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

Bilgin, Mert Can
e2655181@ceng.metu.edu.tr

LastName2, FirstName2
exxxxxxx@ceng.metu.edu.tr

May 14, 2023

---

1. From synthesis equation:

$$x(t) = \sum_{k=-\infty}^{\infty} a_k e^{jkw_o t}$$

If we take the integral of both sides, we get:

$$\int_{-\infty}^{t} x(t)dt = \sum_{k=-\infty}^{\infty} \int_{-\infty}^{t} a_k e^{jkw_o t} dt$$

$$\int_{-\infty}^{t} x(t)dt = \sum_{k=-\infty}^{\infty} a_k \int_{-\infty}^{t} e^{jkw_o t} dt$$

$$\int_{-\infty}^{t} x(t)dt = \sum_{k=-\infty}^{\infty} a_k \frac{e^{jkw_o t}}{jkw_o}$$

When we compare

$$x(t) \ and \ \int_{-\infty}^{t} x(t)dt$$

we see that Fourier series coefficients are

$$\frac{1}{jkw_o} a_k$$

Hence, we proved that Fourier series coefficients of

$$\int_{-\infty}^{t} x(s)dt = \frac{1}{jk\frac{2\pi}{T}} a_k$$

2. (a) From the multiplication property:

$$x(t)x(t) <-> a_k * a_k = \sum_{l=-\infty}^{\infty} a_l a_{k-l}$$

(b) We can write:

$$x(t) = a_0 + \sum_{k=1}^{\infty} (b_k cos(kw_o t) - c_k sin(kw_o t))$$

Because we use even part of x(t), cks are all 0

$$Ev(x(t)) = a_0 + \sum_{k=1}^{\infty} (b_k cos(kw_o t)$$

$$a_k = \frac{1}{2} b_k, b_k = \frac{2}{T} \int_T x(t) cos(kw_o t) dt$$

(c) From the time shifting property:

$$e^{jkw_o t} a_k + e^{-jkw_o t} a_k = a_k (e^{jkw_o t} + e^{jkw_o t})$$

3. If we take derivative of x(t):

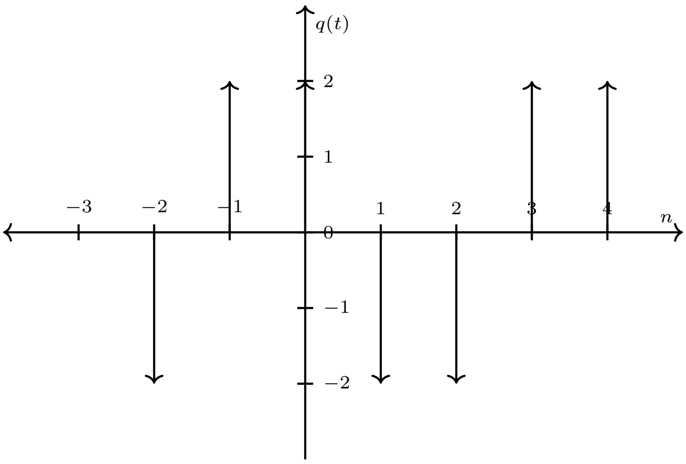

Figure 1: q(t)

We know:

$$\sum_k \delta(t-kT) <-> a_k = \frac{1}{T}$$

$$q(t) = 2x(t+2) - 2x(t-2)$$

$$q(t) <-> b_k = 2e^{jkw_o 2}a_k - 2e^{-jkw_o 2}$$

From Euler's formula, we get:

$$\frac{4j}{T} sin(kw_o 2)$$

By derivative property:

$$q(t) = \frac{dx(t)}{dt} <-> b_k = jkw_o a_k$$

$$=> a_k = \frac{2}{k\pi}, k \neq 0,\ where\ w_o = \frac{2\pi}{4} = \frac{\pi}{2}$$

$$a_o = \frac{the\ area\ of\ a\ period}{period} = \frac{4}{4} = 1$$

4. (a) We can split x(t) as:

$$x(t) = x_1(t) + x_2(t) + x_3(t) + x_4(t)$$

$$x_1(t) = e^{j0w_o t} = 1$$

$$x_2(t) = sin(w_o t) = \frac{e^{jwt} - e^{-jwt}}{2j}$$

$$x_3(t) = 2cos(w_o t) = e^{jwt} + e^{-jwt}$$

$$x_4(t) = cos(2w_o t + \frac{\pi}{4}) = cos(2wt)cos(\frac{\pi}{4}) - sin(2wt)sin(\frac{\pi}{4}) = \frac{\sqrt{2}}{2}(\frac{e^{2jwt} + e^{-2jwt}}{2} - \frac{e^{2jwt} - e^{-2jwt}}{2j})$$

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

To sketch, we need magnitudes and angles:

$$|a_0| = 1, |a_1| = \frac{\sqrt{5}}{2}, |a_2| = \frac{1}{2}, |a_{-1}| = \frac{\sqrt{5}}{2}, |a_{-2}| = \frac{1}{2}$$

$$\angle a_0 = 0, \angle a_1 = arctan(\frac{-1}{2}) = -0.464rad, \angle a_2 = \frac{\pi}{4}, \angle a_{-1} = arctan(\frac{1}{2}) = 0.464rad, \angle a_{-2} = arctan(-1) = \frac{-\pi}{4}$$

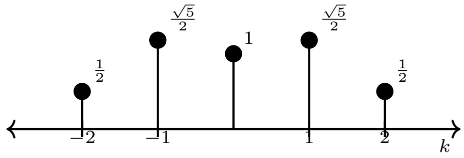

Figure 2: magnitude of ak

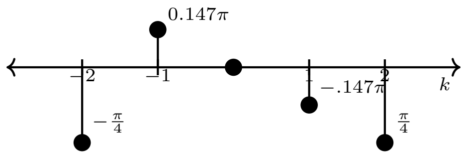

Figure 3: phase of ak

(b)

$$x(t) = e^{jwt},\ then\ y(t) = H(jw)e^{jwt}$$
$$jwH(jw)e^{jwt} + H(jw)e^{jwt} = e^{jwt}$$
$$H(jw)e^{jwt}(1 + jw) = e^{jwt}$$
$$H(jw) = \frac{1}{1 + jw}$$

(c)

$$b_k = a_k H(jkw_o) = a_k \frac{1}{1 + jkw_o}$$
$$b_{-2} = a_{-2}\frac{1}{1-2j} = (\frac{\sqrt{2}}{4} + \frac{\sqrt{2}}{4j})\frac{1}{1-2j} => |b_{-2}| = \frac{1}{2\sqrt{5}}$$
$$b_{-1} = a_{-1}\frac{1}{1-j} = (1 - \frac{1}{2j})\frac{1}{1-j} => |b_{-1}| = \frac{\sqrt{5}}{2\sqrt{2}}$$
$$b_0 = a_0 => |b_0| = 1$$
$$b_1 = a_1\frac{1}{1+j} = (1 + \frac{1}{2j})\frac{1}{1+j} => |b_1| = \frac{\sqrt{5}}{2\sqrt{2}}$$
$$b_2 = a_2\frac{1}{1-2j} = (\frac{\sqrt{2}}{4} - \frac{\sqrt{2}}{4j})\frac{1}{1+2j} => |b_2| = \frac{1}{2\sqrt{5}}$$

Angles:

$$\angle b_{-2} = \frac{\sqrt{5}}{10}, \angle b_{-1} = \frac{\sqrt{10}}{4}, \angle b_0 = 0, \angle b_1 = \frac{\sqrt{10}}{4}, \angle b_2 = \frac{\sqrt{5}}{10}$$

