6.3000: Signal Processing

Discrete Fourier Transform

	analysis	synthesis
DFT:	$X[k] = \frac{1}{N}\sum_{n=0}^{N-1} x[n]e^{-j\frac{2\pi k}{N}n}$	$x[n] = \sum_{k=0}^{N-1} X[k]e^{j\frac{2\pi k}{N}n}$
DTFS:	$X[k] = \frac{1}{N}\sum_{n=\langle N\rangle} x[n]e^{-j\frac{2\pi k}{N}n}$	$x[n] = \sum_{k=\langle N\rangle} X[k]e^{j\frac{2\pi k}{N}n}$
DTFT:	$X(\Omega) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\Omega n}$	$x[n] = \frac{1}{2\pi}\int_{2\pi} X(\Omega)e^{j\Omega n}d\Omega$

March 18, 2025

Single Sinusoid

Create four signals

$$x_1[n] = \cos(8\pi n/100)$$
$$x_2[n] = \cos(8\pi n/100 - \pi/4)$$
$$x_3[n] = \cos(9\pi n/100)$$
$$x_4[n] = \cos(9\pi n/100 - \pi/2)$$

each with a duration of 1 second when the sample frequency is $44,100$ Hz.

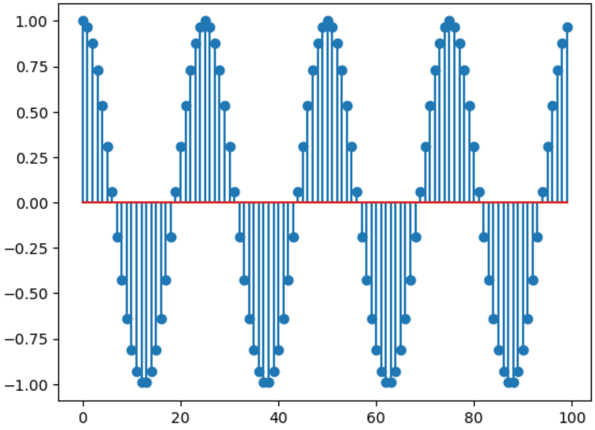
Compare the DFTs of the first 100 samples of each of these signals.

Python Code

```
from math import cos, pi
from lib6003.audio import wav_write
from matplotlib.pyplot import stem, show

fs = 44100
x1 = [cos(8*pi*n/100) for n in range(fs)]
x2 = [cos(8*pi*n/100-pi/4) for n in range(fs)]
x3 = [cos(9*pi*n/100) for n in range(fs)]
x4 = [cos(9*pi*n/100-pi/2) for n in range(fs)]

wav_write(x1,fs,'x1.wav')
wav_write(x2,fs,'x2.wav')
wav_write(x3,fs,'x3.wav')
wav_write(x4,fs,'x4.wav')

stem(x1[0:100])
show()
```

Single Sinusoid

$$x_1[n] = \cos(8\pi n/100)$$



What is the frequency of this tone if the sample rate is $44,100$ Hz?

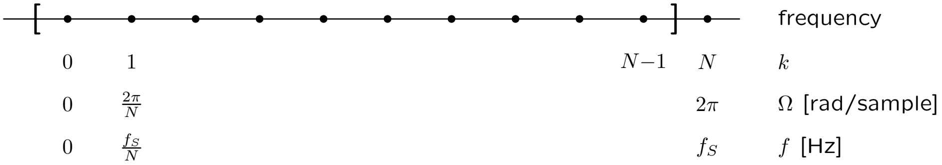
Single Sinusoid

What is the frequency of the tone generated by $x_1[n]$?

Since $x_1[n] = \cos(8\pi n/100)$, we know that the discrete frequency $\Omega_1 = \frac{8\pi}{100}$. Furthermore, the sample frequency $f = f_s$ corresponds to the maximum discrete frequency $\Omega = 2\pi$, and frequencies f in Hz are proportional to discrete frequencies Ω.

$$\frac{f}{f_s} = \frac{\Omega}{2\pi}$$

So $f = \frac{\Omega f_s}{2\pi} = \frac{8\pi/100}{2\pi} \times 44,100\,\text{Hz} = 1764\,\text{Hz}$



Single Sinusoid

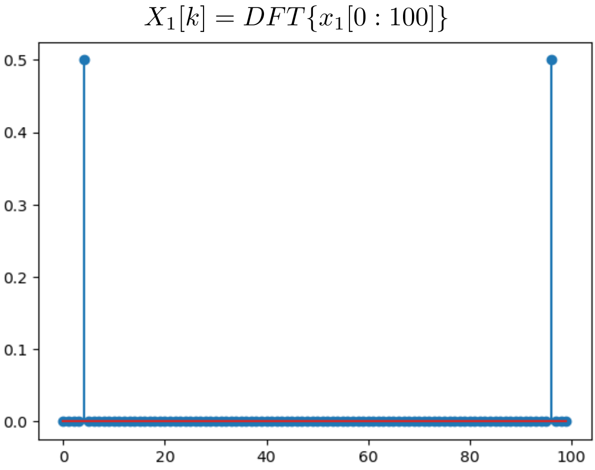
Write a program to calculate the DFT of an input sequence.

