CENG 384 - Signals and Systems for Computer Engineers
Spring 2023
Homework 3

Adıgüzel, Gürhan İlhan
e2448025@ceng.metu.edu.tr

İçen, Anıl
e2448488@ceng.metu.edu.tr

May 14, 2023

---

1. Firstly, we know that $\int_{-\infty}^{\infty} x(t)dt$ is finite valued and periodic only if $a_0 = 0$.

   Then,

   $x(t) = \Sigma_{k=-\infty}^{\infty} a_k e^{jkw_0 t}$

   t $\longrightarrow s$

   $x(s) = \Sigma_{k=-\infty}^{\infty} a_k e^{jkw_0 s}$

   $\int_{-\infty}^{t} x(s)ds = \Sigma_{k=-\infty}^{\infty} a_k \int_{-\infty}^{\infty} e^{jkw_0 s} ds$

   $\int_{-\infty}^{t} x(s)ds = \Sigma_{k=-\infty}^{\infty} a_k \dfrac{e^{jkw_0 s}}{jkw_0}$

   Finally, we can see that the Fourier series coefficients are $\dfrac{1}{jkw_0} a_k$.

   $w_0 = \dfrac{2\pi}{T}$

   Therefore, it is $\dfrac{1}{jk\frac{2\pi}{T}} a_k$.

2. (a) $x(t)x(t)$

   From the Multiplication Property of CT Fourier Series:

   $x(t) \overset{FS}{\longleftrightarrow} a_k$ and $y(t) \overset{FS}{\longleftrightarrow} b_k$

   $x(t)y(t) \overset{FS}{\longleftrightarrow} c_k = a_k * b_k = \sum_{l=-\infty}^{\infty} a_l b_{k-l}$

   We have $x(t)x(t)$:

   $x(t)x(t) \overset{FS}{\longleftrightarrow} c_k = a_k * a_k = \sum_{l=-\infty}^{\infty} a_l a_{k-l}$

   (b) $\mathcal{E}v\{x(t)\}$

   From the Time-Reversal Property of CT Fourier Series:

   $x(t) \overset{FS}{\longleftrightarrow} a_k \quad \longrightarrow \quad x(-t) \overset{FS}{\longleftrightarrow} a_{-k}$

   In Fact, If x(t) is even $a_{-k} = a_k$

   $\frac{a_k + a_{-k}}{2}$

(c) $x(t+t_0)+x(t-t_0)$

From the Time Shifting Property of CT Fourier Series:

$x(t) \overset{FS}{\longleftrightarrow} a_k$

$x(t+t_0) \overset{FS}{\longleftrightarrow} e^{jkw_0t_0}a_k$

$x(t-t_0) \overset{FS}{\longleftrightarrow} e^{-jkw_0t_0}a_k$

From the Linearity Property of CT Fourier Series:

$x(t+t_0)+x(t-t_0) \overset{FS}{\longleftrightarrow} e^{jkw_0t_0}a_k + e^{-jkw_0t_0}a_k$

$x(t+t_0)+x(t-t_0) \overset{FS}{\longleftrightarrow} (e^{jkw_0t_0} + e^{-jkw_0t_0})a_k$

3. $x(t) = x_1(t) - x_1(t-2)$

$x_1(t) \overset{FS}{\longleftrightarrow} a_k$ and $x(t) \overset{FS}{\longleftrightarrow} b_k$

$$a_k = \frac{1}{T}\int_0^1 x_1(t)e^{-jkw_0t}dt = \frac{1}{4}\int_0^1 2e^{-jk\frac{2\pi}{4}t}dt = \frac{1}{2}\frac{e^{-j\frac{\pi}{2}kt}}{-j\frac{\pi}{2}k}\Bigg|_0^1 = \frac{1}{j\pi k}(1-e^{-j\frac{\pi}{2}k})$$

$$\int_T x_1(t-2)e^{-j\frac{2\pi}{T}kt}dt = \int_T x_1(t)e^{-j\frac{2\pi}{T}k(t+2)}dt = e^{-j\frac{2\pi}{T}k2}\int_T x_1(t)e^{-j\frac{2\pi}{T}kt}dt = e^{-j\frac{2\pi}{T}k2}a_k$$

From the Linearity Property of of CT Fourier Series:

$$b_k = a_k - e^{-j\frac{2\pi}{T}k2}a_k = (1-e^{-j\pi k})\frac{1}{j\pi k}(1-e^{-j\frac{\pi}{2}k})$$

The average value of x(t) is zero, so $b_0 = 0$.

