6.300: Signal Processing

Discrete Fourier Transform (DFT)

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

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

Relating the DFT to DTFS and DTFT:

- DTFS of periodically-extended $x_w[n] = x[n]w[n]$
- Sampled $(\Omega \to 2\pi k/N)$, scaled $(1/N)$ DTFT of $x_w[n]$

Frequency resolution: $\Delta f = f_s/N$ and $\Delta\Omega = 2\pi/N$

March 31, 2026

Agenda for Recitation

- Fourier representations (so far)
- Spectral analysis with the DFT

What questions do you have from lecture?

Agenda for Recitation

- Fourier representations (so far)
- Spectral analysis with the DFT

Fourier Representations

	type of time-domain signal			domain	period
CTFS	CT	period T	infinite length	integer k	none
DTFS	DT	period N	infinite length	integer k	N
CTFT	CT	aperiodic	infinite length	real ω	none
DTFT	DT	aperiodic	infinite length	real Ω	2π
DFT	DT	aperiodic	length N	integer k	N

The **discrete Fourier transform (DFT)** is a discrete-time, discrete-frequency Fourier transform.

- for aperiodic discrete-time (n) signals $x[n]$
- yields discrete-frequency (k) representation $X[k]$
- finite length (N) in both time and frequency
- closely related to DTFS and DTFT

Discrete-Time Fourier Representations

Another Fourier transform? Why?

Discrete-Time Fourier Representations

Another Fourier transform? Why?

Discrete-time Fourier series (**DTFS**) are conceptually simple, but their use is limited to periodic signals.

The discrete-time Fourier transform (**DTFT**) works for arbitrary discrete-time signals. The problem is that the frequency Ω is of continuous domain — and this is not amenable for numerical computation.

The discrete Fourier transform (**DFT**) works for arbitrary discrete-time signals, too. There are a finite set of frequencies $k \in [0, N-1]$, which makes the DFT amenable for numerical computation.

All of our work on Fourier representations in the course has built to this — understanding the DFT!

DFT: Periodic Extension

Relation to DTFS: The DFT yields the DTFS coefficients for a periodically-extended $x_w[n] = x[n]w[n]$.

DTFS: $X[k] = \frac{1}{N} \sum_{n=\langle N \rangle} x[n] e^{-jk\frac{2\pi}{N}n}$ **DFT:** $X[k] = \frac{1}{N} \sum_{n=0}^{N-1} x[n] e^{-jk\frac{2\pi}{N}n}$

Let $N = 32$. How many of the following signals have "simple" DFTs (i.e., mostly zeros)?

- $x_1[n] = \cos\left(\frac{2\pi}{32}n\right)$
- $x_2[n] = \cos\left(\frac{2\pi}{64}n\right)$
- $x_3[n] = \cos\left(\frac{2\pi}{16}n - \frac{\pi}{2}\right)$
- $x_4[n] = \cos\left(\frac{2\pi}{4}n\right)\cos\left(\frac{2\pi}{8}n\right)$
- $x_5[n] = \cos\left(\frac{2\pi}{32}n\right) + j\sin\left(\frac{2\pi}{32}n\right)$

DFT: Periodic Extension

Relation to DTFS: The DFT yields the DTFS coefficients for a periodically-extended $x_w[n] = x[n]w[n]$.

DTFS: $X[k] = \frac{1}{N}\sum_{n=\langle N\rangle} x[n]e^{-jk\frac{2\pi}{N}n}$ **DFT:** $X[k] = \frac{1}{N}\sum_{n=0}^{N-1} x[n]e^{-jk\frac{2\pi}{N}n}$

Let $N = 32$. How many of the following signals have "simple" DFTs (i.e., mostly zeros)? **4**

- $x_1[n] = \cos\left(\frac{2\pi}{32}n\right) = \frac{1}{2}e^{j\frac{2\pi}{32}n} +$ c.c. (complex conjugate)
- $x_2[n] = \cos\left(\frac{2\pi}{64}n\right)$
- $x_3[n] = \cos\left(\frac{2\pi}{16}n - \frac{\pi}{2}\right) = \frac{1}{2}e^{-j\frac{\pi}{2}}e^{j2\frac{2\pi}{32}n} +$ c.c.
- $x_4[n] = \cos\left(\frac{2\pi}{4}n\right)\cos\left(\frac{2\pi}{8}n\right) = \frac{1}{4}e^{j4\frac{2\pi}{32}n} + \frac{1}{4}e^{j12\frac{2\pi}{32}n} +$ c.c.
- $x_5[n] = \cos\left(\frac{2\pi}{32}n\right) + j\sin\left(\frac{2\pi}{32}n\right) = e^{j\frac{2\pi}{32}n}$