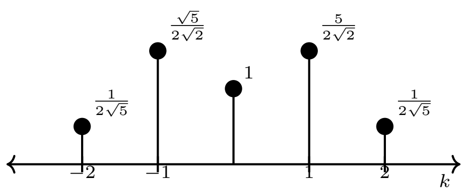

Figure 4: magnitude of bk

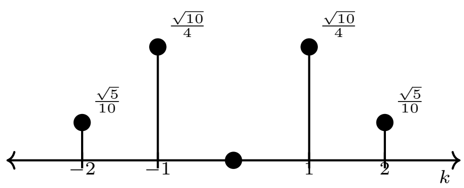

Figure 5: phase of bk

(d)

$$y(t) = \sum_{k=-\infty}^{\infty} b_k e^{jkw_o t}$$

$$y(t) = b_{-2}e^{-2jw_o t} + b_{-1}e^{-jw_o t} + b_0 + b_1 e^{jw_o t} + b_2 e^{2jw_o t}$$

5. (a)

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

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

$$|a_1| = |a_{-1}| = \frac{1}{2}$$

$$\angle a_1 = -\frac{\pi}{2}, \angle a_{-1} = \frac{\pi}{2}$$

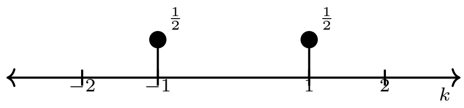

Figure 6: magnitude of ak

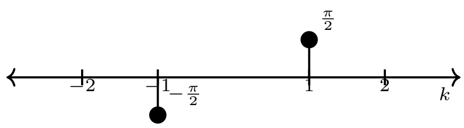

Figure 7: phase of ak

(b)

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

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

$$|b_1| = |b_{-1}| = \frac{1}{2}$$

$$\angle b_1 = \angle b_{-1} = 0$$

(c)

$$x[n]y[n] <-> c_k = \frac{1}{N}\sum_{l=-1}^{N-1} a_l b_{k-l}, \ where\ N = 4$$

$$c_0 = \frac{1}{2}(a_{-1}b_1 + a_0 b_1 + a_1 b_{-1}) = \frac{1}{2}(-\frac{1}{4j} + \frac{1}{4j}) = \frac{1}{2}$$

$$c_1 = \frac{1}{2}(a_{-1}b_2 + a_0 b_1 + a_1 b_0) = \frac{1}{2}(0 + 0 + \frac{1}{2j}) = \frac{1}{4j}$$

$$c_{-1} = \frac{1}{2}(a_{-1}b_0 + a_0 b_{-1} + a_1 b_2) = \frac{1}{2}\frac{1}{2j} = \frac{1}{4j}$$

(d)

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

$$c_k = \frac{1}{N}\sum_{n=<N>} x[n]y[n]e^{-jk\frac{\pi}{2}n}$$

$$c_0 = \frac{1}{4}\sum_{n=-2}^{2} x[n]y[n]$$

$$c_1 = \frac{1}{4}\sum_{n=-2}^{2} x[n]y[n]e^{-j\frac{\pi}{2}n}$$

$$c_{-1} = \frac{1}{4}\sum_{n=-2}^{2} x[n]y[n]e^{j\frac{\pi}{2}n}$$

6. (a)

$$a_k = \frac{1}{N}\sum_{n=<N>} x[n]e^{-jkw_0 n},\ where\ N = 4$$

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

$$= \frac{1}{4}(x[0] + x[1]e^{-jk\frac{\pi}{2}} + x[2]e^{-2jk\frac{\pi}{2}} + x[3]e^{-3jk\frac{\pi}{2}})$$

$$= 0 + \frac{1}{4}(cos(k\frac{\pi}{2}) - jsin(k\frac{\pi}{2})) + \frac{1}{2}(cos(k\pi) - jsin(k\pi)) + \frac{1}{4}(cos(3k\frac{\pi}{2}) - jsin(3k\frac{\pi}{2}))$$

