Fourier Transforms

1 Fourier Series

A periodic function $f(x)$ defined on a finite interval $0 \leq x \leq L$ can be written as a Fourier series:

$$f(x) = \sum_{k=0}^{\infty} \alpha_k \cos\left(\frac{2\pi kx}{L}\right) + \sum_{k=1}^{\infty} \beta_k \sin\left(\frac{2\pi kx}{L}\right) \tag{1}$$

where the α_k and β_k are coefficients that depend on the shape of the function. Recalling that

$$e^{\pm i\theta} = \cos\theta \pm i\sin\theta \tag{2}$$

we see that sine and cosine can be written as (recall that $1/i = -i$):

$$\cos\theta = \frac{1}{2}\left(e^{i\theta} + e^{-i\theta}\right) \tag{3}$$

$$\sin\theta = \frac{1}{2}i\left(e^{-i\theta} - e^{i\theta}\right) \tag{4}$$

Given these identities, we can rewrite the Fourier series as:

$$\begin{aligned}
f(x) &= \frac{1}{2}\sum_{k=0}^{\infty} \alpha_k \left[\exp\left(i\frac{2\pi kx}{L}\right) + \exp\left(-i\frac{2\pi kx}{L}\right)\right] + \frac{i}{2}\sum_{k=1}^{\infty} \beta_k \left[\exp\left(-i\frac{2\pi kx}{L}\right) - \exp\left(i\frac{2\pi kx}{L}\right)\right] \\
&= \alpha_0 + \frac{1}{2}\sum_{k=1}^{\infty}\left[\alpha_k \exp\left(i\frac{2\pi kx}{L}\right) + \alpha_k \exp\left(-i\frac{2\pi kx}{L}\right) + i\beta_k \exp\left(-i\frac{2\pi kx}{L}\right) - i\beta_k \exp\left(i\frac{2\pi kx}{L}\right)\right] \\
&= \alpha_0 + \frac{1}{2}\sum_{k=1}^{\infty}\left[(\alpha_k - i\beta_k)\exp\left(i\frac{2\pi kx}{L}\right) + (\alpha_k + i\beta_k)\exp\left(-i\frac{2\pi kx}{L}\right)\right] \\
&= \alpha_0 + \frac{1}{2}\sum_{k=1}^{\infty}(\alpha_k - i\beta_k)\exp\left(i\frac{2\pi kx}{L}\right) + \frac{1}{2}\sum_{k=1}^{\infty}(\alpha_k + i\beta_k)\exp\left(i\frac{2\pi(-k)x}{L}\right) \\
&= \alpha_0 + \frac{1}{2}\sum_{k=1}^{\infty}(\alpha_k - i\beta_k)\exp\left(i\frac{2\pi kx}{L}\right) + \frac{1}{2}\sum_{k=-\infty}^{-1}(\alpha_{-k} + i\beta_{-k})\exp\left(i\frac{2\pi kx}{L}\right) \qquad (5)
\end{aligned}$$

This can be written more simply as

$$f(x) = \sum_{k=-\infty}^{\infty} \gamma_k \exp\left(i\frac{2\pi kx}{L}\right) \tag{6}$$

where

$$\gamma_k = \begin{cases} \frac{1}{2}(\alpha_{-k} + i\beta_{-k}) & k < 0 \\ \alpha_0 & k = 0 \\ \frac{1}{2}(\alpha_k - i\beta_k) & k > 0 \end{cases} \tag{7}$$

The coefficients γ_k are complex in general. They can be calculated by evaluating the integral

$$\begin{aligned} \int_0^L f(x) \exp\left(-i\frac{2\pi kx}{L}\right) dx &= \int_0^L \sum_{k'=-\infty}^{\infty} \gamma_{k'} \exp\left(i\frac{2\pi k'x}{L}\right) \exp\left(-i\frac{2\pi kx}{L}\right) dx \\ &= \sum_{k'=-\infty}^{\infty} \gamma_{k'} \int_0^L \exp\left(i\frac{2\pi(k'-k)x}{L}\right) dx \end{aligned} \tag{8}$$

where k, k' are integers and therefore $k' - k$ is also an integer. Note that k takes one and only one value. The sum is over all possible values of k'. There will be only one term in the sum for which $k' = k$; in other words, k' can take values $-\infty, \ldots, k-1, k, k+1, \ldots, \infty$.

First let's calculate the integral inside the sum for terms where $k' \neq k$:

$$\begin{aligned} \int_0^L \exp\left(i\frac{2\pi(k'-k)x}{L}\right) dx &= \frac{L}{2\pi(k'-k)} \exp\left(i\frac{2\pi(k'-k)x}{L}\right)\bigg|_0^L \\ &= \frac{L}{2\pi(k'-k)}\left[e^{i2\pi(k'-k)} - 1\right] \\ &= \frac{L}{2\pi(k'-k)}[1-1] \\ &= 0 \end{aligned} \tag{9}$$

because

$$\begin{aligned} e^{i2\pi n} &= \cos(2\pi n) + i\sin(2\pi n) \\ &= 1 + i(0) \\ &= 1 \end{aligned} \tag{10}$$

when n is an integer.

Now let's calculate the integral inside the sum for the one term where if $k' = k$:

$$\begin{aligned} \int_0^L \exp\left(i\frac{2\pi(k'-k)x}{L}\right) dx &= \int_0^L dx \\ &= L \end{aligned} \tag{11}$$

Now let's write out the sum again:

$$\begin{aligned} \int_0^L f(x) \exp\left(-i\frac{2\pi kx}{L}\right) dx &= \sum_{k'=-\infty}^{\infty} \gamma_{k'} \int_0^L \exp\left(i\frac{2\pi(k'-k)x}{L}\right) dx \\ &= \ldots + \gamma_{k'=k-1}(0) + \gamma_{k'=k}(L) + \gamma_{k'=k+1}(0) + \ldots \\ &= \gamma_k L \end{aligned} \tag{12}$$

Therefore

$$\gamma_k = \frac{1}{L}\int_0^L f(x)\exp\left(-i\frac{2\pi kx}{L}\right)dx \tag{13}$$

Given the function $f(x)$, we can find the Fourier coefficients γ_k using the equation above; alternatively, given the coefficients γ_k, we can find $f(x)$ from the Fourier series, equation 6.