Use that program to calculate $X_1[k]$, which is the DFT of the first 100 samples of $x_1[n]$.

```
def dft(x):
    N = len(x)
    answer = [0 for k in range(N)]
    for k in range(N):
        for n in range(N):
            answer[k] += (1/N)*x[n]*e**(-2j*pi*k*n/N)
    return answer

X1 = dft(x1[0:100])
```

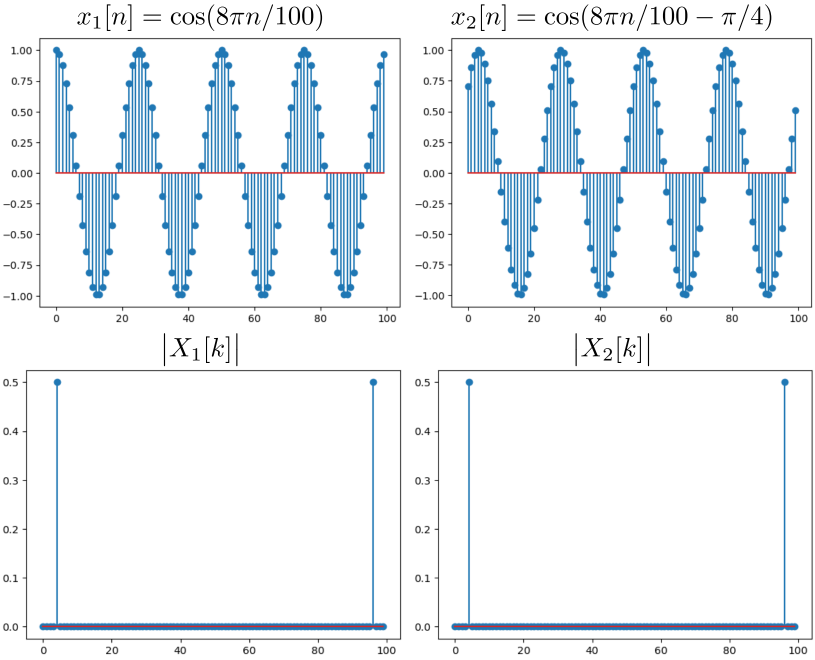
Single Sinusoid

Plot the magnitude of $X_1[\cdot]$.



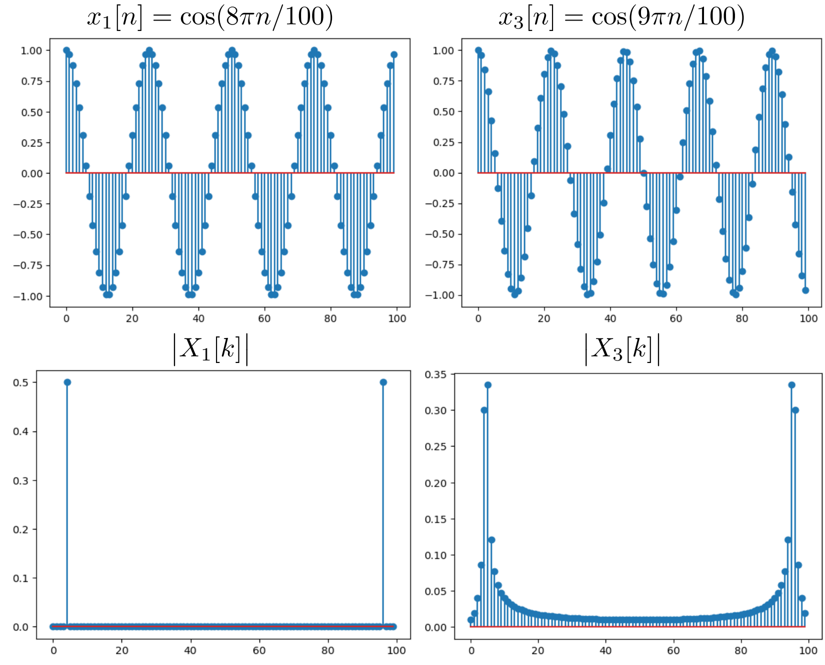
Which values of k are non-zero?