$$a_0 = \frac{1}{4} + \frac{1}{2} + \frac{1}{4} = 1$$

$$a_1 = -\frac{j}{4} - \frac{1}{2} + \frac{j}{4} = -\frac{1}{2}$$

$$a_2 = -\frac{1}{4} + \frac{1}{2} - \frac{1}{4} = 0$$

$$a_3 = \frac{j}{4} - \frac{1}{2} - \frac{j}{4} = -\frac{1}{2}$$

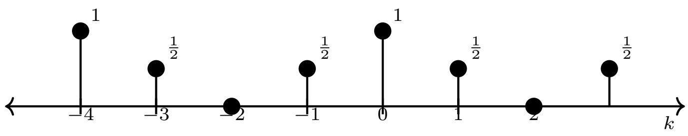

Figure 8: magnitude of ak

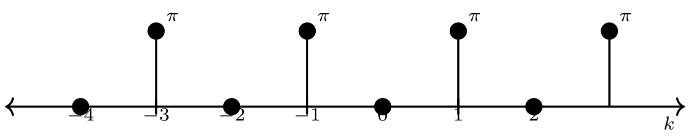

Figure 9: phase of ak

(b) i) We can express y[n] as

$$y[n] = x[n] - \sum_{k=-\infty}^{\infty} \delta(n + 1 - 4k)$$

ii)

$$b_k = \frac{1}{N}\sum_{n=<N>} x[n]e^{-jkw_0 n},\ where\ N = 4$$

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

$$= \frac{1}{4}(x[0] + x[1]e^{-jk\frac{\pi}{2}} + x[2]e^{-2jk\frac{\pi}{2}} + x[3])$$

$$= 0 + \frac{1}{4}(cos(k\frac{\pi}{2}) - jsin(k\frac{\pi}{2})) + \frac{1}{2}(cos(k\pi) - jsin(k\pi)) + 0$$

$$b_0 = \frac{1}{4} + \frac{1}{2} = \frac{3}{4}$$
$$b_1 = -\frac{j}{4} - \frac{1}{2} = -\frac{1}{4}(j+2)$$
$$b_2 = -\frac{1}{4} + \frac{1}{2} = \frac{1}{4}$$
$$b_3 = \frac{j}{4} - \frac{1}{2} = \frac{1}{4}(j-2)$$

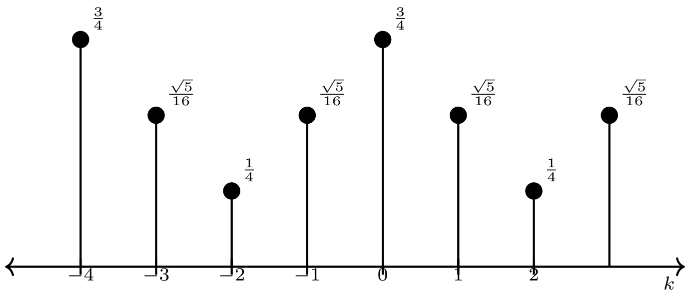

Figure 10: magnitude of bk

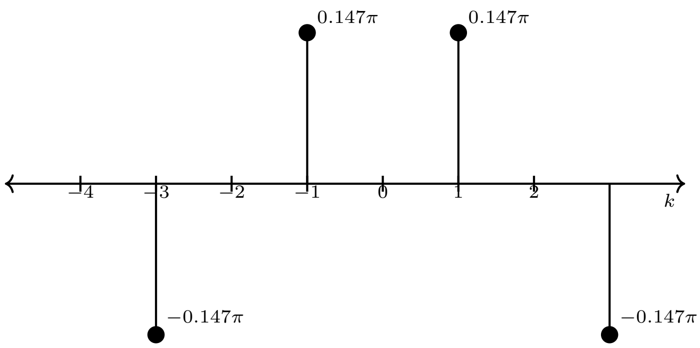

Figure 11: phase of bk

7. (a) We know:

$$if\ x(t) = e^{jwt},\ then\ y(t) = H(jw)e^{jwt}$$

It is an ideal LPF, and if y(t) = x(t),

$$a_k = 0\ for\ |w| > 80$$
$$w_o = \frac{2\pi}{T} = 2K$$

Hence,

$$a_k = 0,\ for\ |k| > \frac{80}{2K}$$

(b)

$$If\ y(t) \neq x(t),\ then\ a_k\ must\ have\ non-zero\ value\ for\ |w| > 80$$
$$a_k \neq 0\ for\ |k| > \frac{80}{2K}$$