Section 9, Exercise 1

2 The Discrete Fourier Transform

For some specific functions $f(x)$, the integral above to determine the coefficients can be performed analytically. However, if $f(x)$ is complicated, or if $f(x)$ is not known analytically (for example if $f(x)$ is a series of data points), then the integral needs to be performed numerically.

Let's calculate the integral using the trapezoidal rule. We will use N evenly spaced points from 0 to L, with $\Delta x = L/N$, so that $x_n = n(L/N)$ (and therefore $x_0 = 0$, $x_N = L$).

$$\begin{aligned}
\gamma_k &= \frac{1}{L}\int_0^L f(x)\exp\left(-i\frac{2\pi kx}{L}\right)dx \\
&= \frac{1}{L}\frac{L}{N}\left[\frac{1}{2}f(0)\exp\left(-i\frac{2\pi k(0)}{L}\right) + \frac{1}{2}f(L)\exp\left(-i\frac{2\pi kL}{L}\right) + \sum_{n=1}^{N-1} f(x_n)\exp\left(-i\frac{2\pi kx_n}{L}\right)\right] \\
&= \frac{1}{N}\left[\frac{1}{2}f(0) + \frac{1}{2}f(L) + \sum_{n=1}^{N-1} f(x_n)\exp\left(-i\frac{2\pi kx_n}{L}\right)\right] \qquad (14)
\end{aligned}$$

By definition, our function $f(x)$ is periodic on the interval 0 to L, meaning that $f(0) = f(L)$, so we can rewrite this as

$$\begin{aligned}
\gamma_k &= \frac{1}{N}\left[\frac{1}{2}f(0) + \frac{1}{2}f(0) + \sum_{n=1}^{N-1} f(x_n)\exp\left(-i\frac{2\pi kx_n}{L}\right)\right] \\
&= \frac{1}{N}\left[f(0) + \sum_{n=1}^{N-1} f(x_n)\exp\left(-i\frac{2\pi kx_n}{L}\right)\right] \\
&= \frac{1}{N}\left[f(0)\exp\left(-i\frac{2\pi kx_0}{L}\right) + \sum_{n=1}^{N-1} f(x_n)\exp\left(-i\frac{2\pi kx_n}{L}\right)\right] \\
&= \frac{1}{N}\sum_{n=0}^{N-1} f(x_n)\exp\left(-i\frac{2\pi kx_n}{L}\right) \qquad (15)
\end{aligned}$$

We can write this more simply as by saying $f(x_n) = y_n$, and using the definition $x_n = n(L/N)$:

$$\begin{aligned}\gamma_k &= \frac{1}{N}\sum_{n=0}^{N-1} y_n \exp\left(-i\frac{2\pi knL}{LN}\right)\\ &= \frac{1}{N}\sum_{n=0}^{N-1} y_n \exp\left(-i\frac{2\pi kn}{N}\right)\end{aligned} \tag{16}$$

Written this way, we don't need to know the specific positions x_n, or the width of the full interval L. All that is needed is the total number of samples from the function N and the function values y_n.

This is known as the *discrete Fourier transform* (DFT) of the samples y_n. It is usually written as

$$c_k = \sum_{n=0}^{N-1} y_n \exp\left(-i\frac{2\pi kn}{N}\right) \tag{17}$$

where the new constant c_k is related to γ_k just by a factor of N.

We arrived at this result by approximating the integral given in Equation 13 using the trapezoidal rule. This means that our solution for the coefficients in equation 17 is an approximation of those coefficients.

We start by rewriting the sum in Equation 17 using a geomtric series. Recall that the sum of the first N terms in a geometric series is given by

$$\sum_{k=0}^{N-1} r^k = \frac{1-r^N}{1-r} \tag{18}$$

Let

$$r = \exp\left(i\frac{2\pi m}{N}\right) \tag{19}$$

where m is an integer. Consider the sum

$$\sum_{k=0}^{N-1} r^k = \sum_{k=0}^{N-1} \exp\left(i\frac{2\pi km}{N}\right) \tag{20}$$

If $m = 0$ or m is a multiple of N ($m = jN$ where j is an integer), then we have

$$\sum_{k=0}^{N-1} \exp\left(i\frac{2\pi k(0)}{N}\right) = \sum_{k=0}^{N-1} 1 = N \tag{21}$$

$$\sum_{k=0}^{N-1} \exp\left(i2\pi kj\right) = \sum_{k=0}^{N-1} 1 = N \tag{22}$$

If $m \neq 0$ and is not a mutiple of N, we have

$$\begin{aligned}
\sum_{k=0}^{N-1} \exp\left(i\frac{2\pi k m}{N}\right) &= \frac{1-\exp\left(i2\pi m\right)}{1-\exp\left(i\frac{2\pi m}{N}\right)} \\
&= \frac{1-1}{1-\exp\left(i\frac{2\pi m}{N}\right)} \\
&= 0
\end{aligned} \tag{23}$$

Summarizing the results:

$$\sum_{k=0}^{N-1} \exp\left(i\frac{2\pi k m}{N}\right) = \begin{cases} N & m = 0 \text{ or } m \text{ is a multiple of } N \\ 0 & \text{otherwise} \end{cases} \tag{24}$$