DFT: Sampling the DTFT

Relation to DTFT: Multiplication by a window function $w[n]$ in time corresponds to convolution with the Fourier transform of the window function, $W(\Omega)$, in frequency.

$$x_w[n] = x[n]w[n] \iff X_w(\Omega) = \frac{1}{2\pi}(X * W)(\Omega)$$

The DFT is a sampled ($\Omega \to \frac{2\pi}{N}k$), scaled ($1/N$) DTFT.

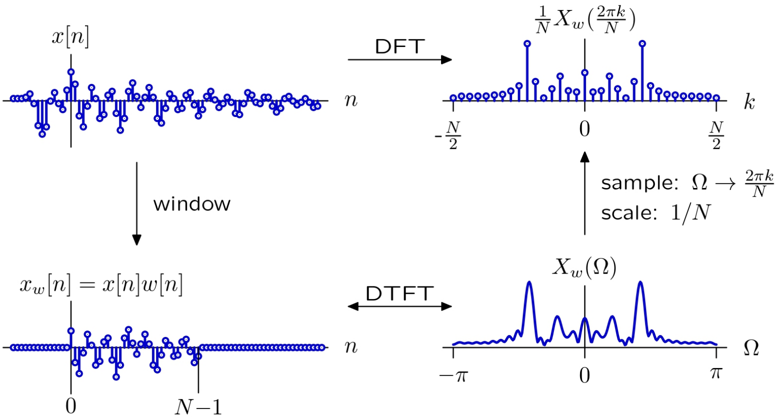
$$X[k] = \frac{1}{N} X_w(\frac{2\pi}{N}k)$$

Notice that the distance between adjacent DFT frequency samples (or "bins") is $\Delta\Omega = 2\pi/N$ radians per sample. Accordingly, as N increases, $\Delta\Omega$ decreases, and this yields finer frequency resolution.[1]

[1] In general, the length of the DFT analysis window (N) can exceed the length of the data. Ignore this case ("zero-padding") for now.

Relation Between DFT and DTFT

Graphical depiction of relation between DFT and DTFT.



Graphic: Professor Denny Freeman (freeman@mit.edu)

Agenda for Recitation

- Fourier representations (so far)
- Spectral analysis with the DFT

Agenda for Recitation

- Fourier representations (so far)
- Spectral analysis with the DFT

Spectral Analysis with the DFT

Consider the four discrete-time signals below.

$$
\begin{aligned}
x_1[n] &= \cos\left(\tfrac{8\pi}{100} n\right) \\
x_2[n] &= \cos\left(\tfrac{8\pi}{100} n - \tfrac{\pi}{4}\right) \\
x_3[n] &= \cos\left(\tfrac{9\pi}{100} n\right) \\
x_4[n] &= \cos\left(\tfrac{9\pi}{100} n - \tfrac{\pi}{2}\right)
\end{aligned}
$$

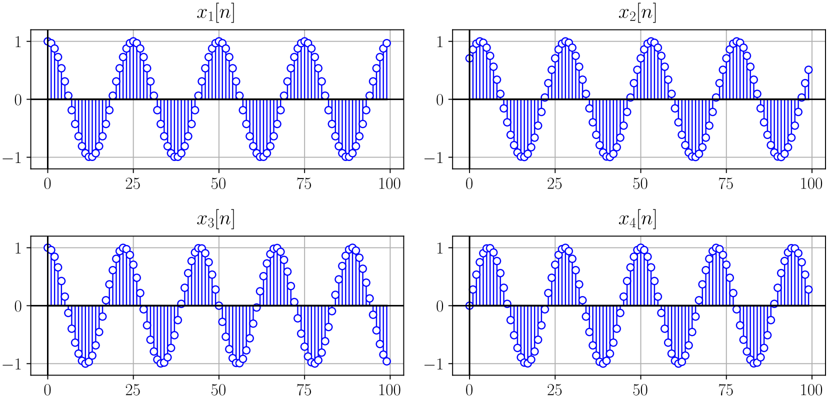
Suppose that the sampling rate is $\texttt{fs} = 44100$ samples per second. How would you write a Python program to generate a second's worth of samples for each of the signals above?

