Recitation 21
A tour of Fourier representations

After studying this chapter, you should be able to:
• explain the connection between Fourier series and transforms in terms of sampling the frequency representation; and
• explain the connection between discrete-time and continuous-time Fourier representations in terms of sampling the time representation.

To explore the relations between the four Fourier representations, we use this continuous-time, aperiodic signal $f(t)$:

It looks like a segment of a squared cosine, but its exact time representation is

$$f(t) = \begin{cases} 
0 & \text{for } t < -3; \\
(t + 3)^2/2 & \text{for } -3 \leq t \leq -1; \\
3 - t^2 & \text{for } -1 \leq t \leq 1; \\
(t - 3)^2/2 & \text{for } 1 \leq t \leq 3; \\
0 & \text{for } t > 3. 
\end{cases}$$

These equations look messy but they result from the simple recipe of convolving a pulse with itself twice:

To see that the shape of $f(t)$ is plausible, think about how convolution affects smoothness. The pulse itself has discontinuities. Convolving the pulse with itself is a form of low-pass filtering, so it produces a smoother signal: a triangle with no discontinuities. The triangle has slope discontinuities at its vertices. Convolving the triangle with the pulse low-pass filters the triangle, making a signal $f(t)$ with neither discontinuities nor slope discontinuities.
Exercise 96. Confirm that \( f(t) \) has no slope discontinuities even at the two joints \( t = -1 \) and \( t = 1 \).

Exercise 97. Compute the triple convolution to confirm that \( f(t) \) is indeed as claimed.

We use \( f(t) \) to illustrate connections among:

1. the continuous-time Fourier transform (CTFT),
2. the continuous-time Fourier series (CTFS),
3. the discrete-time Fourier transform (DTFT), and
4. the discrete-time Fourier series (DTFS), usually implemented (and misnamed) as the fast Fourier transform (FFT).

The continuous-time Fourier transform places no restrictions on the signal: It may or may not be periodic, and it may or may not be sampled. This \( f(t) \) is general in that it is aperiodic and nonsampled. The other three transforms restrict the signal. as we examine each transform, we make a corresponding signal from \( f(t) \) that meets the restrictions of that transform.

### 21.1 Continuous-time Fourier transform

The continuous-time Fourier transform of \( f(t) \) is

\[
f(\omega) = \left( 2 \frac{\sin \omega}{\omega} \right)^3.
\]

which looks like:

To check the transform, look at the DC value \( f(\omega = 0) \). It is \( f(\omega = 0) = 8 \), which means that \( f(t) \) should have area 8. And it does, as you can see from a convolution argument. The pulse has area 2, so the triple convolution of the pulse produces a function with area \( 2 \times 2 \times 2 = 8 \).

Exercise 98. Confirm that the area under \( f(t) \) is 8 by integrating \( f(t) \).

Exercise 99. Compute the Fourier transform to confirm that

\[
f(\omega) = \left( 2 \frac{\sin \omega}{\omega} \right)^3.
\]

*Hint*: Use the convolution property instead of doing the Fourier transform integral directly.
21.2 Continuous-time Fourier series

A Fourier series represents only periodic signals, so let’s make a periodic version of \( f(t) \). There are ways to do so. Any periodic version of \( f(t) \) will illustrate the connection between the Fourier transform and Fourier series. But the simplest periodic version is obtained by taking the nonzero region of \( f(t) \) as the period; in mathematical jargon, we are using the support of \( f(t) \) as the period. That region is \(-3 < t < 3\), making the period \( T = 6\). The resulting signal \( f_p(t) \) looks like

\[
\text{We next find the Fourier transform of } f_p(t), \text{ which we can do for any signal. The transform will turn out almost identical to the Fourier series. To find the transform, divide and conquer. The periodic signal is the convolution of } f(t) \text{ with a comb – which is the fancy name for an impulse train. The comb’s teeth are unit-area delta functions spaced 6 units apart.}
\]

Dividing \( f_p(t) \) into a aperiodic part convolved with a comb makes finding its transform easy because convolution of time representations is equivalent to multiplication of frequency representations. So the Fourier transform of \( f_p(t) \) is:

\[
f_p(\omega) = f(\omega) \times \text{Fourier transform of the comb}.
\]

The Fourier transform of a comb with period \( T = 6 \) and amplitude 1 is a frequency comb with period \( 2\pi/T = \pi/3 \) and amplitude \( 2\pi/T \). [The factor of \( 2\pi \) in the amplitude results from our convention for the inverse Fourier transform, and is the least important factor in this discussion.]

Multiplying any function by a comb samples that function. This process produces regularly spaced delta functions whose amplitudes (areas) are the values of the function at the comb locations. Therefore, the Fourier transform of the periodic function \( f_p(t) \) is a sampled version of the Fourier transform of one period \( f(t) \): Making a function periodic samples its transform.

The Fourier series represents the same information as the sampled transform \( f_p(\omega) \), but represents it conveniently. Rather than using delta functions, which are hard to draw, the Fourier series uses their area directly, and just lists the areas indexed by an integer \( k \) (rather than as a function of frequency). As a bonus, the Fourier series drops an annoying factor of \( 2\pi \). In terms of the transform of one period, the \( k \)th Fourier coefficient \( f_k \) is

\[
f_k = \frac{f(\omega_k)}{2\pi} \quad \text{where } \omega_k = 2\pi k/T.
\]

Those coefficients are illustrated in this diagram:

\[
f_k
\]

The tops of the samples sketch out the shape of \( f(\omega) \).