Now consider this sum, and substitute for c_k:

$$\begin{aligned}
\sum_{k=0}^{N-1} c_k \exp\left(i\frac{2\pi k n}{N}\right) &= \sum_{k=0}^{N-1}\left(\sum_{n'=0}^{N-1} y_{n'} \exp\left(-i\frac{2\pi k n'}{N}\right)\right) \exp\left(i\frac{2\pi k n}{N}\right) \\
&= \sum_{k=0}^{N-1}\sum_{n'=0}^{N-1} y_{n'} \exp\left(i\frac{2\pi k(n-n')}{N}\right) \\
&= \sum_{n'=0}^{N-1} y_{n'} \sum_{k=0}^{N-1} \exp\left(i\frac{2\pi k(n-n')}{N}\right)
\end{aligned} \tag{25}$$

Let's assume that $0 \leq n \leq N$, so that both n and n' are less than N, and there's no way that $n - n'$ could be a nonzero multiple of N. Let's write out a few terms in the sum over n'.

$$\begin{aligned}
\sum_{k=0}^{N-1} c_k \exp\left(i\frac{2\pi k n}{N}\right) &= \ldots + y_{n'=n-1}\sum_{k=0}^{N-1} \exp\left(i\frac{2\pi k(n-(n-1))}{N}\right) + y_{n'=n}\sum_{k=0}^{N-1} \exp\left(i\frac{2\pi k(n-n)}{N}\right) \\
&\quad + y_{n'=n+1}\sum_{k=0}^{N-1} \exp\left(i\frac{2\pi k(n-(n+1))}{N}\right) + \ldots \\
&= \ldots + y_{n-1}(0) + y_n(N) + y_{n+1}(0) + \ldots \\
\sum_{k=0}^{N-1} c_k \exp\left(i\frac{2\pi k n}{N}\right) &= y_n(N)
\end{aligned} \tag{26}$$

Therefore

$$y_n = \frac{1}{N}\sum_{k=0}^{N-1} c_k \exp\left(i\frac{2\pi k n}{N}\right) \tag{27}$$

This is called the *inverse discrete Fourier transform* or inverse DFT. Why is this significant? We said that the formula we got for the coefficients c_k was approximate (since it was based on a numerical approximation to an integral). But using those values of c_k, we can *exactly* recover the values y_n that we used to calculate them. We can go back and forth between c_k and y_n without losing any detail; both give us a complete representation of the original data.

Let's compare this discrete Fourier transform to the Fourier series we started from:

$$f(x) = \sum_{k=-\infty}^{\infty} \gamma_k \exp\left(i\frac{2\pi kx}{L}\right) \tag{28}$$

$$\gamma_k = \frac{1}{L}\int_0^L f(x)\exp\left(-i\frac{2\pi kx}{L}\right)dx \tag{29}$$

$$y_n = \frac{1}{N}\sum_{k=0}^{N-1} c_k \exp\left(i\frac{2\pi kn}{N}\right) \tag{30}$$

$$c_k = \sum_{n=0}^{N-1} y_n \exp\left(-i\frac{2\pi kn}{N}\right) \tag{31}$$

The Fourier series is an infinite sum, while the DFT only has N terms, which is clearly more practical to compute. Also, the DFT only requires us to know the function value at certain points, while the Fourier series is applicable for any value of x. If we are using a set of data points for example, the DFT is the only available method. However, the downside of this is that the coefficients c_k contain no information about what is going on in between the discrete sample points. Two functions could have the same Fourier transform, but behave wildly different in between the sample points (see Figure 1). But, as long as a function is reasonably smooth, the DFT is reasonably good description.

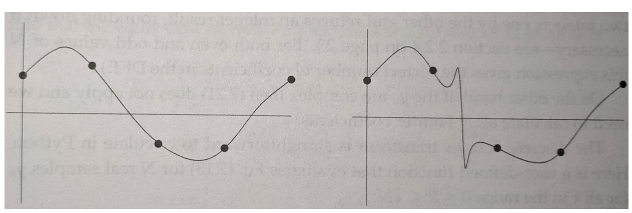

Figure 1:

Everything so far works equally well regardless of whether $f(x)$ is real or complex. However, for physics applications, we are most often dealing with real functions (observables). We can save some calculation time for the case of a real $f(x)$.

Suppose all the y_n are real, and let's consider the value of c_k for some k that is less than N

but greater than $N/2$, i.e. $k = N - j$, where $1 \leq j < N/2$. Then

$$\begin{aligned}
c_{N-j} &= \sum_{n=0}^{N-1} y_n \exp\left(-i\frac{2\pi(N-j)n}{N}\right) \\
&= \sum_{n=0}^{N-1} y_n \exp\left(-i2\pi n\right) \exp\left(i\frac{2\pi jn}{N}\right) \\
&= \sum_{n=0}^{N-1} y_n \exp\left(i\frac{2\pi jn}{N}\right) \\
&= c_j^* \qquad (32)
\end{aligned}$$

i.e., c_{N-j} is the complex conjugate of c_j for all j less than half N. Therefore, for real numbers y_n, we can get away with only calculating the coefficients for half of the values of k, and the other half are just the complex conjugates of the first half. Note our coefficients c_k go from 0 to $N-1$: $c_0, c_1, c_2, \ldots, c_{N-1}$. The relationship above applies beginning with c_1, not c_0: $c_1 = c_{N-1}^*$, $c_2 = c_{N-1}^*$, and so on.