Compare Two Signals



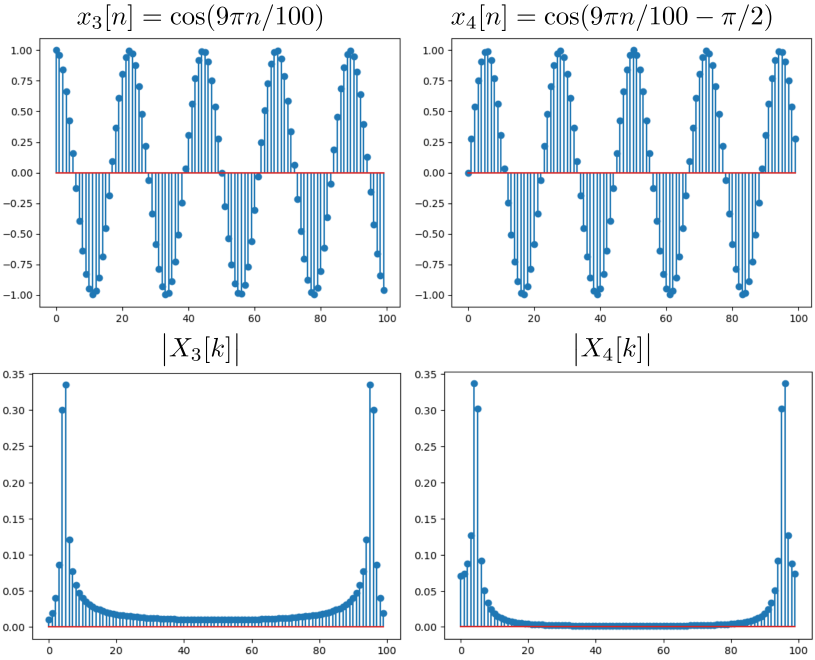
No difference in magnitudes (but the phases are different).

Compare Two Signals



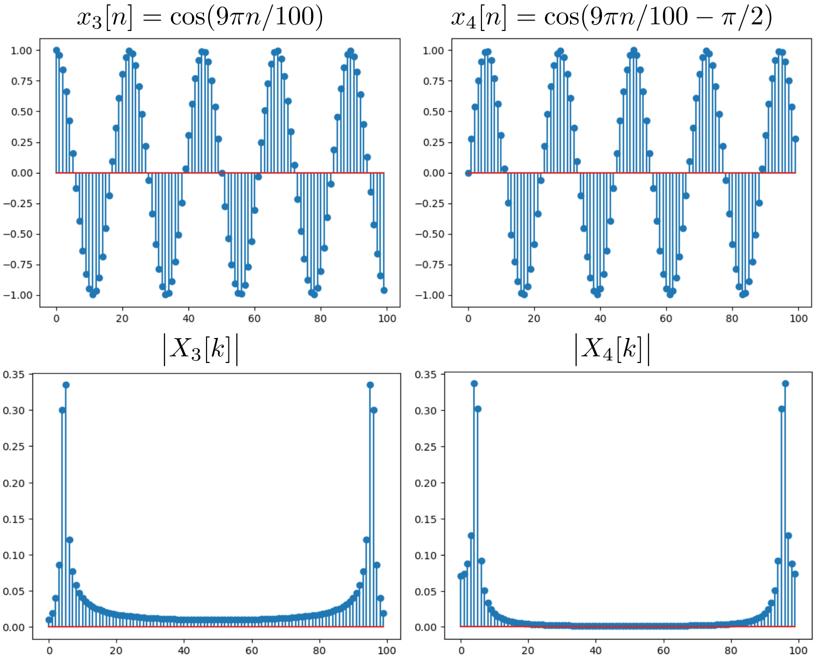
Why are these DFTs so different?

Compare Two Signals



$\Omega_3 = \Omega_4$. But DC bigger: 5 positive half cycles versus 4 negative ones.

Compare Two Signals

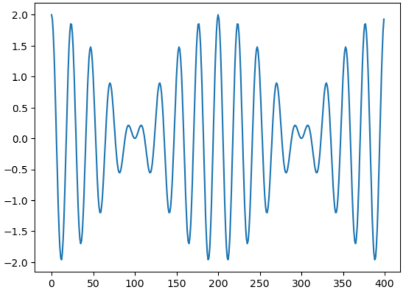


High freq. content of X_4 smaller than X_3: $|x_4[99] - x_4[0]| < |x_3[99] - x_3[0]|$

Analyzing Signals with Multiple Frequencies

What is the minimum window size N needed to resolve $\Omega = 8\pi/100$ from $9\pi/100$?