Spectral Analysis with the DFT

Suppose that the sampling rate is fs $= 44100$ samples per second. How would you write a Python program to generate a second's worth of samples for each of the signals on the previous slide?

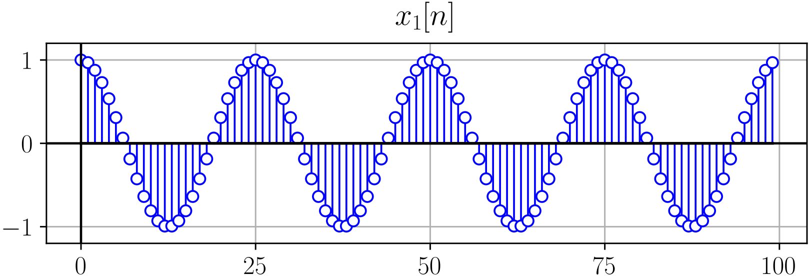
```
from math import cos, pi

fs = 44100 # Sampling rate (samples per second)
num_seconds = 1 # Duration (seconds)
num_samples = int(fs * num_seconds) # Number of samples

x1 = [cos(8*pi*n/100) for n in range(num_samples)]
x2 = [cos(8*pi*n/100-pi/4) for n in range(num_samples)]
x3 = [cos(9*pi*n/100) for n in range(num_samples)]
x4 = [cos(8*pi*n/100-pi/2) for n in range(num_samples)]
```

Spectral Analysis with the DFT



Spectral Analysis with the DFT

The first $N = 100$ samples of $x_1[n]$ are plotted below.



Recall that the sampling rate is $\texttt{fs} = 44100$ samples per second. What continuous-time cyclical frequency f_0 (cycles per second) does discrete-time angular frequency $\Omega_0 = \frac{8\pi}{100}$ (radians per sample) correspond to?

Spectral Analysis with the DFT

Recall that the sampling rate is $\texttt{fs} = 44100$ samples per second. What continuous-time cyclical frequency f_0 (cycles per second) does discrete-time angular frequency $\Omega_0 = \frac{8\pi}{100}$ (radians per sample) correspond to?

Use proportional reasoning.

$$\frac{f_0}{f_s} = \frac{\Omega_0}{2\pi}$$
$$\frac{f_0}{44100} = \frac{\left(\frac{8\pi}{100}\right)}{2\pi}$$

Solve for f_0 to determine that $f_0 = 1764$ cycles per second (Hz) is the continuous-time cyclical frequency.

How do we generalize this?

Spectral Analysis with the DFT

To map a DFT frequency index (or "bin") k to the corresponding continuous-time frequency f_0, we use

$$f_0 = \frac{k}{N} f_s$$

for $k \in [0, \frac{1}{2}N]$, where N is the DFT analysis window length. Remember that $k \in \left(\frac{1}{2}N, N-1\right]$ correspond to negative frequencies.

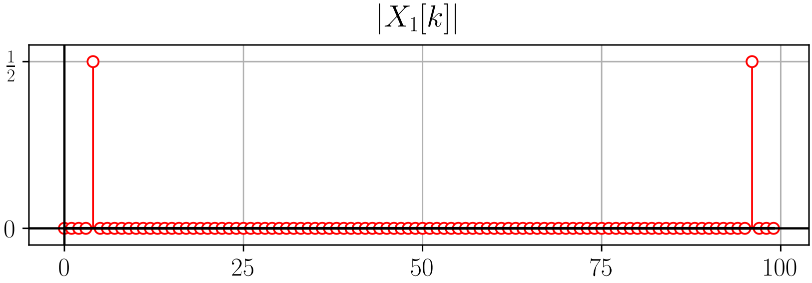
As N increases, we get finer frequency resolution: $\Delta f \to 0$.

$$f_0 = k\,\Delta f \text{ where } \Delta f \triangleq \frac{f_s}{N} \text{ hertz per bin}$$

We cannot distinguish between frequencies which are closer than Δf. Such frequencies are "smeared" together in the DFT magnitude plot.