Example: DFT Coefficients

Now that we can calculate the DFT coefficients, let's compare results for a function for which we can find the Fourier series analytically. Suppose $f(x) = \sin x = \sin(2\pi x/(2\pi)$, so $L = 2\pi$, i.e. $\sin x$ is periodic on the interval $0 \leq x \leq L$. We can solve for the coefficients γ_k:

$$\begin{aligned}
\gamma_k &= \frac{1}{L}\int_0^L f(x) \exp\left(-i\frac{2\pi kx}{L}\right) dx \\
&= \frac{1}{2\pi}\int_0^{2\pi} \sin x \exp\left(-i\frac{2\pi kx}{2\pi}\right) dx \\
&= \frac{1}{2\pi}\int_0^{2\pi} \sin x \exp\left(-ikx\right) dx \qquad (33)
\end{aligned}$$

If $k = 1$

$$
\begin{aligned}
\gamma_1 &= \frac{1}{2\pi}\int_0^{2\pi} \sin x \exp\left(-ix\right) dx \\
&= \frac{1}{2\pi}\int_0^{2\pi} \frac{1}{2} i \left(e^{-ix} - e^{ix}\right) \exp\left(-ix\right) dx \\
&= \frac{i}{4\pi}\int_0^{2\pi} \left(e^{-i2x} - 1\right) dx \\
&= \frac{i}{4\pi}\left(-\frac{1}{2i}e^{-2ix} - x\right)\Bigg|_0^{2\pi} \\
&= \frac{i}{4\pi}\left(-\frac{1}{2i}e^{-4\pi i} - 2\pi - (-\frac{1}{2i} - 0)\right) \\
&= \frac{i}{4\pi}\left(-\frac{1}{2i} - 2\pi + \frac{1}{2i}\right) \\
&= -\frac{i}{2}
\end{aligned} \tag{34}
$$

Similarly, $\gamma_{-1} = \frac{i}{2}$.

If $k \neq \pm 1$, we have

$$
\begin{aligned}
\gamma_k &= \frac{1}{2\pi}\int_0^{2\pi} \sin x \exp\left(-ikx\right) dx \\
&= \frac{1}{2\pi}\int_0^{2\pi} \frac{1}{2} i \left(e^{-ix} - e^{ix}\right) \exp\left(-ikx\right) dx \\
&= \frac{i}{4\pi}\int_0^{2\pi} \left(e^{-i(k+1)x} - e^{-i(k-1)x}\right) dx \\
&= \frac{i}{4\pi}\left(-\frac{1}{i(k+1)}e^{-i(k+1)x} + \frac{1}{i(k-1)}e^{-i(k-1)x}\right)\Bigg|_0^{2\pi} \\
&= \frac{i}{4\pi}\left[-\frac{1}{i(k+1)} + \frac{1}{i(k-1)} - \left(\frac{1}{i(k+1)} + \frac{1}{i(k-1)}\right)\right] \\
&= 0
\end{aligned} \tag{35}
$$

Now let's find $f(x)$ using these coefficients:

$$
\begin{aligned}
f(x) &= \sum_{k=-\infty}^{\infty} \gamma_k \exp\left(i\frac{2\pi k x}{L}\right) \\
&= \ldots + 0 + \gamma_{-1}\exp\left(i\frac{2\pi(-1)x}{2\pi}\right) + \gamma_1 \exp\left(i\frac{2\pi(1)x}{2\pi}\right) + 0 + \ldots \\
&= \ldots + 0 + \frac{i}{2}e^{-ix} + 0 + -\frac{i}{2}e^{ix} + 0 + \ldots \\
&= \frac{i}{2}\left(e^{-ix} - e^{ix}\right) \\
&= \sin x
\end{aligned} \tag{36}
$$

Let's compare this result to the DFT for $f(x) = \sin x$.

Example: DFT for sin(x)

First, notice that c_1 is the only coefficient that isn't close to zero, as we expect based on our calculation for the Fourier transform above. The fact that the other coefficients aren't exactly zero is a consequence of doing a discrete transform. Also notice that c_1 differs in magnitude by a factor of N compared to the γ_1 we calculated above, as expected. Now let's look at the DFT for some other simple functions.

Example: DFT for simple functions

So we see that the amplitude of the sine wave affects the *magnitude* of the coefficients; the frequency of the sine wave affects *which* coefficients are non-zero. The value of k for the coefficients c_k is related to the frequency. In physics, for a function that's periodic in time, we call ω in $\sin(\omega t)$ the frequency; for a function that's periodic in space, we usually call the spatial frequency the wave number, like h in $\sin(hx)$. For our purposes here, we'll call both frequencies. The frequency (temporal or spatial) is related to the period T, the interval over which the function is periodic, as $T = 2\pi/\omega$.

Section 9, Exercise 2

Dominant frequencies are peaks in the coefficient spectrum. Sometimes the k's neighboring the dominant ones have small but non-zero values. The more points (N) we use, the coefficients at the non-dominant frequencies get closer to zero. If we use a function that's not periodic, like $f(x) = x$, the dominant k is zero. The interval L used for the function must include at least one period of the function, or the DFT will find that the function is not periodic, i.e. will find that $k = 0$ is dominant.

This brings us to some of the main uses of Fourier transforms in physics. One is detecting periodicity. If you have some data that looks like it might be periodic, but you're not sure because of noise or background, etc., a DFT can detect the frequency of the oscillations, if there are oscillations. Also, the DFT can be used as a "spectral analyzer," detecting all frequency components, even the less dominant ones that might not be obvious by eye.

Example: Oscillatory Signal With Noise

This next data set represents the number of observed sunspots each month from January 1749 until November 2010. A sunspot is an area on the surface of the Sun that appears darker than surrounding areas. In this example, we plot square of the the magnitude of the coefficients; this is called the *power spectrum*, since power is typically proportional to the square of the amplitude of a sine wave (think of a light wave for example).

Example: Detecting Periodicity

The most prominent peak in the power spectrum occurs at $k = 0$. This is a sign that the data is not perfectly periodic; not surprising since the amplitude of the peak is not consistent over time, even though the period looks to be. We see another peak in the power spectrum around $k \sim 25$. The data spans 3,142 months, so $k = 25$ corresponds to a period of $T = 3,142$ months/25 = 125.68 months $\sim$ 10.5 years. This matches well with the number that is usually quoted for the sunspot cycle, approximately 11 years.