4. (a) $x(t) = 1 + sin(w_0t) + 2cos(w_0t) + cos(2w_0t + \frac{\pi}{4})$

$1 + \frac{1}{2j}(e^{jw_0t} - e^{-jw_0}t) + \frac{2}{2}(e^{jw_0t} + e^{-jw_0t}) + \frac{1}{2}(e^{j(2w_0t+\frac{\pi}{4})} + e^{-j(-2w_0t+\frac{\pi}{4})})$

$1 + \frac{(-j)}{2}(e^{jw_0t} - e^{-jw_0}t) + (e^{jw_0t} + e^{-jw_0t}) + \frac{1}{2}(e^{j2w_0t} * e^{j\frac{\pi}{4}} + e^{-j2w_0t} * e^{-j\frac{\pi}{4}})$

$e^{j\frac{\pi}{4}} = (\frac{\sqrt{2}}{2} + \frac{\sqrt{2}}{2}j)$

$e^{-j\frac{\pi}{4}} = (\frac{\sqrt{2}}{2} - \frac{\sqrt{2}}{2}j)$

$1 + (1-\frac{j}{2})e^{jw_0t} + (1+\frac{j}{2})e^{-jw_0t} + \frac{1}{2}(e^{2jw_0t} * (\frac{\sqrt{2}}{2} + \frac{\sqrt{2}}{2}j) + e^{-2jw_0t} * (\frac{\sqrt{2}}{2} - \frac{\sqrt{2}}{2}j))$

$1 + (1-\frac{j}{2})e^{jw_0t} + (1+\frac{j}{2})e^{-jw_0t} + (\frac{\sqrt{2}}{4} + \frac{\sqrt{2}}{4}j)e^{2jw_0t} + (\frac{\sqrt{2}}{4} - \frac{\sqrt{2}}{4}j)e^{-2jw_0t}$

$a_0 = 1$, $a_1 = (1-\frac{j}{2})$, $a_{-1} = (1+\frac{j}{2})$, $a_2 = (\frac{\sqrt{2}}{4} + \frac{\sqrt{2}}{4}j)$, $a_{-2} = (\frac{\sqrt{2}}{4} - \frac{\sqrt{2}}{4}j)$

All other a_k's are zeros.

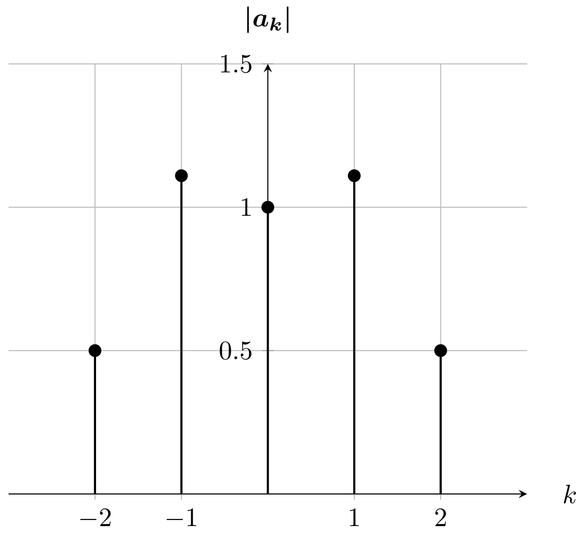

Figure 1: k vs $|a_k|$

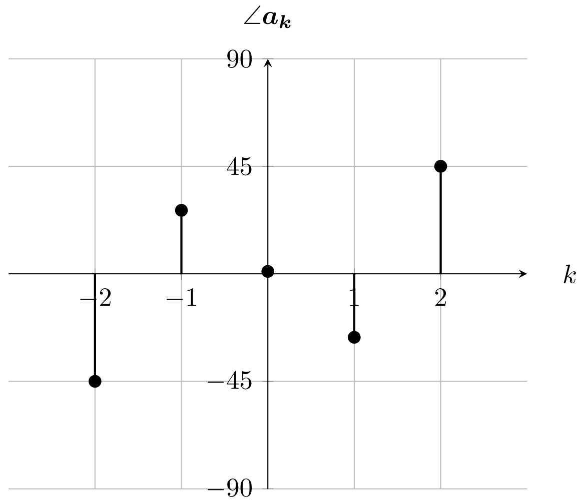

Figure 2: k vs $\angle a_k$

(b) Given the system equation:

$y'(t) + y(t) = x(t)$

we can rewrite it in the frequency domain using the Fourier transform. Let's say x(t) is X(w) and y(t) is Y(w). The derivative in the time domain equals to multiplication by jw in the frequency domain.