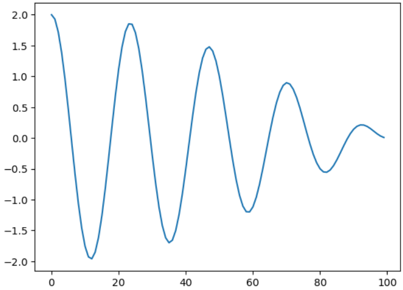
$$x_5[n] = \cos(8\pi n/100) + \cos(9\pi n/100)$$



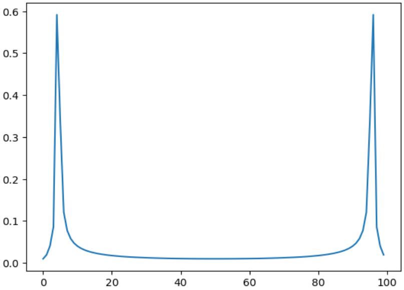
Analyzing Signals with Multiple Frequencies

If the analysis window is small (here $N{=}100$), the two frequencies $8\pi/100$ and $9\pi/100$ generate a single peak in the DFT at $k = 4$ (along with its partner at $k = 100{-}4 = 96$).

$x_5[n] = \cos(8\pi n/100) + \cos(9\pi n/100)$



$\left|X_5[k]\right|$

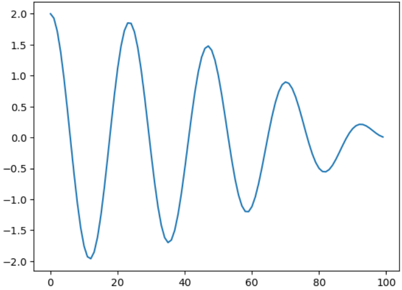


Analyzing Signals with Multiple Frequencies

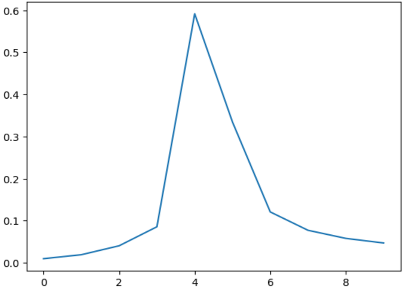
Two frequencies can look like one if analysis window is too small.

$N = 100$ zoomed

$x_5[n] = \cos(8\pi n/100) + \cos(9\pi n/100)$



$\left|X_5[k]\right|$

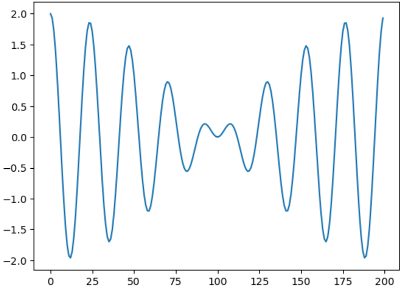


Analyzing Signals with Multiple Frequencies

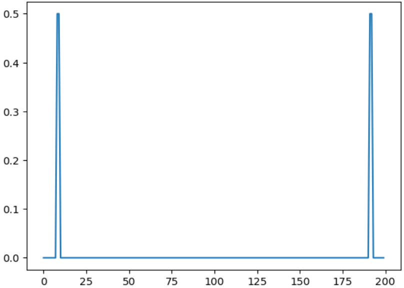
Two frequencies can look like one if analysis window is too small.

$N = 200$

$x_5[n] = \cos(8\pi n/100) + \cos(9\pi n/100)$



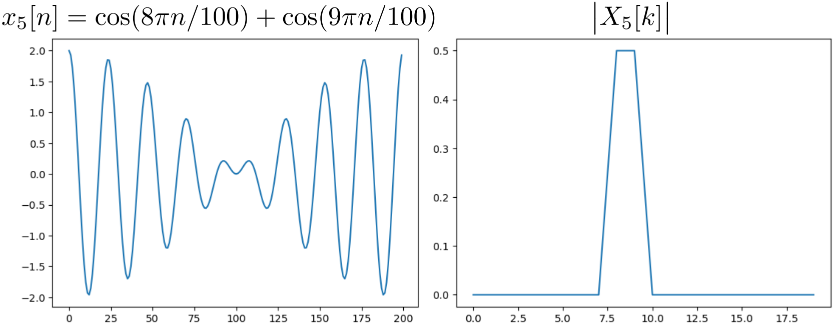
$\left|X_5[k]\right|$



Analyzing Signals with Multiple Frequencies

Two frequencies can look like one if analysis window is too small.

