequency domains without loss of information. In applications where we just want to estimate the spectral sinusoids and their frequencies as well as possible, invertibility is not valuable or even relevant—such as in the climate research application mentioned in the *Administration* section above. Moreover, that orthogonality comes at a price.

One disadvantage is that the DFT pre-determines which frequencies it assumes are in g, and these frequencies are determined merely from the data length N and the extent E (or equivalently, the sampling frequency f_S). So if g happens to be a sum of sampled sinusoids at frequencies other than those predetermined ones, the DFT will construct g from frequencies or sinusoids that are not in fact present in g—and while it will do a reasonable job, using nearby frequencies as one might expect, it will not be correct.

Another disadvantage is that all of the sinusoids used by the DFT have an integral number of periods in the length of the time series, so the continuous-time function made from those sinusoids repeats itself infinitely, with a period equal to the length of the time series. In many applications this does not matter, but it is worth bearing in mind because often it is obvious that the continuous-time function is not in fact periodic and does not repeat forever with a period equal to the length of data that happened to be collected.

7 The Regular Suprod Functions

To develop discrete versions of the Fourier transform that can express a time series using sinusoids at frequencies other than the pre-determined ones employed by the DFT, we need to let the frequency index ν be non-integral instead of just integral. With the DFT we considered only the frequency indices $0,1,\ldots,\vartheta_C$, where $\vartheta_C = \lfloor N/2 \rfloor$; now we consider all real frequency indices, that is, all the real numbers in $[0, N/2]$.

This means that the sampled sinusoids used to synthesize the time series will no longer necessarily be orthogonal under time summation. We therefore need to explore what happens with non-orthogonal sinusoids. This leads to the four special functions that are the subject of this section, the "regular suprod functions". (The name "suprod", reminiscent of "sums of products", is coined here. The "regular" in the name is because they are only used to analyze regular time series; we later use a slight variation to analyze irregular time series.).

When the frequency index ν is integral the values of the suprods are zero, a half, or one, so they do not appear to be explicitly present—though of course they are present, just that their effect is fully accounted for by applying the sinusoidal orthogonality relationships. But when ν is allowed to be non-integral the suprods are explicitly required, in all their messy detail.

The suprod functions generalize the sinusoidal orthogonality relationships. They are the scaling factors that connect the cosine and sine sums with the cosine and sine parts.

The four regular suprod functions for any positive integer N and real numbers ν and μ are defined by

$$
\begin{aligned}
\mathrm{cc}_N(\nu,\mu) &= N^{-1}\sum_{\tau=0}^{N-1}\cos\left(\frac{2\pi\nu\tau}{N}\right)\cos\left(\frac{2\pi\mu\tau}{N}\right)\\
\mathrm{cs}_N(\nu,\mu) &= N^{-1}\sum_{\tau=0}^{N-1}\cos\left(\frac{2\pi\nu\tau}{N}\right)\sin\left(\frac{2\pi\mu\tau}{N}\right)\\
\mathrm{sc}_N(\nu,\mu) &= N^{-1}\sum_{\tau=0}^{N-1}\sin\left(\frac{2\pi\nu\tau}{N}\right)\cos\left(\frac{2\pi\mu\tau}{N}\right) = \mathrm{cs}_N(\mu,\nu)\\
\mathrm{ss}_N(\nu,\mu) &= N^{-1}\sum_{\tau=0}^{N-1}\sin\left(\frac{2\pi\nu\tau}{N}\right)\sin\left(\frac{2\pi\mu\tau}{N}\right).
\end{aligned}
\tag{43}
$$

In our use, N is the number of data points in the time series under consideration, while ν and μ are frequency indices. To avoid redundant (aliased) frequencies, we are only interested in frequency indices in $[0, N/2]$:

$$
\begin{aligned}
\mathrm{cc}_N(\nu+jN,\mu+kN) &= \mathrm{cc}_N(\nu,\mu)\\
\mathrm{cs}_N(\nu+jN,\mu+kN) &= \mathrm{cs}_N(\nu,\mu)\\
\mathrm{sc}_N(\nu+jN,\mu+kN) &= \mathrm{sc}_N(\nu,\mu)\\
\mathrm{ss}_N(\nu+jN,\mu+kN) &= \mathrm{ss}_N(\nu,\mu)
\end{aligned}
\tag{44}
$$

for any integers j and k, while

$$
\begin{aligned}
\mathrm{cc}_N(\pm\nu,\pm\mu) &= \mathrm{cc}_N(\nu,\mu)\\
\mathrm{cc}_N(\pm\nu,\mp\mu) &= \mathrm{cc}_N(\nu,\mu)\\
\mathrm{cs}_N(\pm\nu,+\mu) &= \mathrm{cs}_N(\nu,\mu)\\
\mathrm{cs}_N(\pm\nu,-\mu) &= -\mathrm{cs}_N(\nu,\mu)\\
\mathrm{sc}_N(+\nu,\pm\mu) &= \mathrm{sc}_N(\nu,\mu)\\
\mathrm{sc}_N(-\nu,\mp\mu) &= -\mathrm{sc}_N(\nu,\mu)\\
\mathrm{ss}_N(\pm\nu,\pm\mu) &= \mathrm{ss}_N(\nu,\mu)\\
\mathrm{ss}_N(\pm\nu,\mp\mu) &= -\mathrm{ss}_N(\nu,\mu).
\end{aligned}
\tag{45}
$$

For integral values of ν and μ, the values of the regular suprod functions were calculated when we looked at the sinusoidal orthogonality for regular time series:

$$
\begin{aligned}
\mathrm{cc}_N(\nu,\mu) &= 2^{-\eta} I_{\nu=\mu}\\
\mathrm{cs}_N(\nu,\mu) &= 0\\
\mathrm{sc}_N(\nu,\mu) &= 0\\
\mathrm{ss}_N(\nu,\mu) &= 2^{-\eta} I_{\nu=\mu} I_{1\le\mu\le\vartheta_S}
\end{aligned}
\tag{46}
$$

Otherwise, the values of the sums of products are best viewed by numerical calculation:

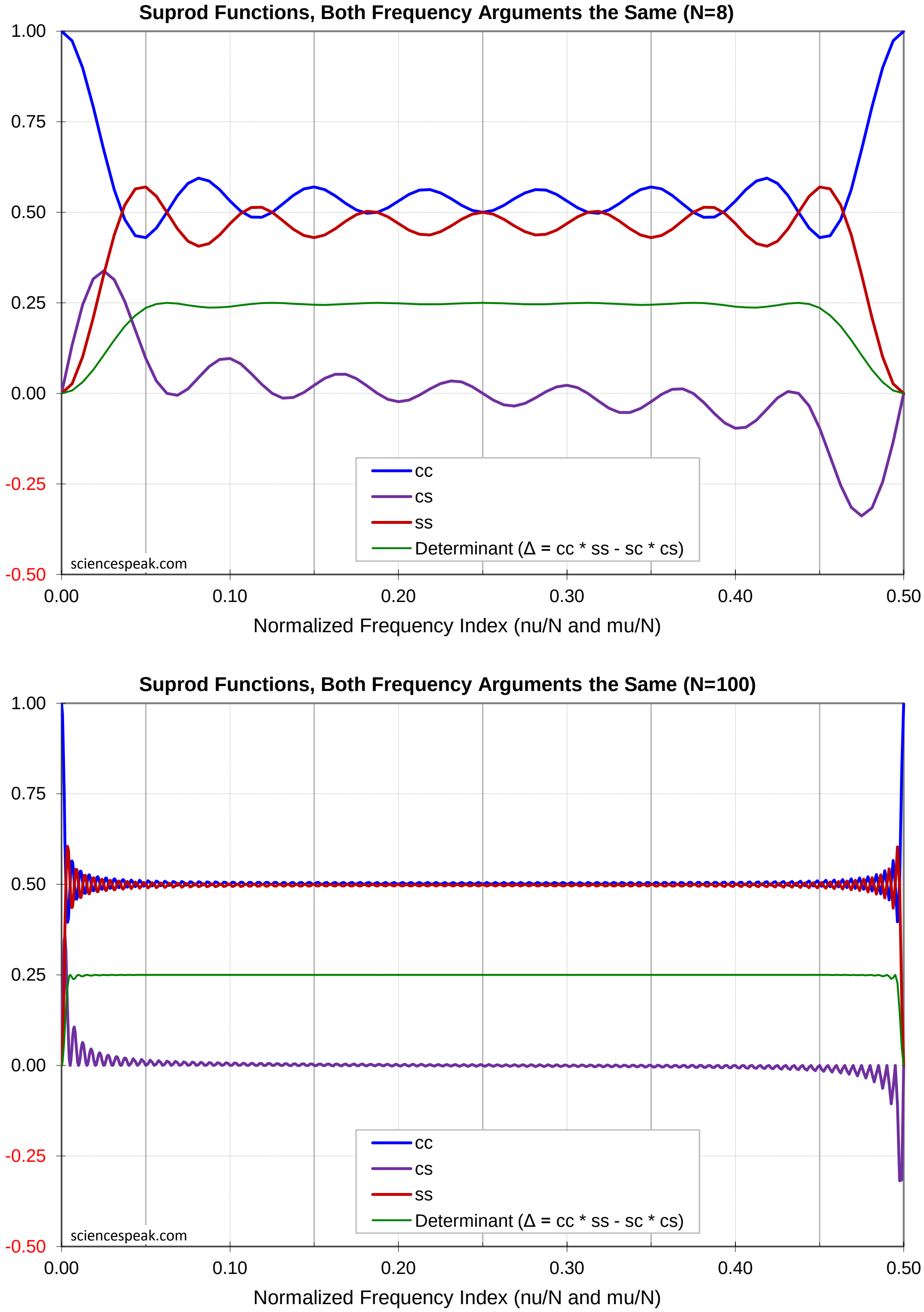
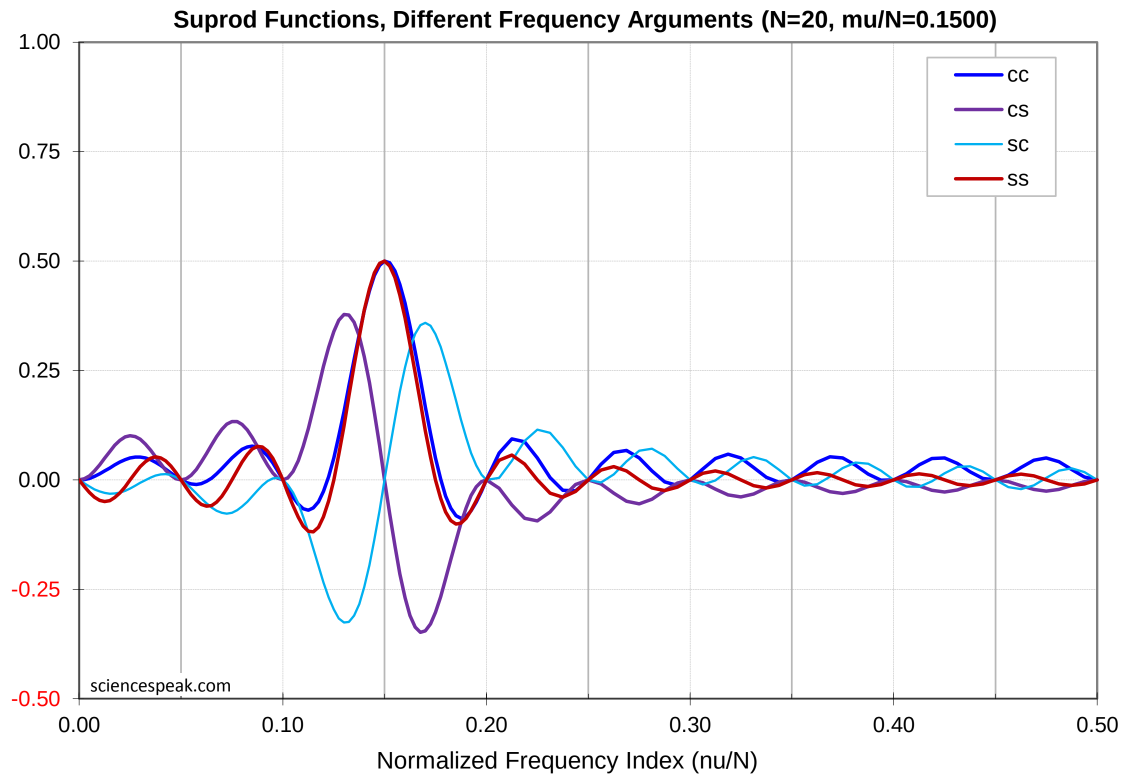


Figure 2: The suprod functions when the two frequency arguments are the same, showing the scaling factors for the cosine and sine parts.