8. a)

```
import numpy as np

def fourier_series_coeff(signal, period, num_coeffs):
    # Compute the DC component
    dc_coeff = np.mean(signal)

    # Compute the coefficients for the cosine terms
    cos_coeffs = np.zeros(num_coeffs)
    for n in range(1, num_coeffs+1):
        cos_coeffs[n-1] = 2*np.mean(signal*np.cos(2*np.pi*n*signal/period))

    # Compute the coefficients for the sine terms
    sin_coeffs = np.zeros(num_coeffs)
    for n in range(1, num_coeffs+1):
        sin_coeffs[n-1] = 2*np.mean(signal*np.sin(2*np.pi*n*signal/period))

    return dc_coeff, cos_coeffs, sin_coeffs
```

b)

```
    def generate_approximate_function(dc_coeff, cos_coeffs, sin_coeffs, period, num_points)

    # Generate the x values
    x = np.linspace(0, period, num_points)

    # Compute the y values of the approximate function
    y = np.zeros(num_points)
    for n in range(1, len(cos_coeffs)+1):
        y += cos_coeffs[n-1]*np.cos(2*np.pi*n*x/period) + sin_coeffs[n-1]*np.sin(2*np.pi*n*
    y += dc_coeff

    return x, y
```

c)

```
import matplotlib.pyplot as plt

# Define the square wave function
def square_wave(x):
    if x < 0:
        return -1
    else:
        return 1

# Generate the signal over one cycle
period = 1
num_points = 1000
x = np.linspace(-0.5, 0.5, num_points)
signal = np.array([square_wave(xi) for xi in x])

# Compute the Fourier Series coefficients
num_coeffs = 100
dc_coeff, cos_coeffs, sin_coeffs = fourier_series_coeff(signal, period, num_coeffs)

# Generate the approximate function
x_approx, y_approx = generate_approximate_function(dc_coeff, cos_coeffs[:100], sin_coeffs[:

# Plot the original and approximate functions
plt.plot(x, signal, label='Original Function')
plt.plot(x_approx, y_approx, label='Approximate Function (n=100)')
plt.legend()
plt.show()

# Plot the original and approximate functions for different values of n
n_values = [1, 5, 10, 50, 100]
```

```
for n in n_values:
    x_approx, y_approx = generate_approximate_function(dc_coeff, cos_coeffs[:n], sin_coeffs
    plt.plot(x, signal, label='Original Function')
    plt.plot(x_approx, y_approx, label=f'Approximate Function (n={n})')
    plt.legend()
    plt.show()
```

d)

```
import scipy.signal as signal

# Generate the signal over one cycle
period = 1
num_points = 1000
x = np.linspace(-0.5, 0.5, num_points)
signal = signal.sawtooth(2 * np.pi * x / period, width=0.5)

# Compute the Fourier Series coefficients
num_coeffs = 100
dc_coeff, cos_coeffs, sin_coeffs = fourier_series_coeff(signal, period, num_coeffs)

# Generate the approximate function
x_approx, y_approx = generate_approximate_function(dc_coeff, cos_coeffs[:100], sin_coeffs[:

# Plot the original and approximate functions
plt.plot(x, signal, label='Original Function')
plt.plot(x_approx, y_approx, label='Approximate Function (n=100)')
plt.legend()
plt.show()

# Plot the original and approximate functions for different values of n
n_values = [1, 5, 10, 50, 100]
for n in n_values:
    x_approx, y_approx = generate_approximate_function(dc_coeff, cos_coeffs[:n], sin_coeffs
    plt.plot(x, signal, label='Original Function')
    plt.plot(x_approx, y_approx, label=f'Approximate Function (n={n})')
    plt.legend()
    plt.show()
```

What is the effect of increasing n?
It leads to a more accurate approximation of the original function