3 Two-Dimensional Fourier Transforms

Suppose we have an $M \times N$ grid of samples y_{mn}. To perform a two-dimensional (2D) Fourier transform, we first perform a one-dimensional Fourier transform on each of the M rows, using Equation 17:

$$c'_{ml} = \sum_{n=0}^{N-1} y_{mn} \exp\left(-i\frac{2\pi ln}{N}\right) \tag{37}$$

Now we have N coefficients for each value of l. We can now take the lth coefficient in each of the M rows and Fourier transform these M values again:

$$c_{kl} = \sum_{m=0}^{M-1} c'_{ml} \exp\left(-i\frac{2\pi km}{M}\right) \tag{38}$$

Combining these two equations, we have

$$\begin{aligned} c_{kl} &= \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} y_{mn} \exp\left(-i\frac{2\pi ln}{N}\right) \exp\left(-i\frac{2\pi km}{M}\right) \\ &= \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} y_{mn} \exp\left[-i2\pi\left(\frac{km}{M} + \frac{ln}{N}\right)\right] \end{aligned} \tag{39}$$

The inverse transform is

$$y_{mn} = \frac{1}{MN}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} c_{kl} \exp\left[i2\pi\left(\frac{km}{M} + \frac{ln}{N}\right)\right] \tag{40}$$

If the samples y_{mn} are real, then when we do the first set of Fourier transforms given by Equation 37, we'll have $N/2+1$ $((N+1)/2)$ independent coefficients if N is even (odd), as we saw before. But these coefficients are complex in general, so when we do the second set of transforms given by Equation 38, we can't take the shortcut - we have to calculate all M coefficients. So for a 2D Fourier transform of an $M \times N$ grid of real numbers is a grid of complex numbers with $M \times (N/2+1)$ or $M \times (N+1)/2$ independent coefficients.

2D transforms are used in image processing, for example.

---

4 Direct Cosine Transforms

Consider the cosine series

$$f(x) = \sum_{k=0}^{\infty} \alpha_k \cos\left(\frac{2\pi k x}{L}\right) \tag{41}$$

All functions of this form are symmetric about the midpoint of the interval ($L/2$). Since not all functions have this feature, it may seem that this series would not be very useful. But, we can turn any function in a finite interval into a symmetric function by adding to it the mirror image of itself (See Figure 2). Therefore the cosine series can be used for essentially any function.

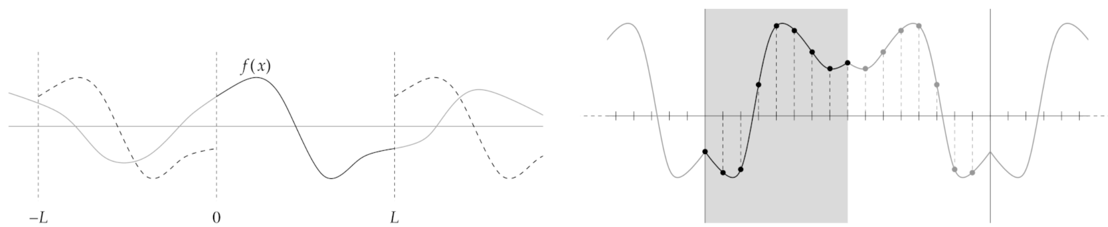

Figure 2: Left: Even if $f(x)$ is not periodic (solid curve), we can calculate a Fourier series for an interval $x = 0$ to $x = L$ by replacing the function outside the interval with a repetition of the portion of the curve inside the interval (dotted curve). Right: Similarly, we can create a symmetric periodic function by taking $f(x)$ on some interval, making the mirror image, and making this repeat outside the interval of interest.

We won't go through the details of working out the coefficients for this special case, but it proceeds similarly to what we've done, using some special features of the cosine series. The discrete cosine transform (DCT) is usually preferred for data that are not inherently periodic. You can certainly apply the DFT to data that are not periodic; you just assume that the function repeats itself exactly outside the given integral (see Figure 2). However, this can create a discontinuity that can cause problems for the DFT. This problem is avoided using the DCT.

Discrete cosine transforms have important technological uses, including the image format jpeg, the video compression format mpeg, and the audio file format mp3. These basically work by taking data and performing a DCT, and discarding all the small coefficients. This makes the file size smaller, since not all the data is kept, and the original image can be (approximately) reconstructed using the large coefficients. For mp3's, coefficients are not only discarded on the basis of magnitude, but also based on what the human ear can and cannot hear.

5 Fast Fourier Transforms

Section 9, Exercise 3

Let's assume that we have N samples, where $N = 2^m$ and m is an integer. Let's consider the sum to calculate each coefficient, and divide the terms into two equally sized groups (which we can always do if N is a power of 2).

$$\begin{aligned}
c_k &= \sum_{n=0}^{N-1} y_n \exp\left(-i\frac{2\pi k n}{N}\right) \\
&= \left[y_0 \exp\left(-i\frac{2\pi k(0)}{N}\right) + y_2 \exp\left(-i\frac{2\pi k(2)}{N}\right) + y_4 \exp\left(-i\frac{2\pi k(4)}{N}\right) + \dots\right] \\
&+ \left[y_1 \exp\left(-i\frac{2\pi k(1)}{N}\right) + y_3 \exp\left(-i\frac{2\pi k(3)}{N}\right) + y_5 \exp\left(-i\frac{2\pi k(5)}{N}\right) + \dots\right] \\
&= \sum_{r=0}^{N/2-1} y_{2r} \exp\left(-i\frac{2\pi k(2r)}{N}\right) + \sum_{r=0}^{N/2-1} y_{2r+1} \exp\left(-i\frac{2\pi k(2r+1)}{N}\right) \\
&= \sum_{r=0}^{N/2-1} y_{2r} \exp\left(-i\frac{2\pi k(2r)}{N}\right) + \sum_{r=0}^{N/2-1} y_{2r+1} \exp\left(-i\frac{2\pi k(2r)}{N}\right) \exp\left(-i\frac{2\pi k}{N}\right) \\
&= \sum_{r=0}^{N/2-1} y_{2r} \exp\left(-i\frac{2\pi k(2r)}{N}\right) + \exp\left(-i\frac{2\pi k}{N}\right) \sum_{r=0}^{N/2-1} y_{2r+1} \exp\left(-i\frac{2\pi k(2r)}{N}\right)
\end{aligned} \tag{42}$$