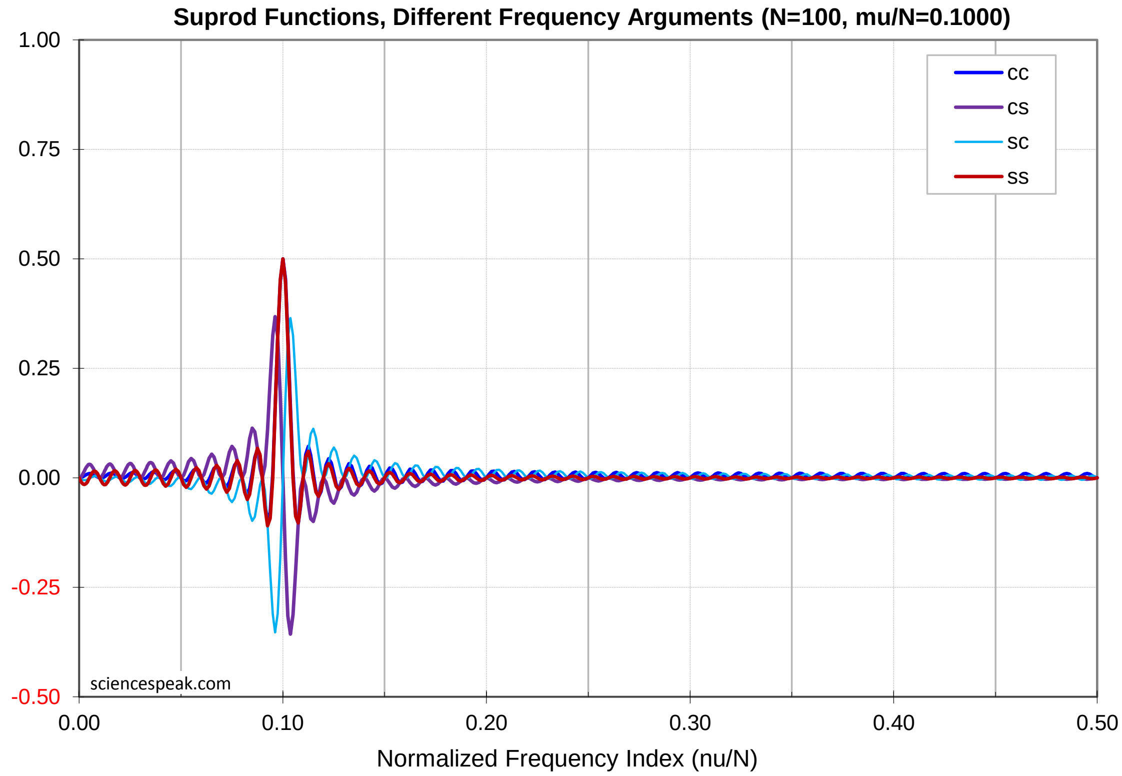


Figure 3: The suprod functions when the frequency arguments are different, showing the price for non-orthogonality.

To speed computation, we can exploit symmetry in each sum around $\tau = N/2$. Note that

$$\begin{aligned}\cos\left(\frac{2\pi\nu(N-\tau)}{N}\right) &= \cos(2\pi\nu)\cos\left(\frac{2\pi\nu\tau}{N}\right)+\sin(2\pi\nu)\sin\left(\frac{2\pi\nu\tau}{N}\right)\\ \sin\left(\frac{2\pi\nu(N-\tau)}{N}\right) &= \sin(2\pi\nu)\cos\left(\frac{2\pi\nu\tau}{N}\right)-\cos(2\pi\nu)\sin\left(\frac{2\pi\nu\tau}{N}\right)\end{aligned} \tag{47}$$

and, for brevity, let

$$\begin{aligned}\mathrm{CC} &= \cos\left(\frac{2\pi\nu\tau}{N}\right)\cos\left(\frac{2\pi\mu\tau}{N}\right) & \mathrm{cc} &= \cos(2\pi\nu)\cos(2\pi\mu)\\ \mathrm{CS} &= \cos\left(\frac{2\pi\nu\tau}{N}\right)\sin\left(\frac{2\pi\mu\tau}{N}\right) & \mathrm{cs} &= \cos(2\pi\nu)\sin(2\pi\mu)\\ \mathrm{SC} &= \sin\left(\frac{2\pi\nu\tau}{N}\right)\cos\left(\frac{2\pi\mu\tau}{N}\right) & \mathrm{sc} &= \sin(2\pi\nu)\cos(2\pi\mu)\\ \mathrm{SS} &= \sin\left(\frac{2\pi\nu\tau}{N}\right)\sin\left(\frac{2\pi\mu\tau}{N}\right) & \mathrm{ss} &= \sin(2\pi\nu)\sin(2\pi\mu).\end{aligned} \tag{48}$$

Then

$$\begin{aligned}N\,\mathrm{cc}_N(\nu,\mu) &= 1+\cos(\pi\nu)\cos(\pi\mu)I_{N\text{ even}}\\ &\quad+\sum_{\tau=1}^{\vartheta_C}\cos\left(\frac{2\pi\nu\tau}{N}\right)\cos\left(\frac{2\pi\mu\tau}{N}\right)+\sum_{\tau=1}^{\vartheta_C}\cos\left(\frac{2\pi\nu(N-\tau)}{N}\right)\cos\left(\frac{2\pi\mu(N-\tau)}{N}\right)\\ &= 1+\cos(\pi\nu)\cos(\pi\mu)I_{N\text{ even}}+\sum_{\tau=1}^{\vartheta_C}\mathrm{CC}+\sum_{\tau=1}^{\vartheta_C}\left[\mathrm{cc}\cdot\mathrm{CC}+\mathrm{cs}\cdot\mathrm{CS}+\mathrm{sc}\cdot\mathrm{SC}+\mathrm{ss}\cdot\mathrm{SS}\right].\end{aligned}$$

Similarly, for all four regular suprod functions:

$$\begin{aligned}N\,\mathrm{cc}_N(\nu,\mu) &= 1+\cos(\pi\nu)\cos(\pi\mu)I_{N\text{ even}}+\sum_{\tau=1}^{\vartheta_C}\mathrm{CC}+\sum_{\tau=1}^{\vartheta_C}\left[\mathrm{cc}\cdot\mathrm{CC}+\mathrm{cs}\cdot\mathrm{CS}+\mathrm{sc}\cdot\mathrm{SC}+\mathrm{ss}\cdot\mathrm{SS}\right].\\ N\,\mathrm{cs}_N(\nu,\mu) &= \cos(\pi\nu)\sin(\pi\mu)I_{N\text{ even}}+\sum_{\tau=1}^{\vartheta_C}\mathrm{CS}+\sum_{\tau=1}^{\vartheta_C}\left[\mathrm{cs}\cdot\mathrm{CC}-\mathrm{cc}\cdot\mathrm{CS}+\mathrm{ss}\cdot\mathrm{SC}-\mathrm{sc}\cdot\mathrm{SS}\right]\\ N\,\mathrm{sc}_N(\nu,\mu) &= \sin(\pi\nu)\cos(\pi\mu)I_{N\text{ even}}+\sum_{\tau=1}^{\vartheta_C}\mathrm{SC}+\sum_{\tau=1}^{\vartheta_C}\left[\mathrm{sc}\cdot\mathrm{CC}+\mathrm{ss}\cdot\mathrm{CS}-\mathrm{cc}\cdot\mathrm{SC}-\widehat{\mathrm{cs}\cdot\mathrm{SS}}\right]\\ N\,\mathrm{ss}_N(\nu,\mu) &= \sin(\pi\nu)\sin(\pi\mu)I_{N\text{ even}}+\sum_{\tau=1}^{\vartheta_C}\mathrm{SS}+\sum_{\tau=1}^{\vartheta_C}\left[\widehat{\mathrm{ss}\cdot\mathrm{CC}}-\widehat{\mathrm{sc}\cdot\mathrm{CS}}-\widehat{\mathrm{cs}\cdot\mathrm{SC}}+\widehat{\mathrm{cc}\cdot\mathrm{SS}}\right].\end{aligned} \tag{49}$$

This cuts computation by as much as a half if cosine and sine evaluations are much more expensive than individual multiplications and additions, because the hat-quantities are independent of τ. Extending further:

$$\begin{aligned}\cos\left(\frac{2\pi\nu(\frac{1}{2}N\pm\tau)}{N}\right) &= \cos(\pi\nu)\cos\left(\frac{2\pi\nu\tau}{N}\right)\mp\sin(\pi\nu)\sin\left(\frac{2\pi\nu\tau}{N}\right)\\ \sin\left(\frac{2\pi\nu(\frac{1}{2}N\pm\tau)}{N}\right) &= \sin(\pi\nu)\cos\left(\frac{2\pi\nu\tau}{N}\right)\pm\cos(\pi\nu)\sin\left(\frac{2\pi\nu\tau}{N}\right)\end{aligned}$$

and so on ($N/4\pm\tau, N/8\pm\tau,$ etc).

More decisively from the point of view of fast computation, the regular suprods are amenable to a divide and conquer strategy, like the FFT does for the DFT. Let p be any integral divisor of N greater than or equal to two (in practice p will be a prime). The key observation is

$$\begin{aligned} \mathrm{cc}_{N/p}\left(\tfrac{1}{p}\nu, \tfrac{1}{p}\mu\right) &= \frac{p}{N}\sum_{\tau=0}^{N/p-1} \cos\left(\frac{2\pi(\nu/p)\tau}{N/p}\right)\cos\left(\frac{2\pi(\mu/p)\tau}{N/p}\right) \\ &= \frac{p}{N}\sum_{\tau=0}^{N/p-1} \cos\left(\frac{2\pi\nu\tau}{N}\right)\cos\left(\frac{2\pi\mu\tau}{N}\right). \end{aligned} \tag{50}$$

For brevity, let

$$\begin{aligned} \mathrm{cc}_i &= \cos\left(\frac{2\pi i\nu}{p}\right)\cos\left(\frac{2\pi i\mu}{p}\right) & \mathrm{CC}_p &= \mathrm{cc}_{N/p}\left(\tfrac{1}{p}\nu, \tfrac{1}{p}\mu\right) \\ \mathrm{cs}_i &= \cos\left(\frac{2\pi i\nu}{p}\right)\sin\left(\frac{2\pi i\mu}{p}\right) & \mathrm{CS}_p &= \mathrm{cs}_{N/p}\left(\tfrac{1}{p}\nu, \tfrac{1}{p}\mu\right) \\ \mathrm{sc}_i &= \sin\left(\frac{2\pi i\nu}{p}\right)\cos\left(\frac{2\pi i\mu}{p}\right) & \mathrm{SC}_p &= \mathrm{sc}_{N/p}\left(\tfrac{1}{p}\nu, \tfrac{1}{p}\mu\right) \\ \mathrm{ss}_i &= \sin\left(\frac{2\pi i\nu}{p}\right)\sin\left(\frac{2\pi i\mu}{p}\right) & \mathrm{SS}_p &= \mathrm{ss}_{N/p}\left(\tfrac{1}{p}\nu, \tfrac{1}{p}\mu\right). \end{aligned}$$

