ECE 310 Fall 2023

Lecture 21

Discrete Fourier transform

Corey Snyder

Learning Objectives

After this lecture, you should be able to:

- Explain why the DTFT summation is intractable to fully compute in practice and why this motivates using the DFT.
- Explain the relationship between the DTFT and the DFT, and describe how the two transforms are different.

Recap from previous lecture

We concluded our discussion of sampling in the last lecture with aliasing effect and the consequences of sampling below the Nyquist rate. We will switch topics this lecture to introduce our last fundamental transform of the course: the Discrete Fourier Transform (DFT). We will motivate the use of the DFT and explain how the DFT and DTFT are distinct transforms but closely connected to one another.

1 Computing the DTFT

The discrete-time Fourier transform (DTFT), as the name tells us, computes the Fourier transform of discrete-time signals. However, for the purpose of digital signal processing and more broadly digital computing, we need to address how we may actually go about computing the DTFT of a discrete-time sequence. For example, suppose we want to compute the DTFT of a given signal $x[n]$. We know well by now that

$$X(\omega) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n}. \tag{1}$$

Imagine how you may define and compute $X(\omega)$ in computer code. You may notice two clear issues:

(1) Computing an infinite-length summation. The DTFT is defined as an infinite-length summation. Clearly, this means we cannot compute the full summation if the signal is right-sided, left-sided, or two-sided (infinite-length, in general). We can approximate the DTFT by computing the summation over some finite-length interval. Let $\hat{X}(\omega)$ denote our "finite-length DTFT":

$$\hat{X}(\omega) = \sum_{n=0}^{N-1} x[n]e^{-j\omega n}. \tag{2}$$

Note, however, that this is equivalent to computing

$$\hat{X}(\omega) = \sum_{n=-\infty}^{\infty} x[n]w[n]e^{-j\omega n}, \tag{3}$$

where

$$w[n] = \begin{cases} 1, & 0 \leq n < N \\ 0, & \text{else} \end{cases}. \tag{4}$$

We are effectively *windowing* $x[n]$ in order to compute a finite-length DTFT. This windowing will introduce some distortion in the spectrum via the multiplication in time/windowing property of the DTFT. Recall,

$$x[n]w[n] \overset{\mathcal{F}}{\longleftrightarrow} \frac{1}{2\pi} X(\omega) * W(\omega). \tag{5}$$

Thus, $\hat{X}(\omega) \neq X(\omega)$ due to the windowing effect. We will discuss windowing functions and their practical use further in upcoming lectures.

Still, you may point out: what if our signal $x[n]$ is already finite-length anyway? For example, suppose $x[n]$ is a rectangular pulse signal. This is where the second issue is also important!

(2) $X(\omega)$ evaluates over continuous frequency ω. Even if the signal $x[n]$ is finite-length, e.g. length-N, the DTFT is still defined over the continuous frequency variable ω. The DTFT is 2π periodic so we only need to evaluate over a finite-width support, e.g. $[-\pi, \pi]$; however, we still have infinitely many points between $[-\pi, \pi]$. A potential solution would then be to discretize the frequency variable. Instead of evaluating for all values of ω, we could instead define some ω_k that regularly spaces frequencies between $[0, 2\pi)$ for N values of the frequency:

$$\omega_k = \frac{2\pi k}{N},\ 0 \leq k \leq N-1. \tag{6}$$

Let's now incorporate both a finite-length summation and the discretized ω_k into a modified transform given by $\hat{X}_k$:

$$\hat{X}_k = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi k}{N}n}. \tag{7}$$

Equation 7 may look familiar since it is almost exactly our analysis equation for the discrete-time Fourier series! The only difference is we use N instead of N_0 since we do not assume ahead of time that our signal is periodic, but rather just length N whether by windowing or by its own finite length. Note also that we may select $M > N$ discrete frequencies to compute as well. We will explore this later in this lecture and further in future lectures.