Let

$$\begin{aligned}
E_k &= \sum_{r=0}^{N/2-1} y_{2r} \exp\left(-i\frac{2\pi k(2r)}{N}\right) \\
&= \sum_{r=0}^{N/2-1} y_{2r} \exp\left(-i\frac{2\pi k r}{N/2}\right)
\end{aligned} \tag{43}$$

and

$$\begin{aligned}
O_k &= \sum_{r=0}^{N/2-1} y_{2r+1} \exp\left(-i\frac{2\pi k(2r)}{N}\right) \\
&= \sum_{r=0}^{N/2-1} y_{2r+1} \exp\left(-i\frac{2\pi k r}{N/2}\right)
\end{aligned} \tag{44}$$

E_k and O_k are both Fourier transforms of the same function, but with $N/2$ samples - half as many points spaced out twice as far apart.

With these definitions, the coefficients c_k in our original Fourier transform can be calculated as

$$c_k = E_k + \exp\left(-i\frac{2\pi k}{N}\right) O_k \tag{45}$$

where E_k and O_k are the coefficients of the two smaller Fourier transforms. So calculating c_k is easy if we know E_k and O_k - how do we find those? We find those by repeating the process - split each sum into their even and odd terms and express each coefficient as a sum of those two pieces, including that exponential factor.

Note that we made N to be a power of 2, not just even. When you divide an even number by 2, you can get an odd number (like $6/2 = 3$), which then can't be divided into two equal samples. When you divide a power of 2 by 2, you always get another power of 2 ($2^m/2 = 2^{m-1}$), so you can keep dividing by 2 and always get a sample that can be divided in half again.

We keep going with this process until each transform is the transform of a single sample (one number). For just one sample, calculating the coefficient is trivial:

$$c_0 = \sum_{n=0}^{0} y_0 e^0 = y_0 \tag{46}$$

The procedure actually proceeds in reverse from what is described above; we start with N individual samples, which are their own Fourier transforms (i.e. $c_0 = y_0$ as above), and then we combine them in groups of two, then in groups of four, and so on, until we have the full transform.

This process is called the *fast Fourier transform* or FFT, because it has a speed advantage over the DFT. At the first round of calculation we have N samples, and therefore do N transforms with one coefficient each. The next round of calculation, we combine those N samples into pairs, and do $N/2$ Fourier transforms, which each have two coefficients, for a total of $(N/2)2 = N$ coefficients. At the next round, the N samples are grouped in fours, so we do $N/4$ Fourier transforms, which each have four coefficients, for a total of $(N/4)4 = N$ coefficients. So we see we calculate N coefficients at each level. If $N = 2^m$, then we can go though the pairing process $m = \log_2 N$ times, for a total of $N \log_2 N$ coefficients to calculate the full transform. This is fewer calculations than the $\sim N^2/2$ calculations necessary for the DFT. For example, if we have a million (10^6) samples, the DFT requires $10^{12}/2 = 5 \times 10^{11}$ operations, which is not practical on a typical computer. However, the FFT requires $10^6 \log_2(10^6) \approx (10^6)(20) = 2 \times 10^7$ operations, which is reasonable.

Let's talk a little about the mechanics of how this is done. Say we have a sample of $N = 2$ numbers, $y_0 = a$ and $y_1 = b$. According to Equation 45, we should treat these as two separate samples of one number ($M = 1$), let's call them $w_0 = a$ and $v_0 = b$. We find the Fourier transform of each sample using the standard equation 17:

$$\begin{aligned}
c_k^w &= \sum_{n=0}^{M-1} w_n \exp\left(-i\frac{2\pi kn}{M}\right) \\
&= \sum_{n=0}^{0} w_n \exp\left(-i2\pi kn\right) \qquad (47) \\
c_0^w &= w_0 \\
&= a \qquad (48)
\end{aligned}$$

and

$$\begin{aligned}
c_k^v &= \sum_{n=0}^{M-1} v_n \exp\left(-i\frac{2\pi kn}{M}\right) \\
&= \sum_{n=0}^{0} v_n \exp\left(-i2\pi kn\right) \qquad (49) \\
c_0^v &= v_0 \\
&= b \qquad (50)
\end{aligned}$$

where the superscripts w and v are meant to distinguish between the two Fourier transforms of samples w_n and samples v_n.

Now we define c_k^w as E_k and c_k^v as O_k (because we defined w_0 as the even term y_0 and v_0 as the odd term y_1 in the original sequence of 2 numbers). Now we use Equation 45 to calculate the coefficients. But wait - there are four values needed to calculate the two coefficients: c_0 needs E_0 and O_0 and c_1 needs E_1 and O_1, but we only have two values, c_0^w and c_0^v.