Then

$$\begin{aligned} N\,\mathrm{cc}_N(\nu,\mu) &= \sum_{\tau=0}^{N-1} \cos\left(\frac{2\pi\nu\tau}{N}\right)\cos\left(\frac{2\pi\mu\tau}{N}\right) \\ &= \sum_{i=0}^{p-1}\sum_{\tau=iN/p}^{(i+1)N/p-1} \cos\left(\frac{2\pi\nu\tau}{N}\right)\cos\left(\frac{2\pi\mu\tau}{N}\right) \\ &= \sum_{i=0}^{p-1}\sum_{\tau=0}^{N/p-1} \cos\left(\frac{2\pi\nu(\tau+iN/p)}{N}\right)\cos\left(\frac{2\pi\mu(\tau+iN/p)}{N}\right) \\ &= \sum_{i=0}^{p-1}\sum_{\tau=0}^{N/p-1} \cos\left(\frac{2\pi i\nu}{p}+\frac{2\pi\nu\tau}{N}\right)\cos\left(\frac{2\pi i\mu}{p}+\frac{2\pi\mu\tau}{N}\right) \\ &= \sum_{i=0}^{p-1}\sum_{\tau=0}^{N/p-1} \left\{\begin{array}{l} \left[\cos\left(\frac{2\pi i\nu}{p}\right)\cos\left(\frac{2\pi\nu\tau}{N}\right)-\sin\left(\frac{2\pi i\nu}{p}\right)\sin\left(\frac{2\pi\nu\tau}{N}\right)\right] \\ \quad\cdot\left[\cos\left(\frac{2\pi i\mu}{p}\right)\cos\left(\frac{2\pi\mu\tau}{N}\right)-\sin\left(\frac{2\pi i\mu}{p}\right)\sin\left(\frac{2\pi\mu\tau}{N}\right)\right] \end{array}\right\} \\ &= \frac{N}{p}\mathrm{CC}_p + \frac{N}{p}\sum_{i=1}^{p-1}\left[\overbrace{\mathrm{cc}_i\,\mathrm{CC}_p} - \overbrace{\mathrm{cs}_i\,\mathrm{CS}_p} - \overbrace{\mathrm{sc}_i\,\mathrm{SC}_p} + \overbrace{\mathrm{ss}_i\,\mathrm{SS}_p}\right]. \end{aligned}$$

Similarly for the other three suprods, so the suprod extension equations are:

$$\begin{aligned}
p\,\mathrm{cc}_N(\nu,\mu) &= \mathrm{CC}_p + \sum_{i=1}^{p-1}\left[\mathrm{cc}_i\,\mathrm{CC}_p - \mathrm{cs}_i\,\mathrm{CS}_p - \mathrm{sc}_i\,\mathrm{SC}_p + \mathrm{ss}_i\,\mathrm{SS}_p\right]\\
p\,\mathrm{cs}_N(\nu,\mu) &= \mathrm{CS}_p + \sum_{i=1}^{p-1}\left[\mathrm{cs}_i\,\mathrm{CC}_p + \mathrm{cc}_i\,\mathrm{CS}_p - \mathrm{ss}_i\,\mathrm{SC}_p - \mathrm{sc}_i\,\mathrm{SS}_p\right]\\
p\,\mathrm{sc}_N(\nu,\mu) &= \mathrm{SC}_p + \sum_{i=1}^{p-1}\left[\mathrm{sc}_i\,\mathrm{CC}_p - \mathrm{ss}_i\,\mathrm{CS}_p + \mathrm{cc}_i\,\mathrm{SC}_p - \mathrm{cs}_i\,\mathrm{SS}_p\right]\\
p\,\mathrm{ss}_N(\nu,\mu) &= \mathrm{SS}_p + \sum_{i=1}^{p-1}\left[\mathrm{ss}_i\,\mathrm{CC}_p + \mathrm{sc}_i\,\mathrm{CS}_p + \mathrm{cs}_i\,\mathrm{SC}_p + \mathrm{cc}_i\,\mathrm{SS}_p\right].
\end{aligned} \tag{51}$$

Thus the regular suprod function for (ν,μ) of length N can be quickly constructed from the regular suprod function for $\left(\tfrac{1}{p}\nu,\tfrac{1}{p}\mu\right)$ of length N/p. If N can be factored into small prime factors, this tactic reduces its computation from $O(N)$ to $O(\log N)$.

8 Estimating Contained Sinusoids

Once we move away from using the pre-determined frequencies of the DFT, we need to solve the following problem: what are the sampled sinusoids in a given time series at a given set of frequencies, assuming that those are the only frequencies present in the time series?

That is, assuming that the time series is equal to a sum of sampled sinusoids with frequencies drawn only from the given set of frequencies, what are the amplitude and phase of the sinusoid at each of the frequencies in the given set? This problem is called “estimating the sampled sinusoids contained in a time series” here.

The problem has a unique solution, involving suprods. The solution is exact if the assumption is correct (but becomes increasingly wrong as the number and amplitudes of sinusoids at other frequencies in the time series increase). It is this solution that makes the MFT and OFT possible. When the given set of frequencies is the set of frequencies used by the DFT, the answer is essentially the DFT—so the following solution may be considered to be a generalization of the DFT.

8.1 One Sinusoid

Suppose we have time series g containing one sampled sinusoid. Let g be a length-N time series with amplitude A and phase ϕ at some given frequency index $\mu\in[0,N/2]$, where μ may be integral or non-integral:

$$g[\tau] = A\cos\left(\frac{2\pi\mu\tau}{N}-\phi\right) = A\cos(\phi)\cos\left(\frac{2\pi\mu\tau}{N}\right) + A\sin(\phi)\sin\left(\frac{2\pi\mu\tau}{N}\right),\quad \tau = 0,1,\dots N-1. \tag{52}$$

The cosine and sine parts of g are defined as the multipliers of the purely cosine and sine sample sinusoids in the synthesis of g, so, trivially, they are

$$\left.\begin{aligned}
B_C(\nu) &= A\cos(\phi) I_{\nu=\mu}\\
B_S(\nu) &= A\sin(\phi) I_{\nu=\mu} I_{\nu\in(0,N/2)}
\end{aligned}\right\}\quad \text{for } \nu\in[0,N/2]. \tag{53}$$

So how do we find these from *g*? The obvious (and perhaps the only reasonable) starting quantities are the cosine and sine averages of *g*, namely

$$\left.\begin{aligned} C_{\text{avg}}(\nu) &= N^{-1}\sum_{\tau=0}^{N-1} g[\tau]\cos\left(\frac{2\pi\nu\tau}{N}\right) \\ S_{\text{avg}}(\nu) &= N^{-1}\sum_{\tau=0}^{N-1} g[\tau]\sin\left(\frac{2\pi\nu\tau}{N}\right) \end{aligned}\right\} \quad \text{for } \nu\in\left[0, N/2\right]. \tag{54}$$

Substituting for *g* and expanding reveals they are linear combinations of the suprod functions:

$$\left.\begin{aligned} C_{\text{avg}}(\nu) &= A\cos(\phi)\,\text{cc}_N(\nu,\mu) + A\sin(\phi)\,\text{cs}_N(\nu,\mu) \\ S_{\text{avg}}(\nu) &= A\cos(\phi)\,\text{sc}_N(\nu,\mu) + A\sin(\phi)\,\text{ss}_N(\nu,\mu) \end{aligned}\right\} \quad \text{for } \nu\in\left[0, N/2\right]. \tag{55}$$

In particular when $\nu=\mu$, and recognizing the cosine and sine parts:

$$\begin{aligned} C_{\text{avg}}(\mu) &= B_C(\mu)\,\text{cc}_N(\mu,\mu) + B_S(\mu)\,\text{cs}_N(\mu,\mu) \\ S_{\text{avg}}(\mu) &= B_C(\mu)\,\text{sc}_N(\mu,\mu) + B_S(\mu)\,\text{ss}_N(\mu,\mu). \end{aligned} \tag{56}$$

Solving these two equations (Cramer's rule),

$$\begin{aligned} B_C(\mu) &= \frac{\text{ss}_N(\mu,\mu)C_{\text{avg}}(\mu) - \text{cs}_N(\mu,\mu)S_{\text{avg}}(\mu)}{\Delta} \\ B_S(\mu) &= \frac{\text{cc}_N(\mu,\mu)S_{\text{avg}}(\mu) - \text{sc}_N(\mu,\mu)C_{\text{avg}}(\mu)}{\Delta} \end{aligned}$$

where

$$\begin{aligned} \Delta &= \text{cc}_N(\mu,\mu)\,\text{ss}_N(\mu,\mu) - \text{sc}_N(\mu,\mu)\,\text{cs}_N(\mu,\mu) \\ &= \text{cc}_N(\mu,\mu)\,\text{ss}_N(\mu,\mu) - \text{cs}_N^2(\mu,\mu). \end{aligned}$$

These are the exact solutions for the cosine and sine parts at μ; at other frequency indices they are zero, by our original constraint that *g* contains only a single sampled sinusoid, at frequency index μ. Note however that the cosine and sine averages at frequencies other than μ are generally non-zero, close to zero at frequencies far from μ but larger at frequencies closer to μ. It is worth stressing that these are exact solutions only when there truly is only one sinusoid in the time series; if there are other sinusoids in the time series then the solution is an estimate but not quite correct, and the closer the other sinusoids are in frequency the less accurate is the above solution.

If μ is an edge frequency then

- $\sin\left(2\pi\mu\tau/N\right)=0$ for $\tau=0,1,\ldots N-1$
- $\text{cc}_N(\mu,\mu)=1$ and $\text{cs}_N(\mu,\mu)=\text{sc}_N(\mu,\mu)=\text{ss}_N(\mu,\mu)=0$
- $S_{\text{avg}}(\mu)=0$
- $B_S(\mu)=0$ by convention (it is irrelevant to the synthesis equation).
- The system of two linear equation reduces to just one equation:

$$C_{\text{avg}}(\mu)=B_C(\mu).$$

To test this solution, consider the approximations. When μ is not near an edge frequency:

$$\begin{aligned}\mu &\in \left[\tfrac{1}{2}, \tfrac{1}{2}N - \tfrac{1}{2}\right] \\ \mathrm{cc}_N(\mu,\mu) &\simeq \mathrm{ss}_N(\mu,\mu) \simeq \tfrac{1}{2} \\ \mathrm{cs}_N(\mu,\mu) &= \mathrm{sc}_N(\mu,\mu) \simeq 0 \\ \Delta &\simeq 1/4 \\ B_C(\mu) &\simeq 2C_{\mathrm{avg}}(\mu) \\ B_S(\mu) &\simeq 2S_{\mathrm{avg}}(\mu).\end{aligned}$$

When μ is close to an edge frequency:

$$\begin{aligned}\mu &\simeq 0, \tfrac{1}{2}N \\ \mathrm{cc}_N(\mu,\mu) &\simeq 1 \\ \mathrm{cs}_N(\mu,\mu) &\simeq \mathrm{sc}_N(\mu,\mu) \simeq \mathrm{ss}_N(\mu,\mu) \simeq 0 \\ \Delta &\simeq 0 \\ B_C(\mu) &\simeq C_{\mathrm{avg}}(\mu) \\ B_S(\mu) &\simeq 0.\end{aligned}$$

In both approximations the results agree with the DFT of g for integral values of μ.