2 Discrete Fourier transform

We have already argued for the need to (1) compute the DTFT over a finite summation and (2) discretize frequencies to make the transform tractable to compute. What we have effectively done in the previous section is actually derive the analysis equation of the discrete Fourier transform (DFT)!

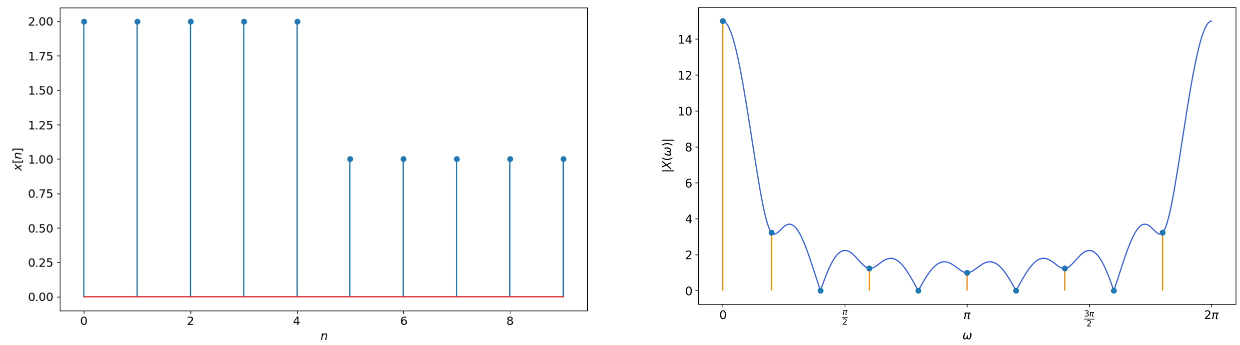
$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi k}{N}n},\ 0 \leq k \leq N-1 \tag{8}$$

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{j\frac{2\pi k}{N}n},\ 0 \leq n \leq N-1. \tag{9}$$

Equations 8 and 9 give us the forward and inverse DFT equations. Alternatively, we refer to the forward DFT as the analysis equation and the inverse DFT as the synthesis equation of the DFT like we have with previous discussion of Fourier analysis.

2.1 Relationship between DTFT and DFT

We have already teased how the DTFT and the DFT relate to one another when we described the computational challenges of the DTFT. Let's now make things more concrete. Suppose we have a finite-length signal



(a) Finite-length $x[n]$ (b) $|X(\omega)|$ and $|X[k]|$

Figure 1: Example of how the DFT evaluates the DTFT of a given finite-length signal. Blue line on the right plot shows the DTFT of the $x[n]$ from the left plot. The orange lines identify where the DFT $X[k]$ samples the DTFT.

$x[n]$ of length N and its corresponding DTFT $X(\omega)$. Thus,

$$X(\omega) = \sum_{n=0}^{N-1} x[n]e^{-j\omega n}. \tag{10}$$

The DFT of $x[n]$, $X[k]$, can be computed according to Eqn. 8. We can also write the DFT sum using the discretized ω_k from Eqn. 6:

$$\begin{aligned} X[k] &= \sum_{n=0}^{N-1} x[n]e^{-j\frac{2\pi k}{N}n} \\ &= \sum_{n=0}^{N-1} x[n]e^{-j\omega_k n}. \end{aligned} \tag{11}$$

$$X[k] = X(\omega)\big|_{w=\omega_k},\ 0 \leq\ k \leq N-1. \tag{12}$$

Equation 12 tells us something very important. *The DFT evaluates the DTFT of a finite-length signal over a regular interval from $\omega = 0$ up to (but not including) $\omega = 2\pi$.* This regular interval is defined by ω_k and the length of the signal. The spacing between the frequencies in ω_k is simply $\frac{2\pi}{N}$. Figure 1 depicts an example of how the DFT extracts values from the DTFT of a finite-length signal.