Spectral Analysis with the DFT

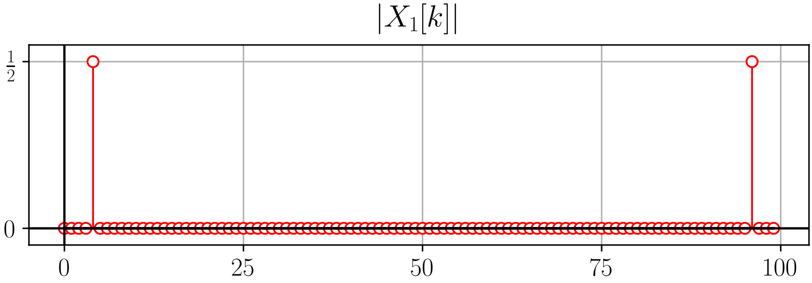
Shown below is $|X_1[k]|$, the magnitude of the DFT of the first $N = 100$ samples of $x_1[n]$.



For which values of k is $|X_1[k]|$ non-zero? Explain.

Spectral Analysis with the DFT

Shown below is $|X_1[k]|$, the magnitude of the DFT of the first $N = 100$ samples of $x_1[n]$.

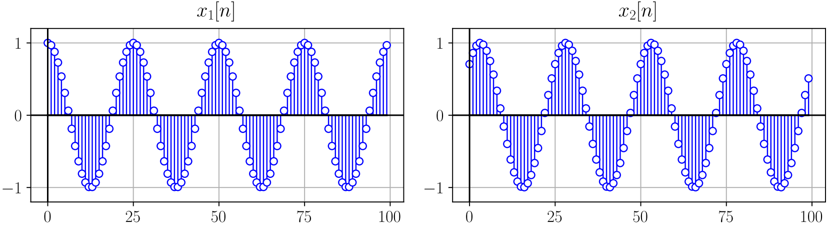


For which values of k is $|X_1[k]|$ non-zero? Explain.

$x_1[n] = \cos\left(4\frac{2\pi}{100}n\right) = \frac{1}{2}e^{j4\frac{2\pi}{100}n} + \frac{1}{2}e^{-j4\frac{2\pi}{100}n} \implies k = 4, 96$ or $k = \pm 4$

Spectral Analysis with the DFT

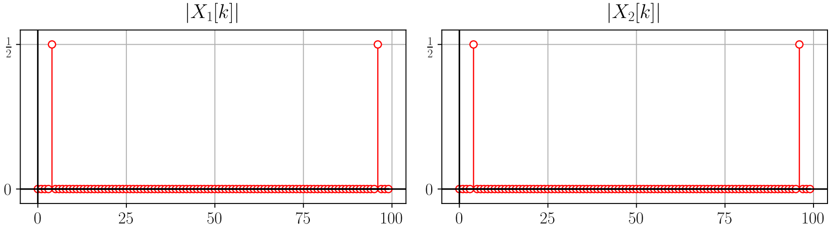
The first $N = 100$ samples of $x_1[n]$ and $x_2[n]$ are plotted below.



How will $|X_1[k]|$ and $|X_2[k]|$ differ?

Spectral Analysis with the DFT

How will $|X_1[k]|$ and $|X_2[k]|$ differ?

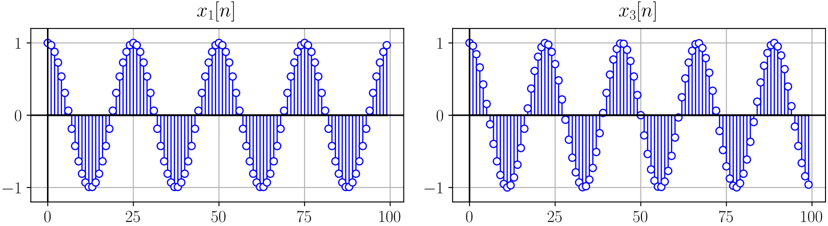


Both $x_1[n]$ and $x_2[n]$ are periodic in $N = 100$, and both $x_1[n]$ and $x_2[n]$ complete 4 full cycles in $N = 100$ samples. So, $|X_1[k]|$ and $|X_2[k]|$ appear identical.