8.2 Two Sinusoids

The next simplest case is a length-N time series g that is the sum of two sampled sinusoids, the first with amplitude A_1 and phase ϕ_1 at frequency $\mu_1 \in [0, N/2]$ and the second with amplitude A_2 and phase ϕ_2 at frequency $\mu_2 \in [0, N/2]$:

$$g[\tau] = A_1 \cos\left(\frac{2\pi\mu_1\tau}{N} - \phi_1\right) + A_2 \cos\left(\frac{2\pi\mu_2\tau}{N} - \phi_2\right), \text{ for } \tau = 0,1,\ldots N-1. \tag{57}$$

The cosine and sine parts of g are thus

$$\left.\begin{aligned}B_C(\nu) &= A_1\cos(\phi_1) I_{\nu=\mu_1} + A_2\cos(\phi_2) I_{\nu=\mu_2} \\ B_S(\nu) &= \left[A_1\sin(\phi_1) I_{\nu=\mu_1} + A_2\sin(\phi_2) I_{\nu=\mu_2}\right] I_{\nu\in(0,N/2)}\end{aligned}\right\} \text{ for } \nu \in [0, N/2]. \tag{58}$$

Substituting our expression for g above into the cosine and sine averages of g,

$$\begin{aligned}C_{\mathrm{avg}}(\nu) = {} & A_1\cos(\phi_1)\,\mathrm{cc}_N(\nu,\mu_1) + A_1\sin(\phi_1)\,\mathrm{cs}_N(\nu,\mu_1) \\ & + A_2\cos(\phi_2)\,\mathrm{cc}_N(\nu,\mu_2) + A_2\sin(\phi_2)\,\mathrm{cs}_N(\nu,\mu_2) \\ S_{\mathrm{avg}}(\nu) = {} & A_1\cos(\phi_1)\,\mathrm{sc}_N(\nu,\mu_1) + A_1\sin(\phi_1)\,\mathrm{ss}_N(\nu,\mu_1) \\ & + A_2\cos(\phi_2)\,\mathrm{sc}_N(\nu,\mu_2) + A_2\sin(\phi_2)\,\mathrm{ss}_N(\nu,\mu_2)\end{aligned}$$

for $\nu \in [0, N/2]$. Setting $\nu = \mu_1$ then $\nu = \mu_2$ and recognizing the cosine and sine parts,

$$\begin{aligned}C_{\mathrm{avg}}(\mu_1) &= B_C(\mu_1)\,\mathrm{cc}_N(\mu_1,\mu_1) + B_S(\mu_1)\,\mathrm{cs}_N(\mu_1,\mu_1) + B_C(\mu_2)\,\mathrm{cc}_N(\mu_1,\mu_2) + B_S(\mu_2)\,\mathrm{cs}_N(\mu_1,\mu_2) \\ S_{\mathrm{avg}}(\mu_1) &= B_C(\mu_1)\,\mathrm{sc}_N(\mu_1,\mu_1) + B_S(\mu_1)\,\mathrm{ss}_N(\mu_1,\mu_1) + B_C(\mu_2)\,\mathrm{sc}_N(\mu_1,\mu_2) + B_S(\mu_2)\,\mathrm{ss}_N(\mu_1,\mu_2) \\ C_{\mathrm{avg}}(\mu_2) &= B_C(\mu_1)\,\mathrm{cc}_N(\mu_2,\mu_1) + B_S(\mu_1)\,\mathrm{cs}_N(\mu_2,\mu_1) + B_C(\mu_2)\,\mathrm{cc}_N(\mu_2,\mu_2) + B_S(\mu_2)\,\mathrm{cs}_N(\mu_2,\mu_2) \\ S_{\mathrm{avg}}(\mu_2) &= B_C(\mu_1)\,\mathrm{sc}_N(\mu_2,\mu_1) + B_S(\mu_1)\,\mathrm{ss}_N(\mu_2,\mu_1) + B_C(\mu_2)\,\mathrm{sc}_N(\mu_2,\mu_2) + B_S(\mu_2)\,\mathrm{ss}_N(\mu_2,\mu_2).\end{aligned}$$

This set of linear equations can always be solved for the cosine and sine parts that are not identically zero, thereby solving the problem:

- If μ_1 and μ_2 are not edge frequencies then there are four linearly independent equations for the four cosine and sine parts.
- If μ_1 is an edge frequency and μ_2 is not, then $B_S(\mu_1)$ is identically zero and

  $$\sin(2\pi\mu_1\tau/N)=0 \quad \text{for } \tau=0,1,\dots N-1.$$

  Thus $\mathrm{cc}_N(\mu_1,\mu_1)=1$, the second equation becomes $S_{\mathrm{avg}}(\mu_1)=0$, and the remaining three equations are

  $$\begin{aligned} C_{\mathrm{avg}}(\mu_1) &= B_C(\mu_1) && +B_C(\mu_2)\mathrm{cc}_N(\mu_1,\mu_2) && +B_S(\mu_2)\mathrm{cs}_N(\mu_1,\mu_2) \\ C_{\mathrm{avg}}(\mu_2) &= B_C(\mu_1)\mathrm{cc}_N(\mu_2,\mu_1) && +B_C(\mu_2)\mathrm{cc}_N(\mu_2,\mu_2) && +B_S(\mu_2)\mathrm{cs}_N(\mu_2,\mu_2) \\ S_{\mathrm{avg}}(\mu_2) &= B_C(\mu_1)\mathrm{sc}_N(\mu_2,\mu_1) && +B_C(\mu_2)\mathrm{sc}_N(\mu_2,\mu_2) && +B_S(\mu_2)\mathrm{ss}_N(\mu_2,\mu_2). \end{aligned}$$

  This is a set of three linearly-independent linear equations, and can be solved for the remaining three cosine and sine parts: $B_C(\mu_1)$, $B_C(\mu_2)$, and $B_S(\mu_2)$.
- If both μ_1 and μ_2 are edge frequencies then $B_S(\mu_1)$ and $B_S(\mu_2)$ are identically zero, the second and fourth equations become $S_{\mathrm{avg}}(\mu_1)=0$ and $S_{\mathrm{avg}}(\mu_2)=0$, and the remaining two equations become

  $$\begin{aligned} C_{\mathrm{avg}}(\mu_1) &= B_C(\mu_1)+B_C(\mu_2)\mathrm{cc}_N(\mu_1,\mu_2) \\ C_{\mathrm{avg}}(\mu_2) &= B_C(\mu_1)\mathrm{cc}_N(\mu_2,\mu_1)+B_C(\mu_2). \end{aligned}$$

  This is a set of two linearly-independent linear equations, and can be solved for the remaining two cosine parts: $B_C(\mu_1)$ and $B_C(\mu_2)$.

A useful approximation arises if μ_1 and μ_2 are not close to each other: their joint suprod values (those with arguments (μ_1,μ_2) or (μ_2,μ_1)) are close to zero, so the four equations become simply

$$\begin{aligned} C_{\mathrm{avg}}(\mu_1) &\simeq B_C(\mu_1)\mathrm{cc}_N(\mu_1,\mu_1)+B_S(\mu_1)\mathrm{cs}_N(\mu_1,\mu_1) \\ S_{\mathrm{avg}}(\mu_1) &\simeq B_C(\mu_1)\mathrm{sc}_N(\mu_1,\mu_1)+B_S(\mu_1)\mathrm{ss}_N(\mu_1,\mu_1) \\ C_{\mathrm{avg}}(\mu_2) &\simeq B_C(\mu_2)\mathrm{cc}_N(\mu_2,\mu_2)+B_S(\mu_2)\mathrm{cs}_N(\mu_2,\mu_2) \\ S_{\mathrm{avg}}(\mu_2) &\simeq B_C(\mu_2)\mathrm{sc}_N(\mu_2,\mu_2)+B_S(\mu_2)\mathrm{ss}_N(\mu_2,\mu_2), \end{aligned}$$

which are just two independent instances of the one-sinusoid case.

8.3 Many Sinusoids

Suppose the length-*N* time series *g* is the sum of *m* sampled sinusoids, $m\in\{1,2,\dots\}$, with amplitude A_i and phase ϕ_i at frequency $\mu_i\in[0,N/2]$ for $i=1,\dots,m$:

$$g[\tau]=\sum_{i=1}^{m}A_i\cos\left(\frac{2\pi\mu_i\tau}{N}-\phi_i\right), \text{ for } \tau=0,1,\dots N-1. \quad (59)$$

The cosine and sine parts of *g* are thus

$$\left.\begin{aligned} B_C(\nu) &= \sum_{k=1}^{m} A_k \cos(\phi_k) I_{\nu=\mu_k} \\ B_S(\nu) &= \left[\sum_{k=1}^{m} A_k \sin(\phi_k) I_{\nu=\mu_k}\right] I_{\nu\in(0,N/2)} \end{aligned}\right\} \quad \text{for } \nu \in [0, N/2]. \tag{60}$$

Substituting our expression for g above into the cosine and sine averages of g,

$$\left.\begin{aligned} C_{\text{avg}}(\nu) &= \sum_{i=1}^{m} \left[A_i \cos(\phi_i)\,\text{cc}_N(\nu,\mu_i) + A_i \sin(\phi_i)\,\text{cs}_N(\nu,\mu_i)\right] \\ S_{\text{avg}}(\nu) &= \sum_{i=1}^{m} \left[A_i \cos(\phi_i)\,\text{sc}_N(\nu,\mu_i) + A_i \sin(\phi_i)\,\text{ss}_N(\nu,\mu_i)\right] \end{aligned}\right\} \quad \text{for } \nu \in [0, N/2]. \tag{61}$$

Setting $\nu = \mu_1$ then $\nu = \mu_2$ and so on to $\nu = \mu_m$, and recognizing the cosine and sine parts, we get the $2m$ linear equations

$$\left.\begin{aligned} C_{\text{avg}}(\mu_j) &= \sum_{i=1}^{m} \left[B_C(\mu_j)\,\text{cc}_N(\mu_j,\mu_i) + B_S(\mu_j)\,\text{cs}_N(\mu_j,\mu_i)\right] \\ S_{\text{avg}}(\mu_j) &= \sum_{i=1}^{m} \left[B_C(\mu_j)\,\text{sc}_N(\mu_j,\mu_i) + B_S(\mu_j)\,\text{ss}_N(\mu_j,\mu_i)\right] \end{aligned}\right\} \quad \text{for } j = 1,\ldots,m. \tag{62}$$

This set of $2m$ linear equations can always be solved for the $2m$ cosine and sine parts. If frequency μ_j is an edge frequency then $S_{\text{avg}}(\mu_j) = 0$ and that equation is removed from the set, and so on as in the case with two sinusoids.

Solving these equations numerically with LU decomposition is stable and quick. Frequency indices very close to an edge frequency are treated as edge frequencies, otherwise roundoff error in computing the suprods dominates the solution for the sine part at that frequency (whose absolute value blows up, often to an absurdly large number). A set of one or two hundred of these can be solved in about a second using VBA in an Excel spreadsheet, so it is quite practical to estimate hundreds of contained sinusoids at once.

9 The Manual Fourier Transform (MFT)

The manual Fourier transform (MFT) is similar to the discrete Fourier transform (DFT): both compute a spectrum of a regular time series, expressing the time series as a sum of sampled sinusoids. The MFT is a more general case of the DFT: in an MFT the user specifies the frequencies of the sampled sinusoids in the spectrum, while with a DFT the frequencies are predetermined.

Because the sampled sinusoids at the specified frequencies are generally not orthogonal under time summation, the MFT is generally not invertible: the sinusoids at the specified frequencies that best sum to the time series do not necessarily sum exactly to the time series. Thus, with the MFT, we cannot talk about a time series as being either “in the time domain” or “in the frequency domain” as we can with a DFT—because information may be lost if we use the MFT to move from the time domain to the frequency domain.

Consequently, a measure of how well the MFT synthesizes the time series from its spectrum of sinusoids is an important part of the MFT results. The measure we use here is the residue of the error time series, that is, of the original time series less the sum of the sinusoids found by the MFT (or inverse transform). The residue of a length-N time series g is defined by

$$\text{residue} = \sum_{\tau=0}^{N-1} \left| g[\tau] \right| . \tag{63}$$

Note that we have used absolute values rather than squares, which is analytically awkward but computationally more appropriate. We define the fractional error of an MFT of g as the residue of the error divided by the residue of g:

$$\text{fractional error} = \frac{\text{residue of } \left\{ g - MFT^{-1}\left[MFT(g) \right] \right\}}{\text{residue of } g} . \tag{64}$$

If the fractional error exceeds a few percent, something is wrong with the MFT computation or the specified frequencies are inappropriate, and the MFT spectrum should not be used. We usually express the fractional error as the error percentage, which is 100 times the fractional error. The notion of a goodness-of-fit parameter like the error percentage is absent from the DFT because the DFT is invertible: its fractional error is always zero.

9.1 MFT Frequencies

Suppose we assume that the sinusoids in our length-N time series g are at the m frequencies

$$f_1, f_2, \ldots, f_m ,$$

where the frequencies are non-negative and ordered: $f_i \geq 0$, and $f_i < f_j$ whenever $i < j$, for $i, j \in \{1, 2, \ldots, m\}$. These frequencies are our specified frequencies, from which the MFT will attempt to synthesize g. They are continuous-time frequencies. The corresponding frequency indices are

$$\nu_1, \nu_2, \ldots, \nu_m , \quad \nu_i = f_i E ,$$

where E is the extent of g. The term “frequency index” is carried over from the DFT, even though the frequency index ν is now a real variable rather than integral (which is rather incompatible with the notion of an index). And the actual index of the MFT frequencies, i and j above, go unnamed—but are free to be 1-indexed because they do not appear in the arguments of the trigonometric functions (as ν does for the DFT). Oh well.