If E_k and O_k are the coefficients of a Fourier transform with $N/2$ samples, then there will be only $N/2$ values of E_k and O_k, while there should be N coefficients c_k for the whole transform with N samples. (In other words, on the left-hand side of Equation 45, k should go from 0 to $N-1$; on the right-hand side, k should go from 0 to $N/2-1$.) That's because the last half of the coefficients c_k will be identical to the first half. Consider $k = s$ and $k = N/2 + s$, where $s < N/2$:

$$E_s = \sum_{r=0}^{N/2-1} y_{2r} \exp\left(-i\frac{2\pi sr}{N/2}\right) \tag{51}$$

$$\begin{aligned}
E_{N/2+s} &= \sum_{r=0}^{N/2-1} y_{2r} \exp\left(-i\frac{2\pi(N/2+s)r}{N/2}\right) \\
&= \sum_{r=0}^{N/2-1} y_{2r} \exp\left(-i2\pi r\right) \exp\left(-i\frac{2\pi sr}{N/2}\right) \\
&= \sum_{r=0}^{N/2-1} y_{2r} \exp\left(-i\frac{2\pi sr}{N/2}\right) \\
&= E_s
\end{aligned} \tag{52}$$

Similarly, for O_k

$$O_s = O_{N/2+s} \tag{53}$$

So in our example with $N = 2$, $E_1 = E_0 = c_0^w$ and $O_1 = O_0 = c_0^v$, and we have

$$c_0 = E_0 + \exp\left(-i\frac{2\pi(0)}{2}\right) O_0 = E_0 + O_0 = c_0^w + c_0^v = a + b \tag{54}$$

$$c_1 = E_1 + \exp\left(-i\frac{2\pi(1)}{2}\right) O_1 = E_1 - O_0 = c_0^w + c_0^v = a - b \tag{55}$$

where we used

$$e^{\pm i\pi} = \cos\pi \pm i\sin\pi = -1 \tag{56}$$

If we calculate the DFT, we get

$$\begin{aligned}
c_k &= \sum_{n=0}^{N-1} y_n \exp\left(-i\frac{2\pi kn}{N}\right) \\
&= \sum_{n=0}^{2-1} y_n \exp\left(-i\frac{2\pi kn}{2}\right) \\
&= \sum_{n=0}^{1} y_n \exp\left(-i\pi kn\right)
\end{aligned} \tag{57}$$

$$c_0 = y_0 e^0 + y_1 e^0 = y_0 + y_1 = a + b \tag{58}$$

$$c_1 = y_0 e^0 + y_1 e^{-i\pi} = y_0 - y_1 = a - b \tag{59}$$

as expected.

Let's write out the math for a general N a little more precisely. We'll call stage 0 the case where we have one set of N samples. At stage 1, we have two sets of $N/2$ samples. In general, at stage m, we have 2^m sets of $N/2^m$ samples. The first of these sets will consist of sample numbers $y_0, y_{2^m}, y_{2\times 2^m}, y_{3\times 2^m}, \ldots$, which we can write as $y_{2^m r}$, with $r = 0 \ldots N/2^m - 1$. The second set will consist of sample numbers $y_{2^m r+1}$, the third set $y_{2^m r+2}$, and so on. The DFT of the jth set of samples is

$$\sum_{r=0}^{N/2^m-1} y_{2^m r+j} \exp\left(-i\frac{2\pi k(2^m r + j)}{N}\right) = \exp\left(-i\frac{2\pi kj}{N}\right) \sum_{r=0}^{N/2^m-1} y_{2^m r+j} \exp\left(-i\frac{2\pi kr}{N/2^m}\right)$$
$$\equiv \exp\left(-i\frac{2\pi kj}{N}\right) E_k^{(m,j)} \tag{60}$$

where $j = 0, \ldots, 2^m - 1$. With this, Equation 45 can be generalized as

$$E_k^{(m,j)} = E_k^{(m+1,j)} + \exp\left(\frac{-i2\pi 2^m k}{N}\right) E_k^{(m+1,j+2^m)} \tag{61}$$

The algorithm then works by starting at the last stage $m = \log_2 N$ and work backwards to $m = 0$. Once we get to $m = 0$, we have the values of the final Fourier coefficients:

$$E_k^{0,0} = \sum_{r=0}^{N-1} y_r \exp\left(-i\frac{2\pi kr}{N}\right)$$
$$= c_k \tag{62}$$

Remember that for any value of m we need to evaluate coefficients for $k = 0$ to $k = N/2^m - 1$. But from the previous stage, we only have coefficients from $k = 0$ to $k = N/2^{m+1} - 1 = (1/2)N/2^m - 1$. But as we saw earlier, the coefficients for values of k beyond $N/2^{m+1} - 1$ are the same as the previous values:

$$E_{N/2^{m+1}+s}^{(m+1,j)} = E_s^{(m+1,j)} \tag{63}$$

There is also an inverse FFT, to go from coefficients c_k to sample values y_n, which proceeds in the same way as the FFT.

We're not going to write our own program to do FFTs, but use the standard functions available in Python. For an FFT of a set of real samples, use the `numpy.fft.rfft` function (where "r" is for "real"). The corresponding inverse FFT is done with `numpy.fft.irfft`. The functions `numpy.fft.fft` and `numpy.fft.ifft` work for a sample of complex numbers. There are also versions of both for 2D FFTs: `numpy.fft.rfft2`, `numpy.fft.irfft2`, `numpy.fft.fft2`, and `numpy.fft.ifft2`.

Example: Fast Fourier Transform

Let's consider how to "smooth" a function. Suppose you have a data set with a lot of noise. This noise shows up has high-frequency wiggles in the data. If you do an FFT on this data, you expect non-zero coefficients at large values of k, corresponding to high frequencies. To smooth the data, you can then set the higher-order coefficients to zero to get rid of the high-frequency noise. Then an inverse FFT will give you a data set with all the high-frequency wiggles filtered out.

In the next example, we examine a data set which contains the daily closing value for each business day from late 2006 until the end of 2010 of the Dow Jones Industrial Average, a measure of average prices on the US stock market. By doing an FFT, setting some fraction of the higher-order coefficients to zero, and then doing an inverse FFT, we can rid the data of tiny wiggles, creating a smoother function.

Example: Smoothing

The next example shows what happens when we apply smoothing to a square wave.