2.2 N-periodicity of the DFT

Recall that the DTFT is 2π periodic. Thus, the DFT also has 2π periodicity since it is evaluating the periodic spectrum. Since we have N distinct frequencies in the DFT $X[k]$, we say the DFT is N-periodic:

$$X[k] = X[k + mN],\ m \in \mathbb{Z}. \tag{13}$$

Furthermore, the inverse DFT gives a surprising result:

$$x[n+N] = \frac{1}{N}\sum_{k=0}^{N-1} X[k]e^{j\frac{2\pi k}{N}(n+N)} \tag{14}$$

$$= \frac{1}{N}\sum_{k=0}^{N-1} X[k]e^{j\frac{2\pi k}{N}n}\ \underbrace{e^{j2\pi kn}}_{=1,\ \forall k,n} \tag{15}$$

$$= \frac{1}{N}\sum_{k=0}^{N-1} X[k]e^{j\frac{2\pi k}{N}n} \tag{16}$$

$$= x[n]. \tag{17}$$

The above proof shows us that the DFT implies *periodic extension* of our finite-length time-domain signal. In other words, if we take the inverse DFT of some DFT $X[k]$, the recovered signal $\bar{x}[n]$ will be N periodic:

$$\bar{x}[n] = \text{IDFT}\{X[k]\} \tag{18}$$
$$\bar{x}[n + mN] = \bar{x}[n], \ m \in \mathbb{Z}, \tag{19}$$

where IDFT denotes the inverse DFT. This property of the DFT is often confusing to students, so do not worry if it feels unintuitive right now. We will return to this result again in the next lecture since it will be critical to establishing important properties of the DFT.

2.3 Zero-padding

We have already shown that the DFT takes N values from the DTFT of a length-N discrete-time signal. What if instead we would like to evaluate more points of the DTFT when computing the DFT? We can accomplish this using a surprisingly simple technique known as *zero-padding*. Let $x[n]$ be the original length-N signal and $\tilde{x}[n]$ be defined as

$$\tilde{x}[n] = \begin{cases} x[n], & 0 \leq n \leq N-1 \\ 0, & N \leq n \leq N+L-1 \end{cases}. \tag{20}$$

Thus, we add L zeros to the end of $x[n]$ to create $\tilde{x}[n]$. Let's check the resulting DFT of $\tilde{x}[n]$. Let $X(\omega)$ be the DTFT of $x[n]$, thus

$$X[k] = X\left(\frac{2\pi k}{N}\right), \ 0 \leq k \leq N-1. \tag{21}$$

We can define $\tilde{X}[k]$ similarly as follows:

$$\tilde{X}[k] = \sum_{n=0}^{N+L-1} \tilde{x}[n] e^{-j\frac{2\pi k}{N+L}n} \tag{22}$$
$$= \sum_{n=0}^{N-1} \tilde{x}[n] e^{-j\frac{2\pi k}{N+L}n} \tag{23}$$
$$= \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi k}{N+L}n} \tag{24}$$
$$= X\left(\frac{2\pi k}{N+L}\right), \ 0 \leq k \leq N+L-1. \tag{25}$$

We see that $\tilde{X}[k]$ evaluates the exact same DTFT spectrum. The only difference is that $\tilde{X}[k]$ provides L more values of the DTFT. This means $\tilde{X}[k]$ provides a denser visualization of the spectrum of $x[n]$. Specifically, $X[k]$ will have spacing $2\pi/N$ between adjacent values while $\tilde{X}[k]$ will have a smaller spacing of $2\pi/(N+L)$ between values. This all comes at the (likely minor) cost of storing more zeros to represent our signal and compute a longer DFT. Note also that the DTFTs $X(\omega)$ and $\tilde{X}(\omega)$ are identical since they have non-zero values in the same places and zeros elsewhere in the DTFT summation.

Exercise 1: Let $x[n]$ be a length-16 signal with DFT $X[k]$. We zero-pad $x[n]$ with 48 zeros to obtain the length-64 $\tilde{x}[n]$ with DFT $\tilde{X}[k]$. Which values of $\tilde{X}[k]$ will be equal to (a) $X[1]$ and (b) $X[9]$?