We have:

$(jw + 1)Y(w) = X(w)$

$Y(w) = \frac{1}{jw + 1}X(w)$

$H(w) = \frac{Y(w)}{X(w)} = \frac{1}{jw + 1}$

To find the eigenvalues of the system, denominator must be equal to 0.

Solving $jw + 1 = 0$, we find:

$w = j$

We found that the eigenvalue for the equation is j.

(c) When the input is $x(t) = e^{st}$, the output is $y(t) = H(s)e^{st}$

If $x(t) = \sum_k a_k e^{s_k t}$ then, $y(t) = \sum_k a_k H(s_k) e^{s_k t}$

$e^{s_k t}$: Eigenfunction

$H(s_k t)$: Eigenvalues

$$y(t) = \sum_{k=-\infty}^{\infty} a_k H(jkw_0) e^{jkw_0 t}$$

Therefore, using the equation $b_k = a_k H(jkw_0)$ for $H(jw)$ and $y(t)$, we get

$$y(t) = \sum_{k=-2}^{2} b_k e^{jkw_0 t}$$

$b_0 = 1$

$b_1 = (1 - \frac{j}{2})\frac{1}{1+jw}$ $\quad$ $b_{-1} = (1 + \frac{j}{2})\frac{1}{1-jw}$

$b_2 = (\frac{\sqrt{2}}{4} + \frac{\sqrt{2}}{4}j)\frac{1}{1+2jw}$ $\quad$ $b_{-2} = (\frac{\sqrt{2}}{4} - \frac{\sqrt{2}}{4}j)\frac{1}{1-2jw}$

(d)

$$y(t) = \sum_{k=-2}^{2} b_k e^{jkw_0 t}$$
$$= b_0 + b_1 e^{jw_0 t} + b_{-1} e^{-jw_0 t} + b_2 e^{2jw_0 t} + b_{-2} e^{-2jw_0 t}$$
$$= 1 + (1 - \frac{j}{2})\frac{1}{1+jw_0} e^{jw_0 t} + (1 + \frac{j}{2})\frac{1}{1-jw_0} e^{-jw_0 t} + (\frac{\sqrt{2}}{4} + \frac{\sqrt{2}}{4}j)\frac{1}{1+2jw_0} e^{2jw_0 t} + (\frac{\sqrt{2}}{4} - \frac{\sqrt{2}}{4}j)\frac{1}{1-2jw_0} e^{-2jw_0 t}$$

5. (a) $x[n] = sin(\frac{\pi}{2}n)$ $\quad$ $w_0 = \frac{\pi}{2}$

$x[n] = \frac{1}{2j}(e^{j\frac{\pi}{2}n} - e^{-j\frac{\pi}{2}n})$

$a_0 = 0,$ $\quad$ $a_1 = \frac{1}{2j} = \frac{-j}{2},$ $\quad$ $a_{-1} = \frac{-1}{2j} = \frac{j}{2}$

All other a_k 's $= 0$.

(b) $y[n] = 1 + cos(\frac{\pi}{2}n)$

$x[n] = 1 + \frac{1}{2}(e^{j\frac{\pi}{2}n} + e^{-j\frac{\pi}{2}n}),$ $\quad$ $w_0 = \frac{\pi}{2}$

$b_0 = 1,$ $\quad$ $b_1 = b_{-1} = \frac{1}{2}$

All other a_k's $= 0$.