Example: Smoothing, part 2

Section 9, Exercise 4

6 Sampling and Aliasing

Assume we are doing a Fourier transform of N points (where N is even), evenly spaced in time with separation Δt, so that the total interval has size $L = N\Delta t$. As we've noted before, if y_n are all real, then the highest-order *independent* Fourier component is given by $k = N/2$, i.e. all the higher-order (higher-k) coefficients are complex conjugates of the coefficients from $k = 1$ to $k = N/2$. This value of k corresponds to a frequency of

$$\begin{aligned}\omega &= \frac{2\pi k}{L} \\ &= \frac{2\pi(N/2)}{N\Delta t} \\ &= \frac{\pi}{\Delta t} \qquad (64)\end{aligned}$$

If t is measured in seconds, then ω is an angular frequency measured in radians/s. The corresponding frequency in Hz is

$$\begin{aligned}f &= \frac{\omega}{2\pi} \\ &= \frac{\pi/\Delta t}{2\pi} \\ &= \frac{1}{2\Delta t} \qquad (65)\end{aligned}$$

This frequency is called the Nyquist frequency

$$f_{\text{Nyquist}} = \frac{1}{2\Delta t} \qquad (66)$$

Since Δt is the spacing of the sample points, $1/\Delta t$ is called the sampling frequency f_s, and the Nyquist frequency is sometimes defined as

$$f_{\text{Nyquist}} = \frac{f_s}{2} \tag{67}$$

For a sine wave at the Nyquist frequency (half our sampling frequency), we would be sampling it only twice during each period of oscillation, since its oscillation period $T = 1/f_{\text{Nyquist}} = 2\Delta t$, is twice the sampling time Δt. Sampling it any more than this is sufficient to capture this Fourier component. This is known as the sampling theorem, basically that the FFT will give a perfect description of the Fourier components as long as the frequencies of those components are below the Nyquist frequency. "If a function contains no frequencies higher than f_{Nyquist} Hz, it is completely determined by giving its function values at a series of points spaced less than $1/(2f_{\text{Nyquist}})$ seconds apart."

Let's see this in practice. We'll choose some value of N (which is a power of 2 since we are using FFTs) and a value for Δt. Then our interval size is $L = N\Delta t$. Let's define a sine function that has only one frequency f:

$$y = \sin(\omega t) = \sin(2\pi f t) \tag{68}$$

Note that f is measured in Hz, and the angular frequency is $\omega = 2\pi f$. Let the period of our function be $T = L/k$, where k is an integer we choose. Then the frequency $f = 1/T = k/L$. The sampling frequency is determined by Δt as $f_s = 1/\Delta t$. The sampling theorem says we can get a perfect sampling of the Fourier components of our sine function as long as the frequencies of those components are below the Nyquist frequency. That means if $f < f_{\text{Nyquist}}$, then we expect to see only one non-zero coefficient, at the value k. Let's try this for different values of k.

Example: Sampling Theorem

So, under the right circumstances, we see we can get the correct value of k for a single-frequency function. Notice that as k gets larger, the period gets smaller, so the function is repeating more often in the given interval. As k gets larger, our sample points do not look like they are from a nice sine function - this is because the space between our sampling points is large enough that we are missing peaks and valleys of the function, give the small period. But, our FFT still gives us the correct single value of k, even if the points don't look like a sine function by eye.

When performing a discrete Fourier transform on an arbitrary function, two parameters are under our control: the sampling interval (Δt, space between the points) and the number of points in the sample. The sampling interval determines the range of frequencies that can be represented, and the number of points determines the details that can be obtained. Other things that matter: whether or not the sampling duration corresponds to whole periods of the signal and whether the number of points match what the particular FFT algorithm is designed for.

We've seen examples that show that with careful selection of the number of points and sampling interval, we can get the "correct" solution in the sense the Fourier transform of a sine wave with one frequency corresponds to exactly one non-zero coefficient. If your function repeats 3 times over the interval L, the $k = 3$ will be the only non-zero coefficient. But if your function repeats a non-integer number of times, i.e. L is not a multiple of the period, then it will take linear combinations of functions with different frequencies to represent the function. This answer is still "exact" in the sense that the inverse Fourier transform will reproduce the initial samples, but not "correct" in the sense of returning the same frequency as the function that we pulled the samples from.

Example: Sampling

What happens when we use a frequency above the Nyquist frequency? In the example below, we consider a single cosine wave with a frequency of 4 Hz. We use an interval of $L = 6.5$ s and $\Delta t = 0.1$ s, which returns one non-zero coefficient at $k = 26$, corresponding to a frequency of $f = k/L = 26/6.5 = 4$ Hz. When we increase Δt to 0.2 s (thereby decreasing the Nyquist frequency to less than 4 Hz), the peak frequency comes at $k = 6$, corresponding to a frequency of $f = k/L = 6/6.5 \approx 0.9$ Hz.

Example: Aliasing

What is the significance of the other frequency? We see that a cosine function at a frequency of ≈ 0.9 Hz passes through all the same points as the 4 Hz cosine wave when sampled every 0.2 s. We can predict the second frequency by "reflecting" about the Nyquist frequency: $f_{\text{true}} - f_{\text{Nyquist}} = 4 - 2.5 = 1.5$ Hz, so we expect to find the second frequency 1.5 Hz below the Nyquist frequency, at $f = 1$ Hz. This "reflection" of frequencies above the Nyquist frequency is known as *aliasing*.

Typically, we try to arrange for the Nyquist frequency to be higher than any of the Fourier components that are expected to be present in the signal, which might mean using a filter to remove components with frequencies above the Nyquist frequency (anti-aliasing filter), or choosing the appropriate time step to set the Nyquist frequency. Of course if the frequency components of interest are much below other components present in the signal, you must balance staying below the Nyquist frequency with having a reasonable number of points. (Low frequency means a long period. If your Δt must stay small, that means more points in a long period.)