However, $X_1[k] \neq X_2[k]$ because $\angle X_1[k] \neq \angle X_2[k]$.
Why? $x_1[n]$ and $x_2[n]$ are offset in time.

Spectral Analysis with the DFT

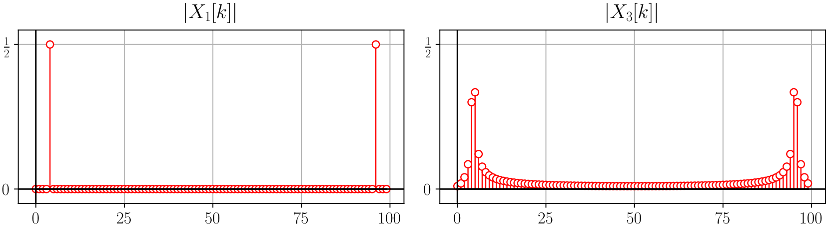
The first $N = 100$ samples of $x_1[n]$ and $x_3[n]$ are plotted below.



How will $|X_1[k]|$ and $|X_3[k]|$ differ?

Spectral Analysis with the DFT

How will $|X_1[k]|$ and $|X_3[k]|$ differ?

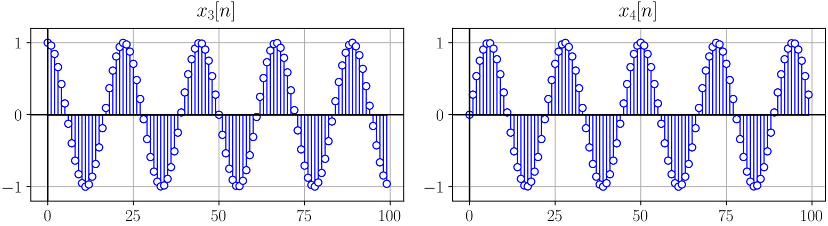


$|X_1[k]|$ and $|X_3[k]|$ are very different! Why? For one, $x_3[n]$ is not periodic in $N = 100$. The peak of $|X_3(\Omega)|$ doesn't fall on an integer value of k.

$$x_3[n] = \cos\left(\frac{9\pi}{100}n\right) = \cos\left(4.5\frac{2\pi}{100}\right) \implies k = \pm 4.5 \text{ (???)}$$

Spectral Analysis with the DFT

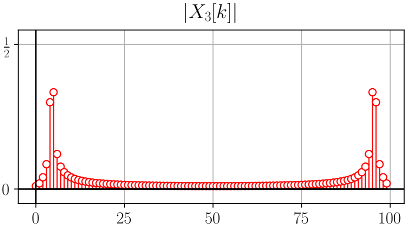
The first $N = 100$ samples of $x_3[n]$ and $x_4[n]$ are plotted below.

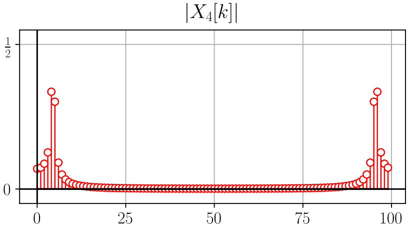


How will $|X_3[k]|$ and $|X_4[k]|$ differ?

Spectral Analysis with the DFT

How will $|X_3[k]|$ and $|X_4[k]|$ differ?

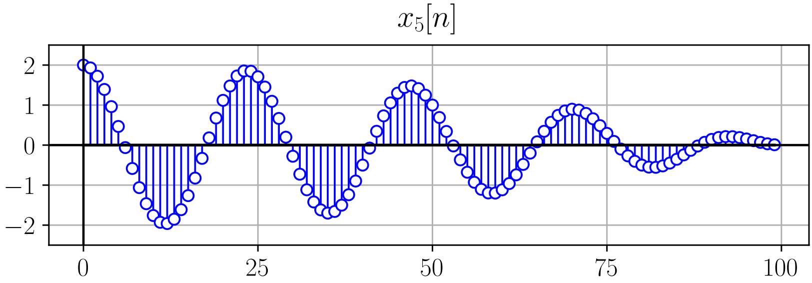




The average ("DC") value of $x_3[n]$ is approximately zero, while the DC value of $x_4[n]$ is non-negative. Notice that $x_4[n]$ has 5 positive half cycles and 4 negative half cycles.