(c) Multiplication Property is as follows

$$x(t) \xleftrightarrow{FS} a_k \text{ and } y(t) \xleftrightarrow{FS} b_k$$

$$a_k * b_k \xleftrightarrow{FS} \sum_{\forall l} a_l b_{k-l}$$

$$d_k = \sum_{l=0}^{3} a_l b_{k-l}, \text{ since } N = 4$$

$$= a_1 b_{k-1} + a_{-1} b_{k-3}$$

$$d_0 = a_1 b_{-1} + a_{-1} b_{-3} = \frac{1}{4j} - \frac{1}{4j} = 0$$

$$d_1 = \frac{1}{2j} = \frac{-j}{2}, \qquad d_{-1} = \frac{-1}{2j} = \frac{j}{2},$$

$$d_2 = d_{-2} = \frac{1}{4j} - \frac{1}{4j}$$

All other d_k's $= 0$.

(d)

$$z[n] = x[n] * y[n] = \sum_{k=-\infty}^{\infty} x[n-k] \cdot y[k]$$

$$z[n] = sin(\frac{\pi}{2}n) + sin(\frac{\pi}{2}n)cos(\frac{\pi}{2}n)$$

$$z[n] = sin(\frac{\pi}{2}n) + \frac{1}{2}sin(\pi n)$$

$sin(\pi n)$ is always 0 and using Euler's relation, we have

$$z[n] = \frac{1}{2j}(e^{j\frac{\pi}{2}n} - e^{-j\frac{\pi}{2}n})$$

$$d_1 = \frac{1}{2j} = \frac{-j}{2}, d_{-1} = \frac{-1}{2j} = \frac{j}{2}$$

All other d_k's $= 0$.
Comparing d_k with the d_k from the part (c), we see that both are the same.

6. (a)

$$x[n] = \sum_{n=0}^{3} a_k e^{jk(2\pi/4)n}$$

$$a_k = \frac{1}{4}\sum_{n=0}^{3} x[n] e^{-jk\frac{\pi}{2}n}$$

The preceding set of linear equations can be reduced to

$$a_0 = \frac{1}{4}\sum_{n=0}^{3} x[n] e^{0} = \frac{1}{4}(0+1+2+1) = 1$$

$$a_1 = \frac{1}{4}\sum_{n=0}^{3} x[n] e^{-j\frac{\pi}{2}n} = \frac{-1}{2}$$

$$a_2 = \frac{1}{4}\sum_{n=0}^{3} x[n] e^{-j\pi n} = 0$$

$$a_3 = \frac{1}{4}\sum_{n=0}^{3} x[n] e^{-j\frac{3\pi}{2}n} = \frac{-1}{2}$$

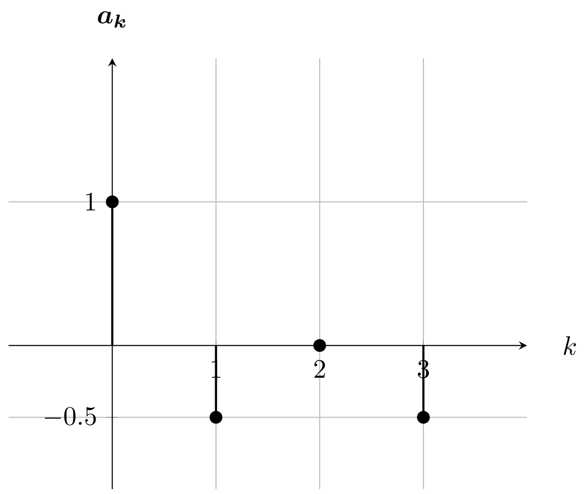

Figure 3: k vs a_k

The magnitude of the spectral coefficients:

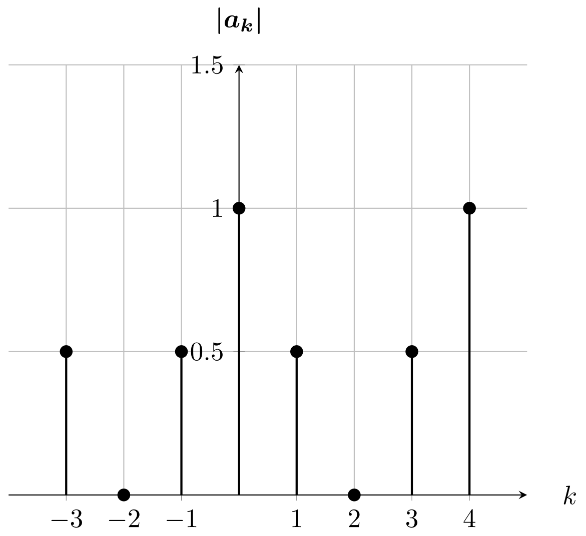

Figure 4: k vs $|a_k|$

Phase of the spectral coefficients:

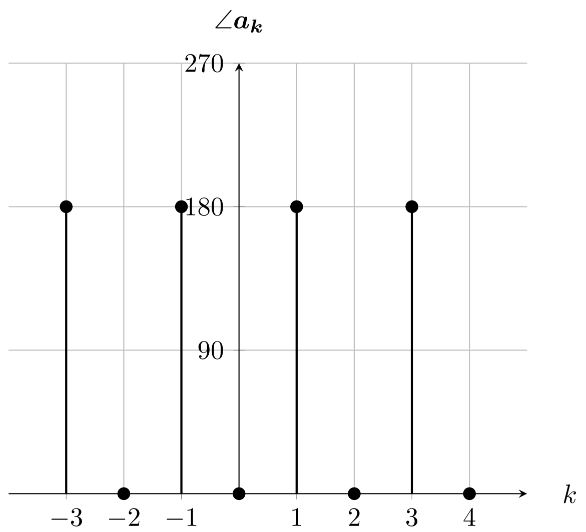

Figure 5: k vs $\angle a_k$

(b) i)
$y[n] = x[n] - \Sigma_{k=-\infty}^{\infty}\delta$ [n-3 + Nk] $k \in Z, N = 4$

ii)
$a_0 = \frac{1}{4}\Sigma_{n=0}^{3}y[n]e^0 = \frac{1}{4}[0+1+2+0] = \frac{3}{4}$
$a_1 = \frac{1}{4}\Sigma_{n=0}^{3}y[n]e^{-j\frac{\pi}{2}n} = \frac{-j}{4} - \frac{1}{2}$
$a_2 = \frac{1}{4}\Sigma_{n=0}^{3}y[n]e^{-j\pi n} = \frac{1}{4}$
$a_3 = \frac{1}{4}\Sigma_{n=0}^{3}y[n]e^{-j\frac{3\pi}{2}n} = \frac{j}{4} - \frac{1}{2}$

$$y[n] = \sum_{n=0}^{3} a_k e^{jk(2\pi/4)n}$$

Period is $N = 4$. So the signal $y[n]$ can be expressed as above.

$$y[0] = a_0 + a_1 + a_2 + a_3 = 0$$

$$y[1] = a_0 + a_1 e^{j(\pi/2)} + a_2 e^{j\pi} + a_3 e^{j(3\pi/2)} = 1$$
$$y[2] = a_0 + a_1 e^{j\pi} + a_2 e^{2j\pi} + a_3 e^{j3\pi} = 2$$
$$y[3] = a_0 + a_1 e^{j(3\pi/2)} + a_2 e^{j3\pi} + a_3 e^{j(9\pi/2)} = 0$$

The preceding set of linear equations can be reduced to

$$a_0 + a_1 + a_2 + a_3 = 0$$

$$a_0 + ja_1 - a_2 - ja_3 = 1$$
$$a_0 - a_1 + a_2 - a_3 = 2$$
$$a_0 - ja_1 - a_2 + ja_3 = 0$$

Solving the equations we get

$$a_0 = \frac{3}{4}, a_2 = \frac{1}{4}, a_1 = \frac{1-2j}{2j} a_3 = \frac{1+2j}{2j} \text{ All other } a_k\text{'s } = 0.$$

The magnitude of spectral coefficients:

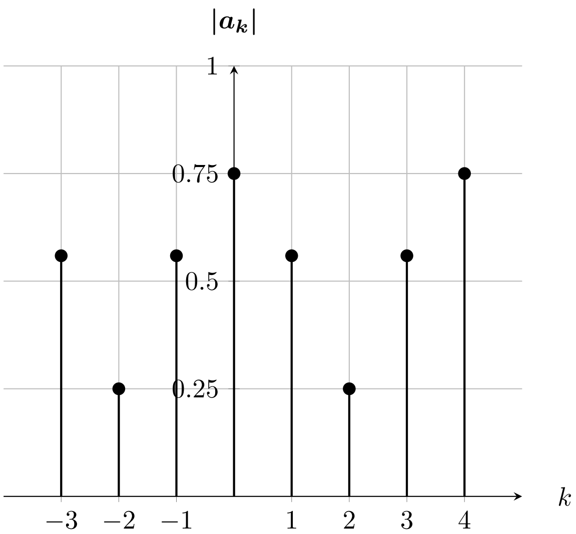

Figure 6: k vs $|a_k|$

Phase spectral coefficients:

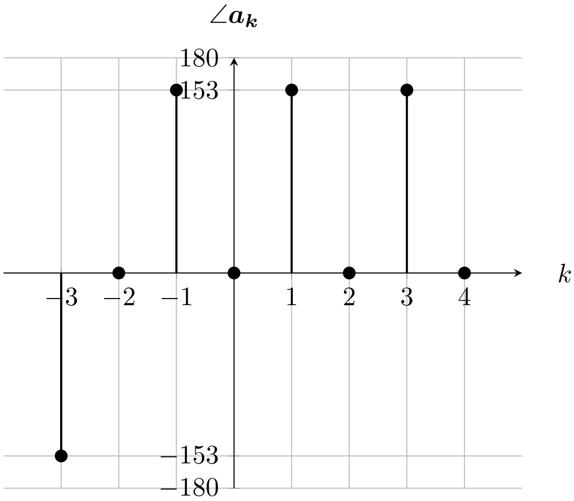

Figure 7: k vs $\angle a_k$

7. (a) To be $|\omega| < 80$, it must be $\omega < 80$ and $-80 < \omega$

$\omega = k\omega_0$

$\omega = k\frac{2\pi}{T}$

$\omega = k\frac{2\pi}{\frac{\pi}{K}}$

$w = k * 2K$

Therefore, $\frac{-40}{K} < k < \frac{40}{K}$

a_k when k is between the interval above, it may be any value because the function is a low pass filter.

When it is outside of the interval, a_k is equal to 0.

(b) $y(t) = h(t) * x(t)$

h(t) can be 1 or 0. To get the inequality y(t) $\neq$ x(t), we need to find h(t) = 0 and choose x(t) $\neq$ 0. That happens only outside the

$\frac{-40}{K} < k < \frac{40}{K}$ interval.

We cannot find the inequality in the interval because h(t) is 1.

8.
```
import numpy as np
import matplotlib.pyplot as plt

def fourierCoefficients(signal, period, n):
    # Compute DC component
    dc = np.mean(signal)
    # Compute Omega0
    omega_0 =(2 * np.pi)/ period
    t = np.linspace(-0.5, 0.5, 1000, endpoint=False)

    # Compute Fourier series coefficients for cosine components
    cos_coeffs = np.zeros(n)
    for i in range(1, n+1):
        cos_coeffs[i-1] = 2*np.sum(signal*np.cos(i*omega_0*t)) / len(signal)
    # Compute Fourier series coefficients for sine components
    sin_coeffs = np.zeros(n)
    for i in range(1, n+1):
        sin_coeffs[i-1] = 2*np.sum(signal*np.sin(i*omega_0*t)) / len(signal)

    return dc, cos_coeffs, sin_coeffs

def fourierApproximation(coeffs, period, t):
    dc, cos_coeffs, sin_coeffs = coeffs
    omega_0 =(2 * np.pi)/ period
    signal = dc + np.sum([a*np.cos(k*omega_0*t) + b*np.sin(k*omega_0*t)
                          for k, (a, b) in enumerate(zip(cos_coeffs, sin_coeffs), start=1)],
    return signal

def square_wave(time):
    if (time < 0): return -1
    else: return 1

def sawtooth_wave(time):
    if (time < 0): return 1 + 2*time
    else: return -1 + 2*time

n_points = 1000
time = np.linspace(-0.5, 0.5, n_points, endpoint=False)
signal = np.zeros(n_points)
# Generate the square wave function
signal = np.array([square_wave(i) for i in time])
# Generate the sawtooth signal
#signal = np.array([sawtooth_wave(i) for i in time])

# Compute the Fourier Series coefficients of the square wave function
period = 1
n_coeffs = [1, 5, 10, 50, 100]
coeffs = [fourierCoefficients(signal, period, n) for n in n_coeffs]

# Generate the approximated functions using the Fourier Series coefficients
signals = [fourierApproximation(coeffs[i], period, time) for i in range(len(n_coeffs))]

# Plot the original and approximated functions
plt.figure(figsize=(8, 6))
plt.plot(time, signal, label='Original')
colors = ['r', 'g', 'b', 'm', 'y']
for i in range(len(n_coeffs)):
    plt.plot(time, signals[i], label=f'n={n_coeffs[i]}', color=colors[i])

plt.legend()
plt.title('Approximation of Square Wave Function using Fourier Series')
plt.xlabel('Time')
plt.ylabel('Amplitude')
plt.show()
```