The minimum positive frequency that is definitely satisfactory is the frequency whose period is the extent E of g, namely

$$f_{\min} = \frac{1}{E} .$$

Frequencies below this have less than a full cycle in the time series, and while they may be present in g, are harder and less reliable to estimate. We have to use them if g trends slowly, but they are not as desirable as sinusoids with at least one cycle in view.

The maximum meaningful frequency is half the average sampling frequency, which is the Nyquist frequency of the time series:

$$f_{\max} = \frac{N}{2E}. \tag{65}$$

A sinusoid at a frequency above the Nyquist frequency has identical samples to some sinusoid with a non-negative frequency below the limit—they are indistinguishable, given the sampling regime. The sinusoid above the Nyquist frequency is an "alias" of the one below, and it is redundant for the purposes of synthesizing *g*.

For comparison, the DFT expresses *g* as a sum of sampled sinusoids at the $\vartheta_C + 1$ frequencies

$$0, \frac{1}{E}, \ldots, \frac{\vartheta_C}{E}, \quad \text{where } \vartheta_C = \lfloor N/2 \rfloor. \tag{66}$$

The conversions between frequency *f* and time *t*, and frequency index ν and time index τ are

$$fE = \nu, \quad t\frac{N}{E} = \tau, \quad \text{and} \quad ft = \frac{\nu\tau}{N}. \tag{67}$$

9.2 MFTs for Real-Valued Time Series

Let *g* be real-valued length-*N* time series. Let the specified frequencies be $f_1, f_2, \ldots, f_m$, so the frequency indices are $\nu_1, \nu_2, \ldots, \nu_m$ where $\nu_i = f_i E$ for $i = 1, \ldots, m$, where *E* is the extent of *g*. Let the cosine and sine parts of the MFT of *g* be B_C and B_S respectively, each a real-valued function on the range of valid frequency indices $[0, N/2]$, defined (as per the DFT) as the multipliers of the cosine and sine sinusoids in the synthesis.

Synthesis:

$$g[\tau] = \sum_{i=1}^{m} \left\{ B_C(\nu_i)\cos\left(\frac{2\pi\nu_i\tau}{N}\right) + B_S(\nu_i)\sin\left(\frac{2\pi\nu_i\tau}{N}\right) \right\} \quad \text{for } \tau = 0, 1, \ldots, N-1. \tag{68}$$

Analysis:

B_C and B_S as determined by estimating the *m* contained sinusoids.
See Appendix 8. This involves solving a set of $2m$ linear equations.

Spectrum of *g*:

$$\left\{ B_C(\nu_i)\cos\left(\frac{2\pi\nu_i\tau}{N}\right) + B_S(\nu_i)\sin\left(\frac{2\pi\nu_i\tau}{N}\right), \; i = 1, \ldots, m \right\} \tag{69}$$

Amplitude spectrum of *g*:

$$\operatorname{amp}(\nu) = |B(\nu)| = \begin{cases} \sqrt{B_C^2(\nu_i) + B_S^2(\nu_i)} & \nu_i \in \{\nu_1, \ldots, \nu_m\} \\ 0 & \text{otherwise} \end{cases} \tag{70}$$

Phase spectrum of *g*:

$$\text{phase}(\nu) = \text{pha}\left[B_C(\nu), B_S(\nu)\right] = \begin{cases} \text{pha}\left[B_C(\nu_i), B_S(\nu_i)\right] & \nu_i \in \{\nu_1, \ldots, \nu_m\} \\ 0 & \text{otherwise} \end{cases} \quad (71)$$

9.3 Frequency Bracketing and the MFT

The frequency specification of the MFT can be "bracketed", in which case the specified frequencies are partitioned into "brackets". The MFT described to this point is a single-bracket MFT, with all the frequencies in the same bracket and all on an equal footing.

The multi-bracket MFT is performed as follows:

- Perform a single-bracket MFT on the time series using just the frequencies in the first bracket. Subtract the sum of the sinusoids thus found from the time series, to form a remaining time series.
- Perform a single-bracket MFT on this remaining time series using just the frequencies in the second bracket. Subtract the sum of the sinusoids thus found from the remaining time series, to form a new remaining time series.
- And so on, until a single-bracket MFT has been performed once for each bracket.

The spectrum found by the multi-bracket MFT is the union of all the sets of sinusoids found by the single-bracket MFTs.

The purpose of bracketing is to be able to specify that some frequencies are more important, or sinusoids at those frequencies expected to have much greater amplitudes, than frequencies in subsequent brackets. If all the frequencies were fitted at once in a single-bracket MFT, the guiding assumption would be that each frequency is on an equal basis, equally likely to have a larger-amplitude sinusoid. However if you expect some main sinusoids at some frequencies and some minor sinusoids at other frequencies, then you should use bracketing to guide the MFT towards the solution you are expecting.

10 The Optimal Fourier Transform (OFT)

The optimal Fourier transform (OFT) is similar to the discrete Fourier transform (DFT) and manual Fourier transform (MFT): each computes a spectrum of a regular time series, expressing the time series as a sum of sampled sinusoids. The OFT is optimal in the sense that it finds the sampled sinusoids in a time series, *including their frequencies*, automatically and accurately (usually, hopefully, depending on the implementation!).

- The OFT is more general than the DFT: an OFT discovers which frequencies are appropriate, while a DFT is constrained to using predetermined frequencies.
- The OFT is more than the MFT: the OFT discovers its own frequencies, then executes an MFT using those frequencies.

The OFT is useful when trying to best estimate the sinusoids in a time series, and in considering frequencies other than those used by the DFT.

Because the sampled sinusoids at the frequencies discovered by the OFT are generally not orthogonal under time summation, the OFT is generally not invertible: the sinusoids in the time series at the discovered frequencies do not necessarily sum exactly to the time series. Thus, with the OFT as with the MFT, we cannot talk about a time series as being either "in the time domain" or "in the frequency domain" as we can with a DFT—because information is often lost if we use the OFT to move from the time domain to the frequency domain.

As with the MFT, a measure of how well the OFT synthesizes the time series from its spectrum of sinusoids is an important part of the OFT results. As per the MFT, we use the fractional error based on residues of absolute values of time series, the residue of the pointwise errors as a fraction of the original time series.

In the absence of pointwise noise (noise that strikes a data point or just the data point and its neighbors independently of what happens at other data points), the OFT usually fits any vaguely smooth time series with far fewer sinusoids than a DFT—with fractional errors of just a few parts per million. Pointwise noise is obviously not amenable to being represented as the sum of a few sampled sinusoids, so when pointwise noise is present the fractional error is much higher. The DFT always has zero fractional error, but it handles the pointwise noise in the time series by modeling it using many high frequency sinusoids—thus transporting the noise to the frequency domain. Depending on your application, it may be better to use an OFT that incurs a higher fractional error but leaves much of the noise behind in the time domain.

If the fractional error of an OFT exceeds a few percent, check its error time series (the difference between the original time series and the time series synthesized from its OFT). If it contains just pointwise noise, then the OFT is good. If the error appears to contain structure or signal, then something has gone wrong with the OFT. If pointwise noise is low the OFT often scores fractional errors of less than one percent, in which case the OFT is very close to being invertible.

10.1 OFT Frequencies

The OFT uses only non-negative frequencies up to the Nyquist frequency (where aliasing and redundancy set in), that is, frequencies in $\left[0, N/2E\right]$ and frequency indices in $\left[0, N/2\right]$, for a length-*N* extent-*E* time series.

10.2 OFTs for Real-Valued Time Series

Let *g* be a real-valued length-*N* extent-*E* time series.

Let the OFT discover *m* frequencies in *g*: there is a sinusoid with non-zero amplitude in the OFT's synthesis of *g* at each of the frequencies $f_1, f_2, \ldots, f_m$ or frequency indices $\nu_1, \nu_2, \ldots, \nu_m$ (where $\nu_i = f_i E$ for $i = 1, \ldots, m$).

Let the cosine and sine parts of the OFT of *g* be B_C and B_S respectively, each a real-valued function on the range of valid frequency indices $\left[0, N/2\right]$, defined (as per the DFT and MFT) as the multipliers of the cosine and sine sinusoids in the synthesis.

Synthesis:

$$g[\tau] = \sum_{i=1}^{m} \left\{ B_C(\nu_i) \cos\left(\frac{2\pi\nu_i\tau}{N}\right) + B_S(\nu_i) \sin\left(\frac{2\pi\nu_i\tau}{N}\right) \right\} \quad \text{for } \tau = 0, 1, \ldots, N-1. \qquad (72)$$

Analysis:

B_C and B_S as determined by the implementation, see the next section.

Spectrum of *g*:

$$\left\{ B_C(\nu_i)\cos\left(\frac{2\pi\nu_i\tau}{N}\right) + B_S(\nu_i)\sin\left(\frac{2\pi\nu_i\tau}{N}\right),\ i=1,\ldots,m \right\} \tag{73}$$

Amplitude spectrum of *g*:

$$\mathrm{amp}(\nu) = |B(\nu)| = \begin{cases} \sqrt{B_C^2(\nu_i) + B_S^2(\nu_i)} & \nu_i \in \{\nu_1,\ldots,\nu_m\} \\ 0 & \text{otherwise} \end{cases} \tag{74}$$

Phase spectrum of *g*:

$$\mathrm{phase}(\nu) = \mathrm{pha}\left[B_C(\nu), B_S(\nu)\right] = \begin{cases} \mathrm{pha}\left[B_C(\nu_i), B_S(\nu_i)\right] & \nu_i \in \{\nu_1,\ldots,\nu_m\} \\ 0 & \text{otherwise} \end{cases} \tag{75}$$

10.3 Implementation

The basic question arises: how do we know which frequencies are present in the time series? The cosine and sine averages are perhaps our only obvious tools for peering into the spectral structure of the time series. Unfortunately the most obvious and direct approach, using local maxima in the amplitude of the cosine and sine averages as frequency is varied, is of such low resolution as to be almost useless. Consider a time series with *m* frequencies:

$$g[\tau] = \sum_{i=1}^{m} A_i \cos\left(\frac{2\pi\mu_i\tau}{N} - \phi_i\right), \text{ for } \tau = 0,1,\ldots N-1. \tag{76}$$

The cosine average is

$$C_{\text{avg}}(\nu) = \sum_{i=1}^{m}\left[A_i\cos(\phi_i)\,\mathrm{cc}_N(\nu,\mu_i) + A_i\sin(\phi_i)\,\mathrm{cs}_N(\nu,\mu_i)\right], \quad \nu \in \left[0, N/2\right]. \tag{77}$$

But the suprod function cc has a broad peak of width in the order of half a unit of frequency index. Thus frequencies less than about two thirds of a unit of frequency index cannot be distinguished by a local maximum in the cosine average; they combine into a single maximum. Also, all the suprod functions go up and down a few times as the frequency index distance increases, so contributions from sinusoids more distant in frequency may be increasing or decreasing at any frequency—leading to many local maxima in the cosine average that are not near any frequencies present in the time series. Finally, due to a combination of both factors, peaks in the amplitude of the cosine averages are often up to half a unit of frequency index from the frequency index of the corresponding sinusoid in the time series.

The core concept of the OFT is to minimize the m-function, a function which:

- Takes a set of frequency indices as an argument.
- Estimates the sampled sinusoids in the time series using those frequencies.

- Returns the residue in the error time series, the original time series less the sum of the sampled sinusoids just found.

The exact algorithm of an OFT is implementation dependent, being affected by issues such as the maximum number of sinusoids considered at once at various stages, the tolerances and number of iterations in the minimizations, the minimization algorithm used, the method for calculating residue, or the width of the frequency-index bands around the edge frequencies within which a frequency index is considered an edge frequency. The OFT algorithm is thus not well-defined (in contrast, the DFT is well defined and the MFT almost is).

Here we perform an OFT in three parts:

1. Reconnaissance – Roughly estimate an initial set of frequencies.
2. Main – Compute the frequencies of any contained sinusoids precisely.
3. MFT – Perform an MFT using the precise frequencies.