Spectral Analysis with the DFT

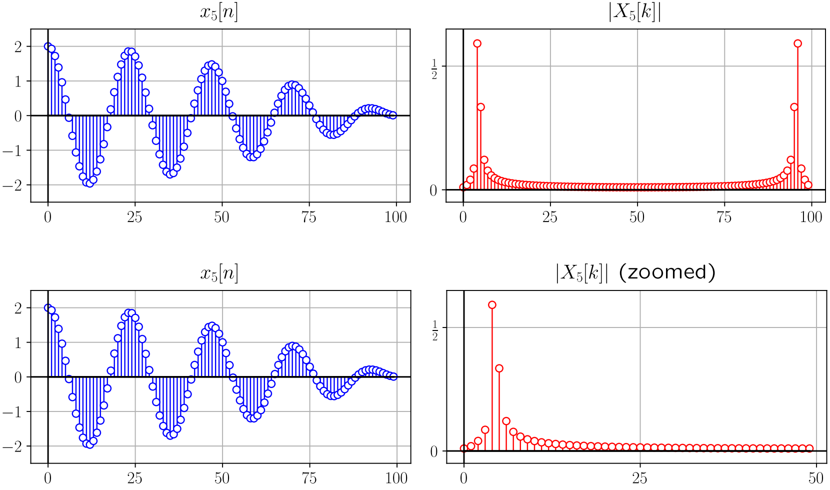
Consider $x_5[n] = \cos\left(\frac{8\pi}{100}n\right) + \cos\left(\frac{9\pi}{100}n\right)$. The first $N = 100$ samples are plotted below.



What is the minimum analysis window length N that's needed to resolve $\Omega_1 = \frac{8\pi}{100}$ from $\Omega_2 = \frac{9\pi}{100}$?

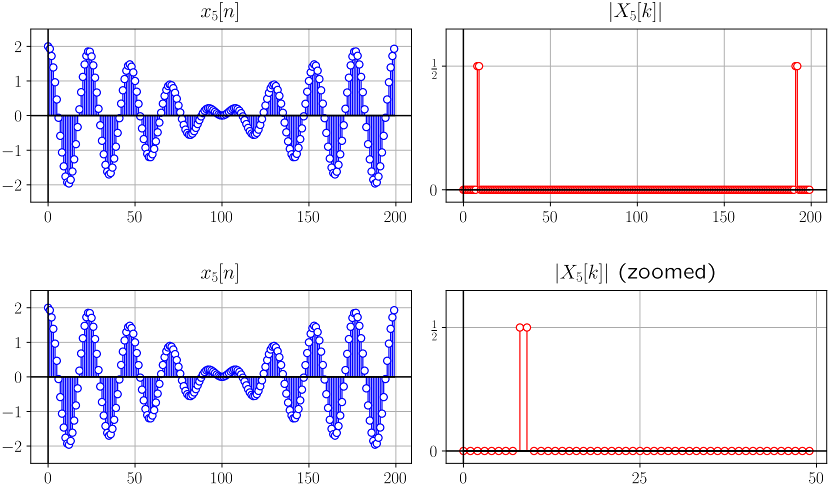
Spectral Analysis with the DFT

Use an analysis window of length $N = 100$.



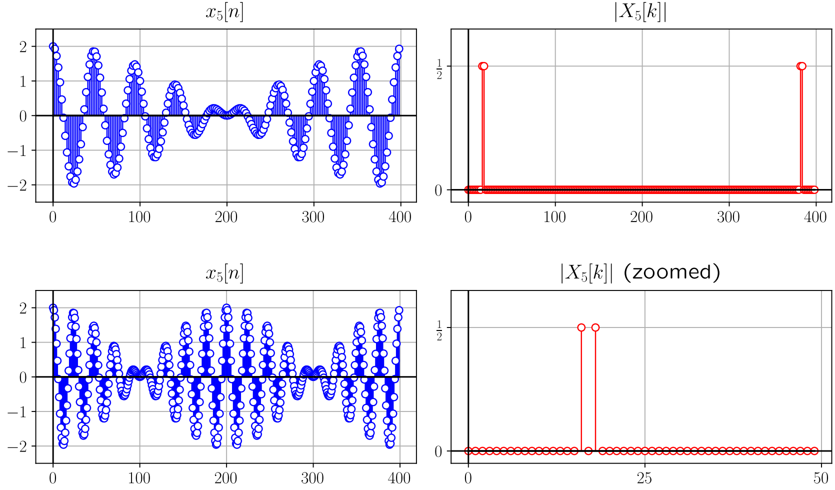
Spectral Analysis with the DFT

Use an analysis window of length $N = 200$.