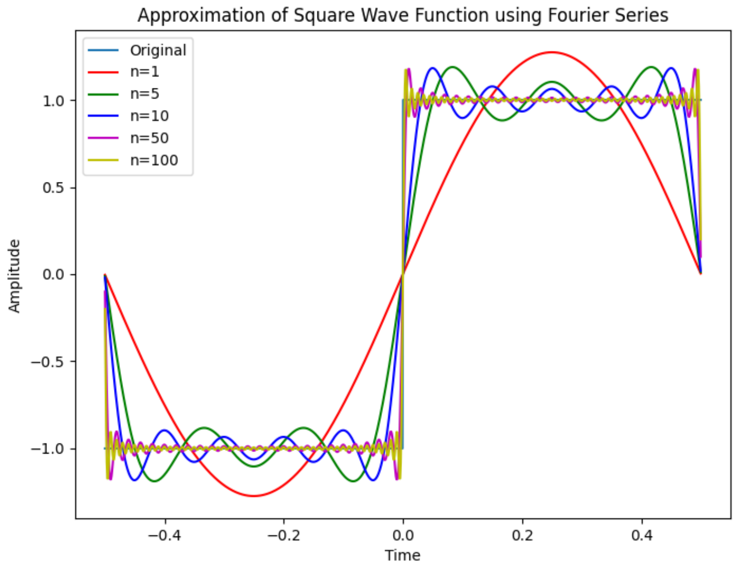

Figure 8: Square Wave Signal Approximation

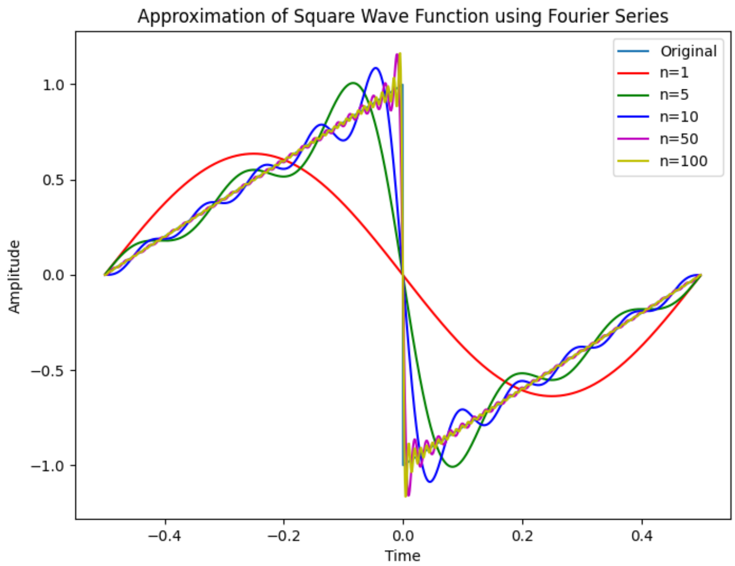

Figure 9: Sawtooth Wave Signal Approximation

Increasing the number of Fourier Series coefficients (n) has the effect of improving the accuracy of the Fourier Series approximation of a given signal.

The Fourier Series is a representation of a periodic function as a sum of sinusoidal functions with different amplitudes and frequencies. The more terms (coefficients) we include in the Fourier Series, the more closely the approximation will approximate the original signal.

In particular, as the number of coefficients increases, the Fourier Series approximation becomes more accurate at representing sharp transitions and irregularities in the signal. This is because higher-order Fourier coefficients correspond to higher-frequency sinusoidal functions, which can capture more complex features of the signal.