The reconnaissance part begins with a DFT of the original time series. The frequencies of the local peaks in the DFT amplitude spectrum become the first set of guessed frequencies. The m-function is minimized, starting with this first set, using a multivariable function minimization algorithm that can vary each of the frequencies in the guess. The cosine and sine parts of the contained sinusoids at the precise frequencies thus discovered are estimated, and these are subtracted from the original time series to form a remaining time series. This is repeated a few times, each loop starting with a DFT of the working time series to guess some initial frequencies, minimizing the m-function to get some precise frequencies, estimating the cosine and sine parts for these precise frequencies, then subtracting these sinusoids from the working time series. Lastly, all the precise frequencies thus found are put in a single set, near-duplicates are consolidated, and the half a dozen or so frequencies whose sinusoids had the highest amplitudes form the frequencies found by the reconnaissance stage.

The main part starts with the reconnaissance stage frequencies as a first guess, and minimizes the m-function (as in the reconnaissance stage) to form precise estimates of contained frequencies. Any similar frequencies are consolidated, because it may happen that two or more frequencies converge towards the same frequency during the minimization. The minimization is repeated if there were any consolidations. The cosine and sine parts for these precise frequencies are estimated, recorded, and subtracted off the original time series to form the working time series. Repeat this procedure until the residue of the working time series is sufficiently low, with the initial guess after the first loop coming from local peaks in the DFT amplitudes. Lastly, take all the frequencies thus found, sort them by amplitude, and form them into brackets based on clumping amplitudes by orders of magnitude.

The MFT part simply performs an MFT on the original time series, using the bracketed frequency specification of the main stage. The spectrum thus computed is the OFT spectrum.

The local peaks in DFT amplitude will often not discover contained sinusoids that are relatively close in frequency, so the reconnaissance part is simply aimed at uncovering such frequencies.

Finally, the result of the process described so far is sometimes not finished, because the error time series (the difference between original time series and the time series synthesized from its OFT) has obvious structure, more than just pointwise noise. So another stage of OFT is applied, with just a minimal number of sinusoids (say 15), to capture this residual structure. This is repeated until the error is apparently just pointwise noise. To determine if the error time series is just pointwise noise, it would be best to apply the Ljung-Box test. However we use a quick and dirty method— just to do another stage of OFT and see if the spectrum is sufficiently flat. If it is not flat then the stage would be required in any case, to determine the residual sinusoids, so in effect it is quick. If the spectrum of the speculative OFT stage is flat, we didn't need those sinusoids and we throw them away and quit.

Computing times and machine speeds limit the number of frequencies considered at various stages. In *Climate.xlsm*, in VBA in an Excel spreadsheet, the reconnaissance stage considers five frequencies at once, the main stage ten frequencies at once ($m \le 10$), and the MFT 150 frequencies at once. As more frequencies are considered at once, the estimation of contained sinusoids improves. While simple and fairly ubiquitous, VBA is not notably fast or efficient. In a faster computing environment we could use a much higher *m*. Whereas DFTs of the climate datasets compute in less than a second, the OFTs take several minutes, even up to two hours, so we batched and cached them.

The OFT orders its spectral sinusoids by amplitude, the sinusoids with the largest amplitudes first (they are presumably the more important sinusoids). The sum of the sinusoids in the spectrum estimated by the OFT will more closely approximate the time series as it uses more sinusoids. If there is pointwise noise in the time series and the signal is larger than the pointwise noise, at some number of sinusoids the additional smaller sinusoids will just be recreating pointwise noise rather than signal and the OFT's spectrum can be usefully truncated at that number. See the climate paper mentioned in the *Administration* section at the top for an example of applying this principle to the OFT of the HadCrut4 temperature dataset—the first 30 or so sinusoids found by the OFT appear to be mainly signal, but after that appear to be mainly representing noise.

11 Examples of the OFT

Example 1: One sinusoid. Consider a time series sampled from the continuous-time function

$$g(t) = 9\cos\left(\frac{2\pi}{208}t - 20\frac{2\pi}{360}\right), \quad t \in \mathbb{R}, \tag{78}$$

which is sinusoidal with period 208 years, amplitude 9, and phase 20°. Let the time series be regular, with 300 samples over 2,000 years starting at time 0. The time series is thus sinusoidal, but as it happens it is not at one of the predetermined frequencies of the DFT (the nearest of which are at periods 9/2000 =222.22 years and 10/2000 = 200.00 years). The DFT, which took 1/20th of a second to compute in *Climate.xlsm*, estimates *g* as

$$
\begin{aligned}
g_{\text{DFT}}(t) = & 0.865\cos\left(\frac{2\pi}{285.71}t - 272.66\frac{2\pi}{360}\right) + 1.508\cos\left(\frac{2\pi}{250.00}t - 271.21\frac{2\pi}{360}\right) \\
& +4.210\cos\left(\frac{2\pi}{222.22}t - 269.94\frac{2\pi}{360}\right) + 7.100\cos\left(\frac{2\pi}{200.00}t - 88.81\frac{2\pi}{360}\right) \\
& +2.064\cos\left(\frac{2\pi}{181.82}t - 87.77\frac{2\pi}{360}\right) + 1.247\cos\left(\frac{2\pi}{166.67}t - 86.80\frac{2\pi}{360}\right) + \ldots
\end{aligned} \tag{79}
$$

The spectrum estimated by the DFT contains 150 non-zero sinusoids, at all 150 of the pre-determined frequencies; Eq. (79) shows only the six with the closest frequencies to the true frequency, which also tend to be the ones with the largest amplitudes. The OFT, which took a third of a second to compute, estimated g as

$$
g_{\text{OFT}}(t) = 9\cos\left(\frac{2\pi}{208.00}t - 20.00\frac{2\pi}{360}\right). \tag{80}
$$

The OFT locked onto the correct frequency and estimated the signal correctly. □

Example 2: Four sinusoids. Let the continuous-time function

$$
\begin{aligned}
g(t) = \; & 11\cos\left(\frac{2\pi}{606}t - 235\frac{2\pi}{360}\right) + 8\cos\left(\frac{2\pi}{303}t\right) \\
& + 13\cos\left(\frac{2\pi}{23}t - 215\frac{2\pi}{360}\right) + 9\cos\left(\frac{2\pi}{202}t - 20\frac{2\pi}{360}\right)
\end{aligned} \tag{81}
$$

be sampled 300 times at regular intervals over 2,000 years starting at 0.

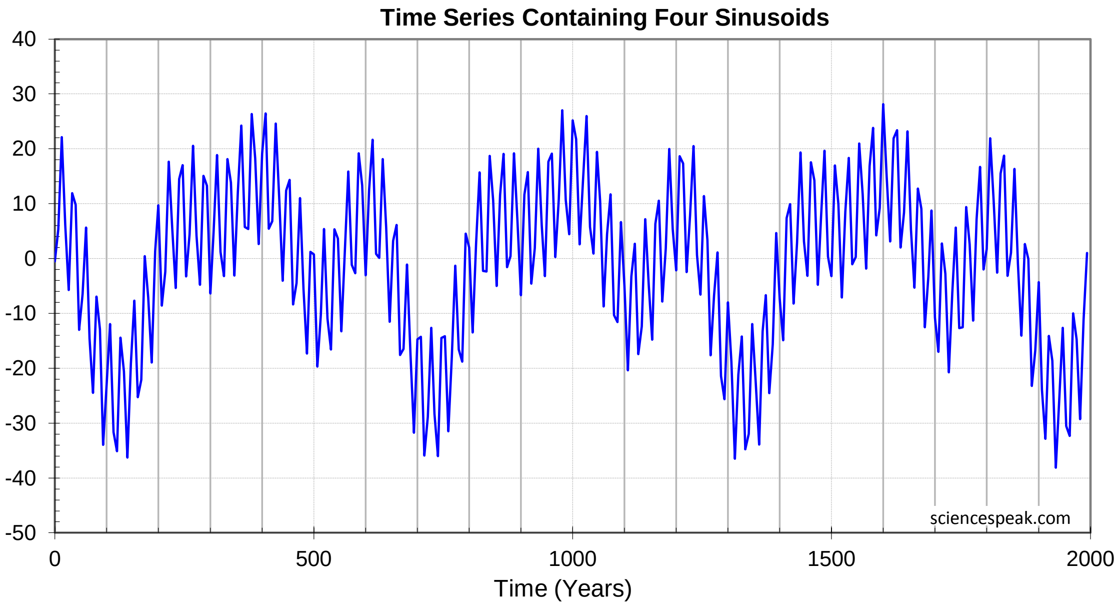


Figure 4: The time series sampled from the function in Eq. (81), containing four sinusoids.

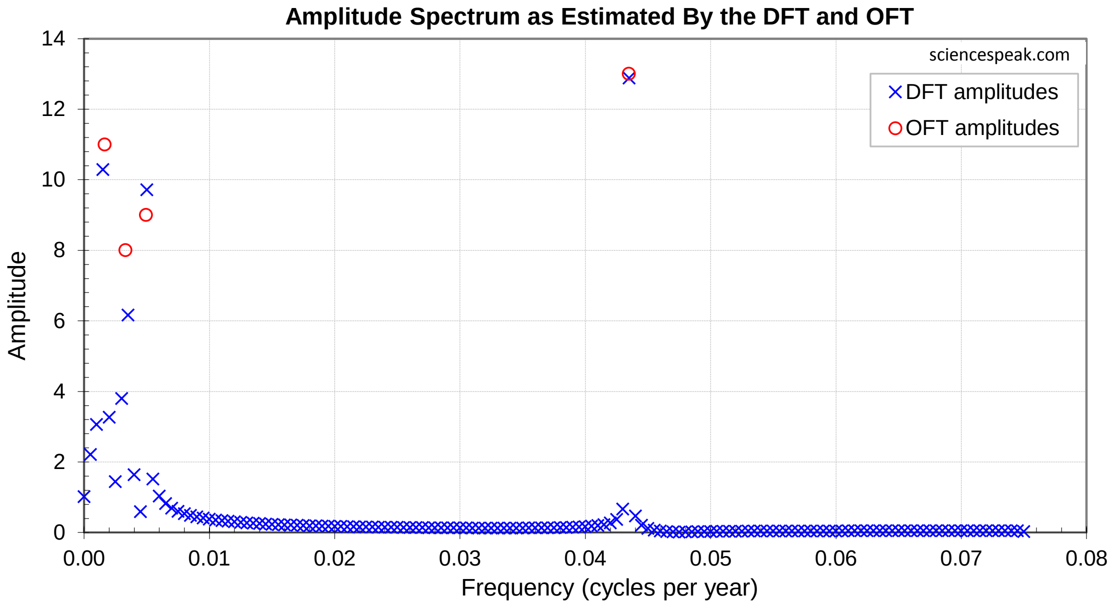


Figure 5: The DFT and OFT of the time series in Fig. 4 (amplitudes only).

The DFT, which took a twentieth of a second to compute, finds a spectrum consisting of 150 non-zero sinusoids, with the largest-amplitude ones near the frequencies in g. The OFT, which took 1.3 seconds to compute, finds the four frequencies in g exactly, then correctly estimates the amplitudes and phases of the four sinusoids in g. □

Example 3: Ten sinusoids. Our implementation of the OFT in *Climate.xlsm* can find up to ten sinusoids at once, so consider a continuous-time function which is the sum of ten sinusoids:

$$\begin{aligned} g(t) = \quad & 11\cos\left(\frac{2\pi}{606}t - 45\frac{2\pi}{360}\right) && +10\cos\left(\frac{2\pi}{404}t\right) \\ & +9\cos\left(\frac{2\pi}{303}t\right) && +9\cos\left(\frac{2\pi}{202}t - 20\frac{2\pi}{360}\right) \\ & +9\cos\left(\frac{2\pi}{153}t - 150\frac{2\pi}{360}\right) && +8\cos\left(\frac{2\pi}{101}t - 15\frac{2\pi}{360}\right) \\ & +8\cos\left(\frac{2\pi}{75}t - 300\frac{2\pi}{360}\right) && +4\cos\left(\frac{2\pi}{49}t - 340\frac{2\pi}{360}\right) \\ & +8\cos\left(\frac{2\pi}{23}t - 215\frac{2\pi}{360}\right) && +7\cos\left(\frac{2\pi}{14}t - 40\frac{2\pi}{360}\right). \end{aligned} \tag{82}$$

Let our time series be g sampled 300 times at regular intervals over 2,000 years starting at 0.