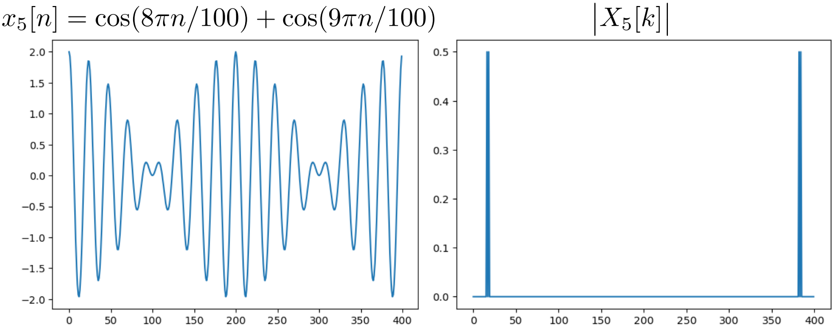
$N = 200$ zoomed



Analyzing Signals with Multiple Frequencies

Two frequencies can look like one if analysis window is too small.

$N = 400$

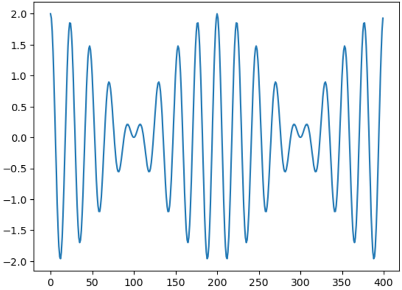


Analyzing Signals with Multiple Frequencies

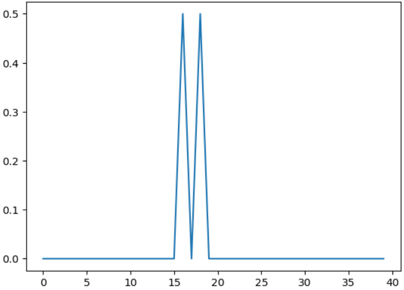
Two frequencies can look like one if analysis window is too small.

$N = 400$ zoomed

$x_5[n] = \cos(8\pi n/100) + \cos(9\pi n/100)$



$\left|X_5[k]\right|$

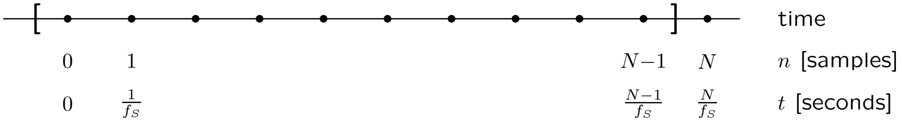


These frequencies are clearly resolved with $N = 400$.

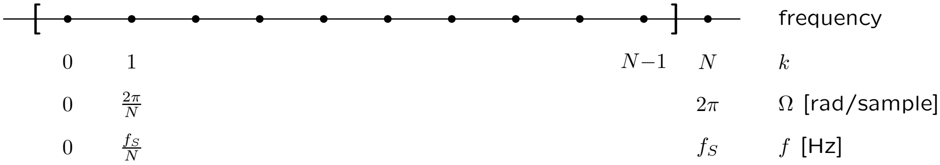
Frequency Scales

We can think of the DFT as having spectral resolution of $(2\pi/N)$ radians, which is equivalent to (f_S/N) Hz.

The time window is divided into N samples numbered $n = 0$ to $N-1$.



Discrete frequencies are similarly numbered as $k = 0$ to $N-1$.



Analyzing Signals with Multiple Frequencies

Two frequencies are resolved if they are separated by more than $\frac{2\pi}{N}$.

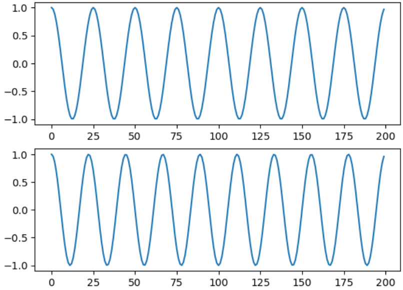
$\Omega_1 = \frac{8\pi}{100}$ and $\Omega_2 = \frac{9\pi}{100}$ will be resolved if

$$\Delta\Omega = \Omega_2 - \Omega_1 = \frac{9\pi}{100} - \frac{8\pi}{100} = \frac{\pi}{100} > \frac{2\pi}{N}$$

That is, if $N > 200$.

We can think of $\frac{2\pi}{N}$ as the frequency resolution of the DFT.

Notice 8 full cycles of Ω_1 and 9 full cycles of Ω_2 fit in $N = 200$.



Summary

Time and frequency resolution are important issues in all Fourier analyses.

Frequency resolution is determined by the number of samples N included in the analysis.