For part (a), we have $k = 1$. For the original sequence $x[n]$, this will correspond to the frequency $\frac{2\pi\cdot(1)}{N} = \frac{\pi}{8}$. We can solve for the necessary value of k in $\tilde{X}[k]$ as follows:

$$\frac{2\pi k}{64} = \frac{\pi}{8} \tag{26}$$
$$k = 4 \tag{27}$$
$$X[1] = \tilde{X}[4]. \tag{28}$$

We can follow the same process for part (b):

$$\frac{2\pi k}{64} = \frac{2\pi\cdot(9)}{16} \tag{29}$$
$$k = 36 \tag{30}$$
$$X[9] = \tilde{X}[36]. \tag{31}$$

2.4 Linear algebra formulation (optional reading material if you are interested!)

We will conclude this lecture by considering how a linear algebra formulation of the DFT offers an interesting perspective of the DFT as well as further computational advantages. We have previously discussed the linear algebra interpretations of the z-transform and the DTFT. In both cases, we showed how each transform can be thought of as a dot product between our given signal $x[n]$ and some other infinite-length vector, e.g. $e^{j\omega n}$ for the DTFT. We can make a similar claim about the DFT, except the DFT is realizable with an actual operator matrix that describes the transform. First, let's look at the DFT again:

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi k}{N}n}.$$

We can say then that $X[k]$ is given by the following dot product:

$$X[k] = v^* u, \ \ u, v, \in \mathbb{C}^N \tag{32}$$
$$u = x[n] \tag{33}$$
$$v = e^{j\frac{2\pi k}{N}n}. \tag{34}$$

Remember that dot products on complex vectors requires the conjugate transpose, v^*, of the second vector. For example, let's compute $X[3]$ for length $N = 5$:

$$X[3] = \begin{bmatrix} 1 & e^{-j\frac{2\pi\cdot 3}{5}\cdot 1} & e^{-j\frac{2\pi\cdot 3}{5}\cdot 2} & e^{-j\frac{2\pi\cdot 3}{5}\cdot 3} & e^{-j\frac{2\pi\cdot 3}{5}\cdot 4} \end{bmatrix} \begin{bmatrix} x[0] \\ x[1] \\ x[2] \\ x[3] \\ x[4] \end{bmatrix}. \tag{35}$$

We can expand this technique to all possible values of $k = \{0, 1, 2, \ldots, N-1\}$. Before doing so, we will define a little bit of notation to make things cleaner. Let

$$W = e^{-j2\pi} \tag{36}$$
$$W_N = e^{-j\frac{2\pi}{N}} \tag{37}$$
$$W_N^{kn} = e^{-j\frac{2\pi k}{N}n}. \tag{38}$$

We refer to W as the *twiddle factor*. We can then write the DFT as

$$\mathbf{X}[k] = \mathbf{W}\mathbf{x} \tag{39}$$

$$\mathbf{W} = \begin{bmatrix} 1 & 1 & 1 & 1 & \cdots & 1 & 1 \\ 1 & W_N^1 & W_N^2 & W_N^3 & \cdots & W_N^{N-2} & W_N^{N-1} \\ 1 & W_N^2 & W_N^4 & W_N^6 & \cdots & W_N^{2(N-2)} & W_N^{2(N-1)} \\ 1 & W_N^3 & W_N^6 & W_N^9 & \cdots & W_N^{3(N-2)} & W_N^{3(N-1)} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & W_N^{N-2} & W_N^{(N-2)\cdot 2} & W_N^{(N-2)\cdot 3} & \cdots & W_N^{(N-2)(N-2)} & W_N^{(N-2)(N-1)} \\ 1 & W_N^{N-1} & W_N^{(N-1)\cdot 2} & W_N^{(N-1)\cdot 3} & \cdots & W_N^{(N-1)(N-2)} & W_N^{(N-1)(N-1)} \end{bmatrix}_{N\times N} \tag{40}$$