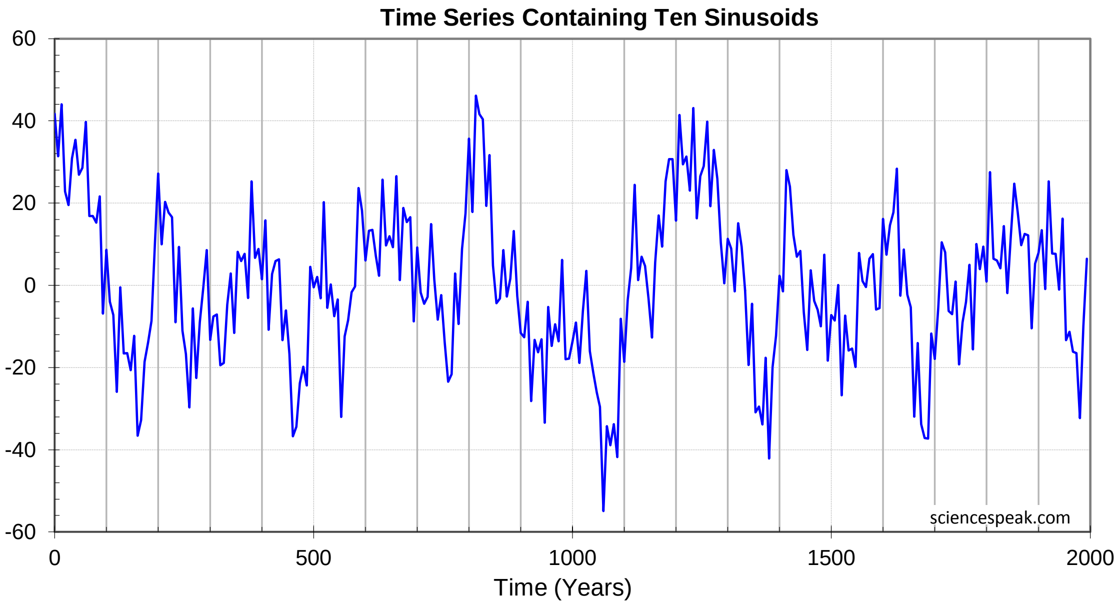


Figure 6: The time series sampled from the function in Eq. (82), containing ten sinusoids.

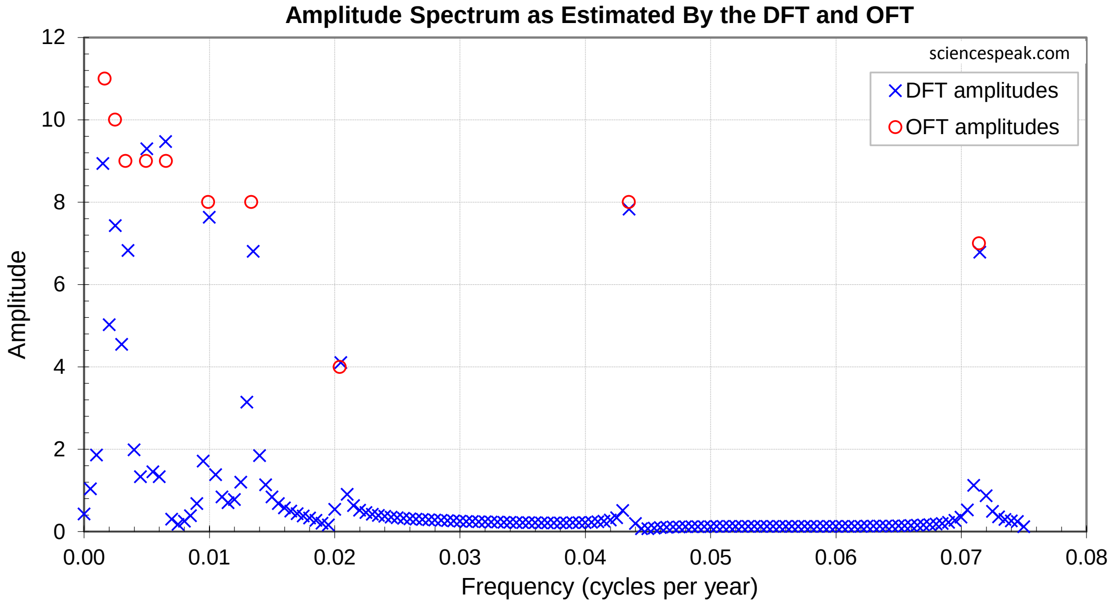


Figure 7: The DFT and OFT of the time series in Fig. 6 (amplitudes only).

As in the last two examples, the DFT took a twentieth of a second to compute and finds a spectrum consisting of 150 non-zero sinusoids, with the largest-amplitude ones near the frequencies in g. The OFT took 40 seconds to compute, and the spectrum it found was:

OFT			
Detected frequencies	Detected periods	AP of time series	
f	1 / f	amp	phase
cycles / yr	years	U	degrees
0.00165	606.00	11.0000	45.00
0.00248	404.00	10.0000	0.00
0.00495	202.00	9.0000	20.00
0.00654	153.00	9.0000	150.00
0.00330	303.00	9.0000	0.00
0.00990	101.00	8.0000	15.00
0.01333	75.00	8.0000	300.00
0.04348	23.00	8.0000	215.00
0.07143	14.00	7.0000	40.00
0.02041	49.00	4.0000	340.00

Figure 8: Estimate of the spectrum of the time series in Fig. 6, by the OFT in *Climate.xlsm*.

The OFT locked on to the ten frequencies simultaneously and correctly estimated the frequencies, amplitudes and phases of the ten sinusoids in g. □

Example 4: Two sinusoids close in frequency, resolvable. Our implementation of the OFT in *Climate.xlsm* can distinguish frequencies that are more than about the Nyquist frequency imposed by the sampling. Let the continuous-time function

$$g(t) = 8\cos\left(\frac{2\pi}{303}t\right) + 9\cos\left(\frac{2\pi}{289}t - 20\frac{2\pi}{360}\right) \quad (83)$$

be sampled 300 times at regular intervals over 2,000 years starting at 0. The time between samples is 2000 / 300 or 6.667 years, so the Nyquist period is twice that or 13.334 years. The difference in periods between our two sinusoids in Eq. (83) is 303 – 289 or 14 years, just above the Nyquist period. (ht: Greg Goodman pointed out the Nyquist connection.)

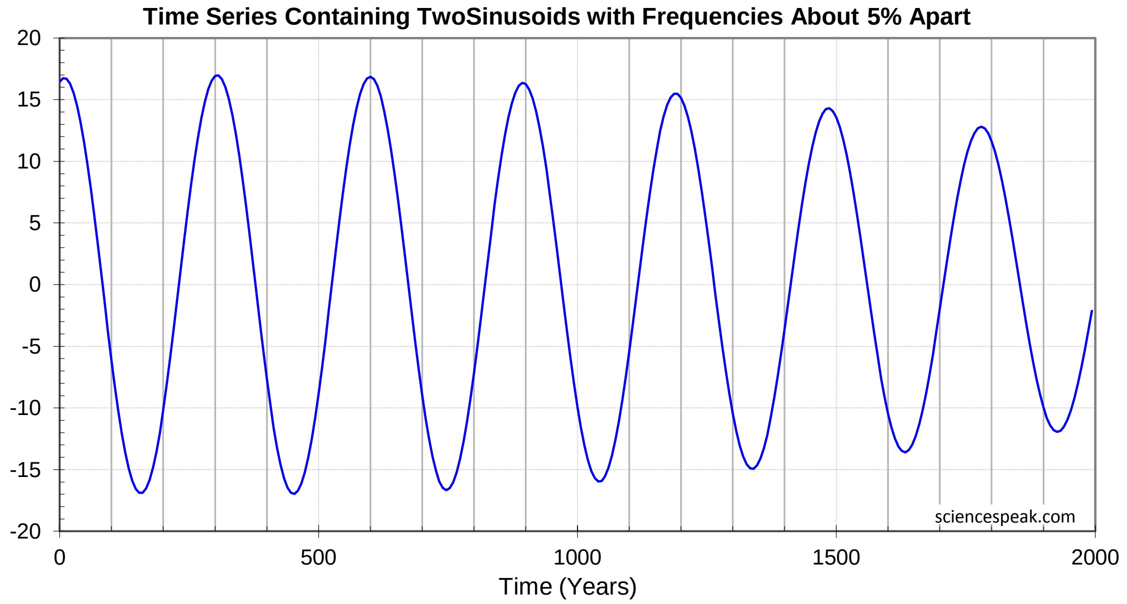


Figure 9: The time series sampled from the function in Eq. (83), where the two sinusoids are a slightly further apart in frequency than the Nyquist frequency. Resolvable by the OFT into two sinusoids.

OFT			
Detected frequencies	*Detected periods*	*AP of time series*	
f	*1 / f*	*amp*	*phase*
cycles / yr	*years*	*U*	*degrees*
0.00346	289.00	9.0000	20.00
0.00330	303.00	8.0000	360.00

Figure 10: Estimate of the spectrum of the time series in Fig. 9, by the OFT in *Climate.xlsm*.

The OFT in *Climate.xlsm* took two seconds to correctly find the two sinusoids and their amplitudes and phases. □

Example 5: Two sinusoids close in frequency, not resolvable. Our implementation of the OFT in *Climate.xlsm* fails to distinguish frequencies that differ by about the Nyquist frequency imposed by the sampling. Let the continuous-time function

$$g(t) = 8\cos\left(\frac{2\pi}{303}t\right) + 9\cos\left(\frac{2\pi}{291}t - 20\frac{2\pi}{360}\right) \tag{84}$$

be sampled 300 times at regular intervals over 2,000 years starting at 0. The time between samples is 2000 / 300 or 6.667 years, so the Nyquist period is twice that or 13.334 years. The difference in periods between our two sinusoids in Eq. (84) is 303 – 291 or 12 years, just below the Nyquist period.

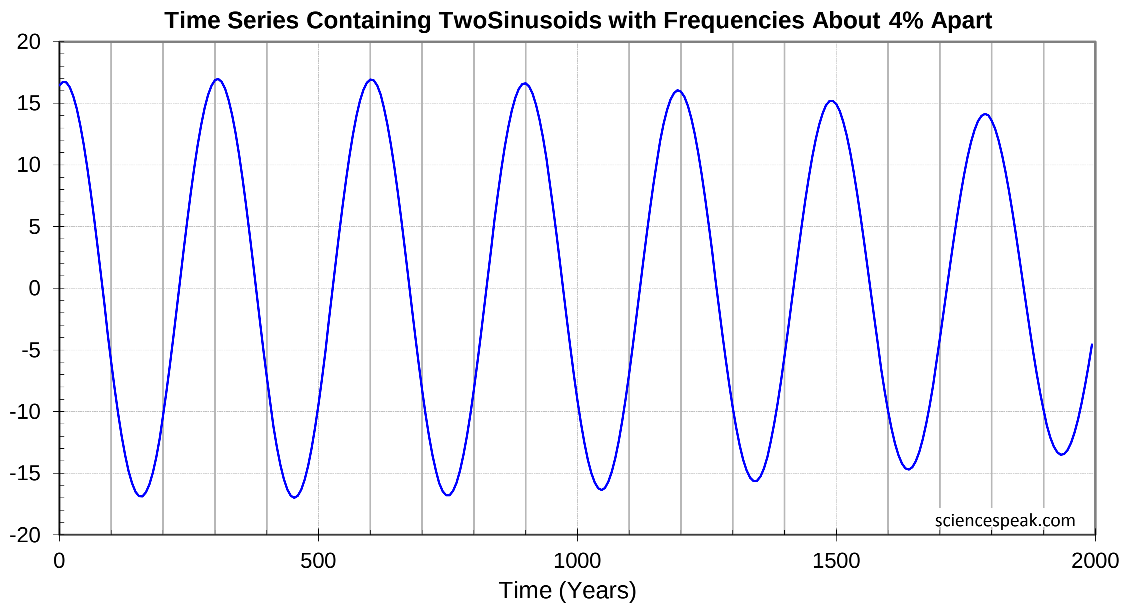


Figure 11: The time series sampled from the function in Eq. (84), where the two sinusoids are a slightly closer in frequency than the Nyquist frequency. Not resolvable by the OFT into two sinusoids.