Spectral Analysis with the DFT

Use an analysis window of length $N = 400$.



Spectral Analysis with the DFT

Two frequencies Ω_1, Ω_2 (radians per sample) are resolved if

$$|\Omega_1 - \Omega_2| > \Delta\Omega = \frac{2\pi}{N} \text{ radians per bin.}$$

Likewise, two frequencies f_1, f_2 (hertz) are resolved if

$$|f_1 - f_2| > \Delta f = \frac{f_s}{N} \text{ hertz per bin.}$$

"Good" frequency resolution means "fine" resolution.
$\implies \Delta\Omega, \Delta f$ are (relatively) small.

"Bad" frequency resolution means "coarse" resolution.
$\implies \Delta\Omega, \Delta f$ are (relatively) big.

Of course, "small" and "big" are relative terms. What they mean depends on the application.

Spectral Analysis with the DFT

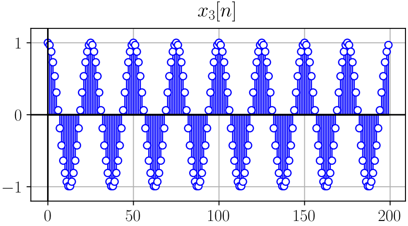
To resolve $\Omega_1 = 8\pi/100$ and $\Omega_2 = 9\pi/100$, the frequency resolution must be

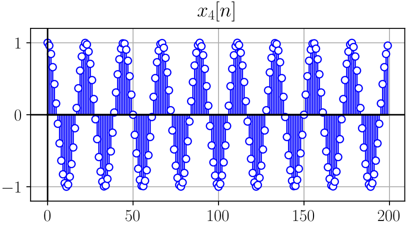
$$\Delta\Omega < \pi/100,$$

so we must use an analysis window of length

$$N > 200$$

samples. Notice that 8 full cycles of $\cos(\Omega_1 n)$ and 9 full cycles of $\cos(\Omega_2 n)$ fit into a DFT analysis window of length $N = 200$.





Spectral Analysis with the DFT

Frequency resolution is proportional to $1/N$. Suppose that we append a bunch of zeros to the end of our finite-length signal to make it longer. This increases N. Does our frequency resolution improve? Explain.

Spectral Analysis with the DFT

Frequency resolution is proportional to $1/N$. Suppose that we append a bunch of zeros to the end of our finite-length signal to make it longer. This increases N. Does our frequency resolution improve? Explain.

Today, we've assumed that we have access to as much data as we want. Consequently, N, the length of the DFT analysis window, determined the frequency resolution.

Oftentimes, however, we have access to only a limited amount of data. In these cases, it is the **length of the data**, not the length of the DFT analysis window, that determines the frequency resolution. Intuitively, **zero-padding** — adding "zero information" — ought not increase the frequency resolution in any meaningful sense.

Lessons Learned

The **discrete Fourier transform (DFT)** is a discrete-time, discrete-frequency Fourier transform.

- for aperiodic discrete-time (n) signals $x[n]$
- yields discrete-frequency (k) representation $X[k]$
- finite length (N) in both time and frequency
- closely related to DTFS and DTFT

Relating the DFT to DTFS and DTFT:

- DTFS of periodically-extended $x_w[n] = x[n]w[n]$
- Sampled ($\Omega \to 2\pi k/N$), scaled ($1/N$) DTFT of $x_w[n]$

Frequency resolution: $\Delta f = f_s/N$ and $\Delta\Omega = 2\pi/N$

Question of the Day

Imagine that we're tuning a musical instrument. Suppose that our tuner does the processing digitally, and it samples at rate $f_s = 44.1$ kHz. For our tuner to have frequency resolution $\Delta f \leq 0.5$ Hz, how long must N, the analysis window length, be? How many seconds does your choice of N correspond to? (That is, how long until your tuner can estimate what note you played? Is that fast enough for you?)