Lecture 4 & 6: Fourier Transform
Practice Exercises

Konstantinos Chatzilygeroudis
costashatz@upatras.gr

29 March 2026

1 Coding Exercises

Question 1.1

Let

$$x[n] = \cos(0.4\pi n), \qquad n = 0, 1, \dots, 63.$$

Approximate the DTFT numerically using `np.fft.fft` with a large FFT size $N = 2048$.
Use `np.fft.fftshift` to center the frequency axis, and plot

$$|X(e^{j\omega})| \quad \text{and} \quad \angle X(e^{j\omega})$$

versus $\omega \in [-\pi, \pi)$.
Then answer:

1. At which frequencies do the main peaks appear?
2. Why do two peaks appear?

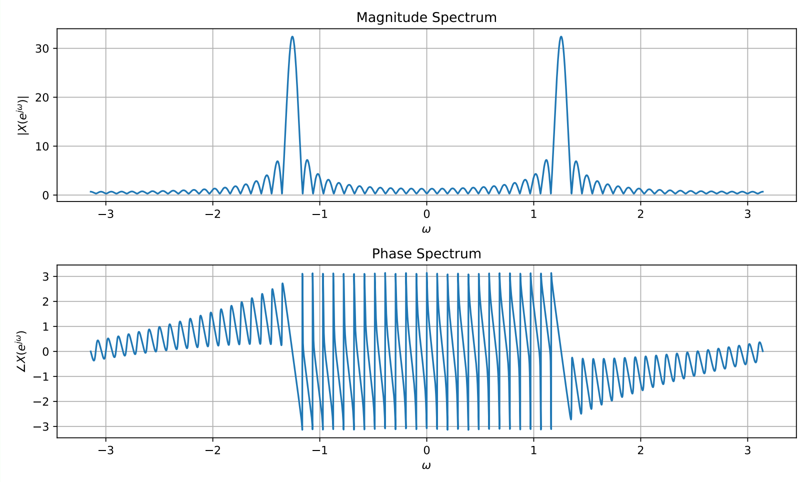
Solution (Code)

```python
import numpy as np
import matplotlib.pyplot as plt

n = np.arange(64)
x = np.cos(0.4 * np.pi * n)

N = 2048
X = np.fft.fft(x, N)
X = np.fft.fftshift(X)

omega = 2 * np.pi * np.fft.fftfreq(N, d=1)
omega = np.fft.fftshift(omega)

magX = np.abs(X)
phaseX = np.angle(X)

plt.figure(figsize=(10, 6))

plt.subplot(2, 1, 1)
plt.plot(omega, magX)
plt.xlabel(r'$\omega$')
plt.ylabel(r'$|X(e^{j\omega})|$')
plt.title('Magnitude Spectrum')
plt.grid(True)

plt.subplot(2, 1, 2)
plt.plot(omega, phaseX)
plt.xlabel(r'$\omega$')
plt.ylabel(r'$\angle X(e^{j\omega})$')
plt.title('Phase Spectrum')
```

```
plt.grid(True)

plt.tight_layout()
plt.show()
```

Solution

Plot:



Answers:

1. The main peaks appear near
$$\omega = \pm 0.4\pi.$$

2. Two peaks appear because
$$\cos(\omega_0 n) = \frac{1}{2}e^{j\omega_0 n} + \frac{1}{2}e^{-j\omega_0 n},$$
so the cosine contains both a positive-frequency and a negative-frequency component.

Question 1.2

Let the frequency-domain samples be defined by

$$X[k] = \begin{cases} 1, & k = 20 \\ 1, & k = N - 20 \\ 0, & \text{otherwise} \end{cases} \qquad \text{with } N = 256.$$

Use `np.ifft` to compute a time-domain sequence $x[n]$ from these frequency samples.
Plot
$$\mathrm{Re}\{x[n]\}, \qquad \mathrm{Im}\{x[n]\}$$
and also plot the magnitude and phase of the frequency samples $X[k]$.
Then answer:

1. Is the time-domain sequence real or complex?

2. What kind of oscillation do you observe in time?

3. Why do we place the nonzero values at symmetric frequency indices?

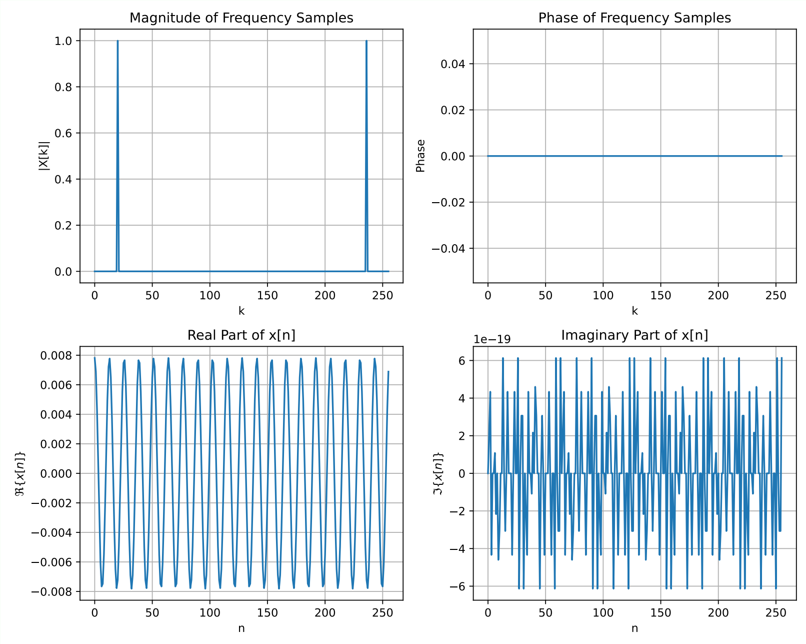
Solution (Code)

```
import numpy as np
import matplotlib.pyplot as plt

N = 256
```

```python
X = np.zeros(N, dtype=complex)

X[20] = 1
X[N - 20] = 1

x = np.fft.ifft(X)

k = np.arange(N)
n = np.arange(N)

magX = np.abs(X)
phaseX = np.angle(X)

plt.figure(figsize=(10, 8))

plt.subplot(2, 2, 1)
plt.plot(k, magX)
plt.xlabel('k')
plt.ylabel('|X[k]|')
plt.title('Magnitude of Frequency Samples')
plt.grid(True)

plt.subplot(2, 2, 2)
plt.plot(k, phaseX)
plt.xlabel('k')
plt.ylabel('Phase')
plt.title('Phase of Frequency Samples')
plt.grid(True)

plt.subplot(2, 2, 3)
plt.plot(n, np.real(x))
plt.xlabel('n')
plt.ylabel(r'$\Re\{x[n]\}$')
plt.title('Real Part of x[n]')
plt.grid(True)

plt.subplot(2, 2, 4)
plt.plot(n, np.imag(x))
plt.xlabel('n')
plt.ylabel(r'$\Im\{x[n]\}$')
plt.title('Imaginary Part of x[n]')
plt.grid(True)

plt.tight_layout()
plt.show()
```

Solution

Plot:



Answers:

1. The sequence is real, up to very small numerical roundoff errors, because the frequency samples are conjugate symmetric.
2. The time-domain sequence looks like a cosine.
3. Symmetric frequency samples produce a real-valued oscillation in time; without this symmetry, the inverse transform would in general be complex-valued.