OFT			
Detected frequencies	Detected periods	AP of time series	
f	1 / f	amp	phase
cycles / yr	years	U	degrees
0.00337	296.47	15.9593	10.66
0.00303	330.35	0.8419	321.89
0.00367	272.69	0.6502	47.91
0.00275	364.24	0.3119	93.56
0.00396	252.24	0.1603	251.39
0.00334	299.47	0.1526	183.08
0.00231	432.64	0.1151	154.67
0.00200	499.76	0.0915	257.82
0.00423	236.65	0.0681	106.96
0.00245	408.07	0.0454	242.01

Figure 12: The first 10 sinusoids of the estimate of the spectrum of the time series in Fig. 11, by the OFT in *Climate.xlsm*. The OFT lists the sinusoids by amplitude.

The OFT in *Climate.xlsm* took 90 seconds but failed to lock onto and find the two frequencies. Instead it estimated the spectrum as a single sinusoid, with frequency and phase about midway between the two actual sinusoids and amplitude roughly equal to the sum of the actual amplitudes, plus 122 other sinusoids mostly with tiny amplitudes. □

12 The Irregular MFT (iMFT) and the Irregular OFT (iOFT)

Some time series, such as temperature datasets going back beyond the last few hundred years, are irregular—meaning that the times between adjacent data points are not all the same. The DFT, MFT, and OFT are only for regular time series. The DFT relies on orthogonal sampled sinusoids which in turn rely on regular spacing of data points in time, so is inherently regular. But the MFT and OFT can be adapted to irregular time series.

12.1 Irregular Time Series

Consider a length-*N* irregular time series *g*, whose data points are at times

$$t_0, t_1, \ldots, t_{N-1}$$

where the times are ordered: $t_i < t_j$ whenever $i < j$, for $i, j \in \{0, 1, \ldots, N-1\}$. The extent *E* of *g* is best estimated, in the absence of further information, by

$$E \approx \frac{N}{N-1}(t_{N-1} - t_0). \tag{85}$$

In the framework here, a regular time series is a special case of an irregular time series.

We wish to reuse the software and formulae for regular time series with as little alteration as possible, so we wish to use frequency indices rather than frequencies when analyzing irregular time series. With regular time series, the time summations vary a time index τ from 0 to $N-1$, and the arguments of the cosine and sine functions are $2\pi\nu\tau/N$. We wish to keep everything except the τ in those arguments, and we need to replace τ by something containing the information about the irregular times $t_0, t_1, \ldots, t_{N-1}$. We therefore need a new time variable

t'_τ in the arguments, indexed by the time index τ and varying over the range $[0, N)$ like the time index τ with the regular time series. The irregular times in fact vary over $[t_0, t_0 + E)$, a fact we now use to gives us the required scaling and offset.

Let the normalized-time discrete variable be defined as

$$t'_\tau = \frac{N}{E}(t_\tau - t_0), \quad \text{for } \tau = 0,1,\ldots N-1. \tag{86}$$

Thus

$$\begin{aligned} t'_0 &= 0 \\ t'_1 &= \frac{N}{E}(t_1 - t_0) \\ t'_2 &= \frac{N}{E}(t_2 - t_0) \\ &\vdots \\ t'_{N-1} &= \frac{N}{E}(t_{N-1} - t_0). \end{aligned}$$

Because the extent is just N times the average distances between adjacent data points, the distance between k points, $t_{k+j} - t_j$, is equal on average to kE/N. So if the distances between adjacent points are all about equal to the average such distance, then

$$t'_k = \frac{N}{E}(t_k - t_0) \simeq k, \quad \text{for } k = 0,1,\ldots N-1. \tag{87}$$

Thus:

- The normalized-time between adjacent data points is one, on average.
- The extent of the time series is N units of normalized-time.

For notational brevity, let the time vector be

$$\mathbf{t} = t_0, t_1, \ldots, t_{N-1}.$$

The normalized time vector is

$$\mathbf{t}' = t'_0, t'_1, \ldots, t'_{N-1} = 0, \frac{N}{E}(t_1 - t_0), \ldots, \frac{N}{E}(t_{N-1} - t_0).$$

In the special case that the time series is regular, its time vector is

$$\mathbf{t} = t_0, t_1, \ldots, t_{N-1} = t_0, t_0 + \frac{E}{N}, t_0 + \frac{2E}{N}, \ldots, t_0 + \frac{(N-1)E}{N} \tag{88}$$

and its normalized time vector is therefore

$$\mathbf{t}' = t'_0, t'_1, \ldots, t'_{N-1} = 0, 1, \ldots, N-1.$$

Therefore, to adapt the regular formula and software to irregular time series:

- Change the argument of the cosine and sine functions from $2\pi\nu\tau/N$ to $2\pi\nu t'_\tau/N$.
- Everything else is the same.

The regular time series then become a special case, in which

$$t'_\tau = \tau .$$

12.2 The Irregular Suprod Functions

In section 6 we introduced the regular suprod functions and explored some of their properties. Now we need suprods for irregular time series.

The irregular suprod functions for any positive integer *N*, time vector $\mathbf{t} = t_0, t_1, \ldots, t_{N-1}$, and real numbers ν and μ, are defined by

$$\begin{aligned}
\mathrm{icc}_{N,\mathbf{t}}(\nu,\mu) &= N^{-1}\sum_{\tau=0}^{N-1}\cos\left(\frac{2\pi\nu t'_\tau}{N}\right)\cos\left(\frac{2\pi\mu t'_\tau}{N}\right)\\
\mathrm{ics}_{N,\mathbf{t}}(\nu,\mu) &= N^{-1}\sum_{\tau=0}^{N-1}\cos\left(\frac{2\pi\nu t'_\tau}{N}\right)\sin\left(\frac{2\pi\mu t'_\tau}{N}\right)\\
\mathrm{isc}_{N,\mathbf{t}}(\nu,\mu) &= N^{-1}\sum_{\tau=0}^{N-1}\sin\left(\frac{2\pi\nu t'_\tau}{N}\right)\cos\left(\frac{2\pi\mu t'_\tau}{N}\right) = \mathrm{ics}_{N,\mathbf{t}}(\mu,\nu)\\
\mathrm{iss}_{N,\mathbf{t}}(\nu,\mu) &= N^{-1}\sum_{\tau=0}^{N-1}\sin\left(\frac{2\pi\nu t'_\tau}{N}\right)\sin\left(\frac{2\pi\mu t'_\tau}{N}\right).
\end{aligned} \tag{89}$$

The irregular suprods have to be computed numerically, because all of the relationships developed for regular suprods fail for irregular suprods. In particular the divide and conquer strategy for fast suprods fails, so the irregular suprods apparently have to be computed by the naïve and slow algorithm (that is, by directly following the defining formulae above).

12.3 Estimating Contained Sinusoids in an Irregular Time Series

In section 7 we estimated the contained sinusoids in a regular time series. Now we do the same for an irregular time series.

Suppose an irregular length-*N* extent-*E* time series *g* with time vector **t** contains one sinusoid at frequency index $\mu \in [0, N/2]$:

$$g[\tau] = A\cos\left(\frac{2\pi\mu t'_\tau}{N} - \phi\right) = A\cos(\phi)\cos\left(\frac{2\pi\mu t'_\tau}{N}\right) + A\sin(\phi)\sin\left(\frac{2\pi\mu t'_\tau}{N}\right), \quad \tau = 0,1,\ldots N-1. \tag{90}$$

The cosine and sine parts of *g* are clearly the same as if *g* was regular:

$$\left.\begin{aligned}
B_C(\nu) &= A\cos(\phi) I_{\nu=\mu}\\
B_S(\nu) &= A\sin(\phi) I_{\nu=\mu} I_{\nu\in(0,N/2)}
\end{aligned}\right\}\quad \text{for } \nu \in [0, N/2]. \tag{91}$$

But the cosine and sine averages are different, due to the sampling at different times:

$$\left.\begin{aligned} C_{\text{avg}}(\nu) &= N^{-1}\sum_{\tau=0}^{N-1} g[\tau]\cos\left(\frac{2\pi\nu t'_\tau}{N}\right) \\ S_{\text{avg}}(\nu) &= N^{-1}\sum_{\tau=0}^{N-1} g[\tau]\sin\left(\frac{2\pi\nu t'_\tau}{N}\right) \end{aligned}\right\} \quad \text{for } \nu \in [0, N/2]. \tag{92}$$

Substituting for g and expanding reveals that they are now the same linear combinations but of the irregular suprod functions instead of the suprod functions:

$$\left.\begin{aligned} C_{\text{avg}}(\nu) &= A\cos(\phi)\operatorname{icc}_{N,\mathbf{t}}(\nu,\mu) + A\sin(\phi)\operatorname{ics}_{N,\mathbf{t}}(\nu,\mu) \\ S_{\text{avg}}(\nu) &= A\cos(\phi)\operatorname{isc}_{N,\mathbf{t}}(\nu,\mu) + A\sin(\phi)\operatorname{iss}_{N,\mathbf{t}}(\nu,\mu) \end{aligned}\right\} \quad \text{for } \nu \in [0, N/2]. \tag{93}$$

In particular when $\nu = \mu$, and recognizing the cosine and sine parts:

$$\begin{aligned} C_{\text{avg}}(\mu) &= B_C(\mu)\operatorname{icc}_{N,\mathbf{t}}(\mu,\mu) + B_S(\mu)\operatorname{ics}_{N,\mathbf{t}}(\mu,\mu) \\ S_{\text{avg}}(\mu) &= B_C(\mu)\operatorname{isc}_{N,\mathbf{t}}(\mu,\mu) + B_S(\mu)\operatorname{iss}_{N,\mathbf{t}}(\mu,\mu). \end{aligned} \tag{94}$$

This is the same set of equations to solve for the cosine and sine parts of g as in the case when g was regular, except that irregular suprods replace the regular suprods.

Similarly for when g contains multiple sinusoids: the cosine and sine averages are the same linear combinations as in the regular case, and the set of equations in the cosine and sine parts is the same as in the regular case, except we use irregular suprods instead of regular suprods.

12.4 The iMFT and iOFT

The iMFT and iOFT are the same as the MFT and OFT, except:

- Replace regular suprods with irregular suprods (which use the time vector of the irregular time series).
- To compute cosine and sine averages, change the argument of the cosine and sine functions from $2\pi\nu\tau/N$ to $2\pi\nu t'_\tau/N$.

Appendix A Special Functions

The following functions are used here but are not standard.

A.1 Indicator function

From the set of all propositions to 0 and 1:

$$I_{\text{proposition}} = \begin{cases} 1 & \text{the proposition is true} \\ 0 & \text{the proposition is false}. \end{cases} \tag{95}$$

For example, for some integer N,

$$5 + 3I_{N \text{ is even}} = \begin{cases} 8 & \text{if } N \text{ is even} \\ 5 & \text{if } N \text{ is odd}. \end{cases}$$

A.2 Phase function

Arctan needs extending to be able to compute polar-coordinate angles, for which we use the phase function pha (pronounced "far"). It gives the angle on a plane, in radians in $[0, 2\pi)$, that the point (x, y) makes with the x-axis:

$$\text{pha}(x, y) = \left[\tan^{-1}(y/x) + \pi\, I_{x<0}\right] \bmod 2\pi\,, \qquad x, y \in \mathbb{R}\,. \tag{96}$$

If (x, y) is in the first quadrant, the phase function simplifies to

$$\text{pha}(x, y) = \tan^{-1}(y/x)\,.$$

For example, $\text{pha}(1,0) = 0$, $\text{pha}(1,\sqrt{3}) = \pi/3$, $\text{pha}(0,1) = \pi/2$, and $\text{pha}(-1,0) = \pi$.

A similar function is the two-argument arctangent function atan2, but its range is $(-\pi, \pi]$.