Above, $\mathbf{W}$ is the *DFT operator matrix* for some length N. We can then use $\mathbf{W}$ and a vectorized version of $x[n]$, $\mathbf{x}$, to compute the DFT:

$$\mathbf{X}[k] = \begin{bmatrix} 1 & 1 & 1 & 1 & \cdots & 1 & 1 \\ 1 & W_N^1 & W_N^2 & W_N^3 & \cdots & W_N^{N-2} & W_N^{N-1} \\ 1 & W_N^2 & W_N^4 & W_N^6 & \cdots & W_N^{2(N-2)} & W_N^{2(N-1)} \\ 1 & W_N^3 & W_N^6 & W_N^9 & \cdots & W_N^{3(N-2)} & W_N^{3(N-1)} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & W_N^{N-2} & W_N^{(N-2)\cdot 2} & W_N^{(N-2)\cdot 3} & \cdots & W_N^{(N-2)(N-2)} & W_N^{(N-2)(N-1)} \\ 1 & W_N^{N-1} & W_N^{(N-1)\cdot 2} & W_N^{(N-1)\cdot 3} & \cdots & W_N^{(N-1)(N-2)} & W_N^{(N-1)(N-1)} \end{bmatrix} \begin{bmatrix} x[0] \\ x[1] \\ x[2] \\ x[3] \\ \vdots \\ x[N-2] \\ x[N-1] \end{bmatrix} \tag{41}$$

This is a lot to look at, but we can build up our understanding from smaller pieces. Each element in $\mathbf{W}$ can be expressed as

$$\mathbf{W}_{kn} = W_N^{kn}, \tag{42}$$

where k is the row and n is the column of $\mathbf{W}$. Note how the twiddle factors in each row have exponents that increase by multiples of the row index k. Each row in $\mathbf{W}$ describes a discrete-time periodic complex exponential of the form $e^{-j\omega_k n}$ where

$$\omega_k = \frac{2\pi k}{N},\ 0 \le k \le N-1.$$

Thus, each row of $\mathbf{W}$ is a periodic complex exponential rotating with frequency ω_k as the columns increase the value of n. If we let $N = 5$ and $k = 3$, we can see how the fourth row (where $k = 3$) in $\mathbf{W}$ gives us our example from line 35 above.

We can also define the inverse DFT by identifying $\mathbf{W}^{-1}$ since

$$\mathbf{x} = \mathbf{W}^{-1}\mathbf{X}[k] \tag{43}$$

$$= \mathbf{W}^{-1}\mathbf{W}\mathbf{x}. \tag{44}$$

For conciseness, we will simply state that the *inverse DFT operator matrix* is given by

$$\mathbf{W}^{-1} = \frac{1}{N}\mathbf{W}^*. \tag{45}$$

Again, remember that $\mathbf{W}^*$ denotes the conjugate transpose, or Hermitian transpose, of $\mathbf{W}$.

Finally, we should discuss why the linear algebra formulation of the DFT is relevant. First, we can gain a great amount of intuition regarding the DFT from it. We plainly see how each entry in $X[k]$ is the dot product between $x[n]$ and the periodic complex exponential described by row k in $\mathbf{W}$. We can also make more significant claims about the DFT that are beyond the scope of this course, e.g. refer to the below lecture exercise. Second, computing the DFT and inverse DFT using such a matrix-vector multiplication formulation is more efficient in practice because (1) we can pre-compute and store $\mathbf{W}$ and $\mathbf{W}^{-1}$, and (2) matrices and vectors are computationally friendly. Following the DFT summation formula may encourage something like a "double for-loop" in computer code to compute the length-N summation for all N values of k. Matrix operations, on the other hand, are highly efficient. Matrices and vectors are commonly stored in contiguous memory using arrays and many programming languages have optimized implementations of fundamental matrix operations like matrix-vector multiplication that save on much of the overhead that a "for-loop" implementation carries.

Lecture exercise: Discuss how the columns of $\mathbf{W}$ form an orthogonal set of basis vectors for $\mathbb{C}^N$.