The coefficients \( f_k \) are of order \( 1/k^3 \) for large \( k \), reflecting the \( 1/\omega^3 \) factor in the transform

\[
f(\omega) = \left(2\frac{\sin \omega}{\omega}\right)^3.
\]
So the time signal \( f(t) \) and its periodic version \( f_p(t) \) hardly use high-frequency oscillations, meaning that they are very smooth. You can understand the \( 1/k^3 \) amplitude roll off by looking at levels of smoothness together with the Fourier coefficients for each level. An infinitely discontinuous signal like a delta function has spectrum \( f_k \sim 1 \), which means no roll off and infinite bandwidth. A finitely discontinuous signal – for example a pulse or a sawtooth – has spectrum \( f_k \sim 1/k \). A continuous signal with slope discontinuities – for example a triangle – has spectrum \( f_k \sim 1/k^2 \). So the signal \( f(t) \), which is one level smoother than the triangle, should and does have spectrum \( f_k \sim 1/k^3 \).

### 21.3 Discrete-time Fourier transform

Our next Fourier representation is the discrete-time Fourier transform. As its name suggests, it is most closely related to the continuous-time Fourier transform. So forget about Fourier series for the moment and return to the Fourier transform of \( f(t) \). Rather than making \( f(t) \) periodic, we now sample \( f(t) \) to make a discrete-time signal. Sampling means multiplication by a comb, which produces the sampled signal composed of delta functions. You can sample using any comb spacing \( \Delta t \), and each spacing (and starting position) generates its own discrete-time signal. For simplicity, we use a comb with unit spacing to get the sampled signal:

\[
\begin{align*}
-3 & -2 & -1 & 0 & 1 & 2 & 3 \\
\hline
 \text{n} & \\
\end{align*}
\]

The spikes are delta functions and the spike heights represent their areas in the sampled signal \( f_s(t) \). The discrete-time signal \( f[n] \) conveniently represents the same information without the annoying delta-function placeholders.

Next we find the Fourier transform of \( f_s(t) \) in order to compare it to the discrete-time Fourier transform of \( f[n] \). Multiplication of time representations is, by duality, equivalent to convolution of frequency representations then division by an annoying \( 2\pi \). The Fourier transform of the unit comb is a frequency comb with spacing \( 2\pi \) and amplitude \( 2\pi \). Convolving \( f(\omega) \) with this comb produces a periodic transform \( f_s(\omega) \). **Sampling in time produces a periodic frequency representation.**

The simplest way to find \( f_s(\omega) \) is not direct convolution. It is easier to transform \( f_s(t) \) directly, since it is composed of only a few simple parts (delta functions). Here is the definition of \( f(t) \) again, which we need in order to find the amplitudes of the delta functions:

\[
f(t) = \begin{cases} 
0 & \text{for } t < -3; \\
(t + 3)^2/2 & \text{for } -3 \leq t \leq -1; \\
3 - t^2 & \text{for } -1 \leq t \leq 1; \\
(t - 3)^2/2 & \text{for } 1 \leq t \leq 3; \\
0 & \text{for } t > 3.
\end{cases}
\]

The samples are at integer times (a choice we made for simplicity). The only nonzero samples are

\[
\begin{align*}
f(-2) &= f(2) = 1/2, \\
f(0) &= 3, \\
f(-1) &= f(1) = 2.
\end{align*}
\]

So \( f_s(t) \) is the sum
\[
f_s(t) = \frac{1}{2} \left( \delta(t+2) + \delta(t-2) \right) + 3 \delta(t) + 2 \left( \delta(t+1) + \delta(t-1) \right).
\]

The transform \( f_s(\omega) \) is
\[
f_s(\omega) = 3 + 4 \cos \omega + \cos 2\omega,
\]
which looks like
\[\Omega = \omega \Delta t\]
It is periodic, which it should be, and the period is \( 2\pi \). In general, the period will be \( 2\pi/\Delta t \).

The discrete-time Fourier transform (DTFT) represents this same information slightly more conveniently by using dimensionless frequency \( \Omega = \omega \Delta t \) as the independent variable instead of \( \omega \). Except for that change, the DTFT is identical to the Fourier transform of the sampled signal \( f_s(t) \).

In this example \( \Delta t = 1 \), so even the \( \Omega \) and \( \omega \) axes are the same.

### 21.4 Discrete-time Fourier series

Our final Fourier representation is the discrete-time Fourier series (DTFS). This representation is often called the discrete Fourier transform (DFT) or the fast Fourier transform (FFT) in its usual algorithmic implementation. Those names misleadingly suggest that the frequency spectrum is continuous, so we will keep calling it the discrete-time Fourier series. Although this representation is the most specialized of the four, it is also the most useful because, being discrete, it can be done by a computer without needing to do difficult or impossible symbolic integrations, and because it has an efficient implementation in the so-called fast Fourier transform (that misleading term again).

The continuous-time Fourier series arises from the continuous-time Fourier transform by making the signal \( f(t) \) periodic. Similarly, the discrete-time Fourier series arises from the discrete-time Fourier transform by making the sampled signal \( f_s(t) \) periodic. Here is the resulting periodic, sampled signal \( f_{ps}(t) \):

To make this signal, use the usual procedure of convolving \( f_s(t) \) with the comb. This comb’s teeth are unit delta functions spaced by \( T = 6 \) time units. Convolving with the time comb is equivalent to multiplying by the frequency comb. Therefore, the Fourier transform of \( f_{ps}(t) \) is a sampled version of the discrete-time Fourier transform:

Since the discrete-time Fourier transform is periodic in frequency, the Fourier transform of \( f_{ps}(t) \) is, like \( f_{ps}(t) \) itself, both periodic and sampled. The discrete-time Fourier series conveniently
represents the same information as the Fourier transform by giving the delta-function amplitudes only for one period of the transform.

21.5 Summary

This diagram shows the time signal as altered for each Fourier representation, along with the corresponding frequency representation.