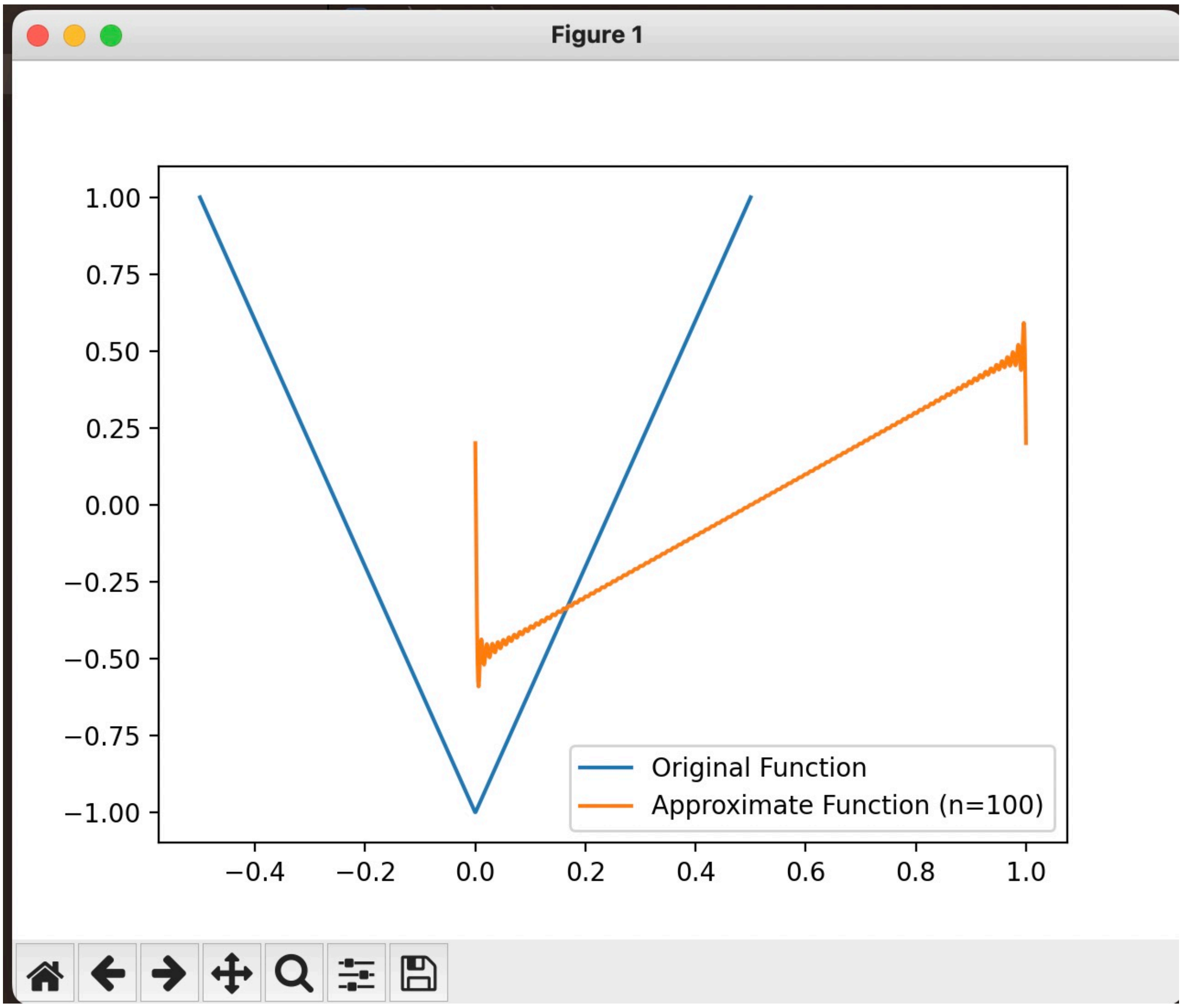
Figure 1
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−0.4
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
Original Function
Approximate Function (n=100)

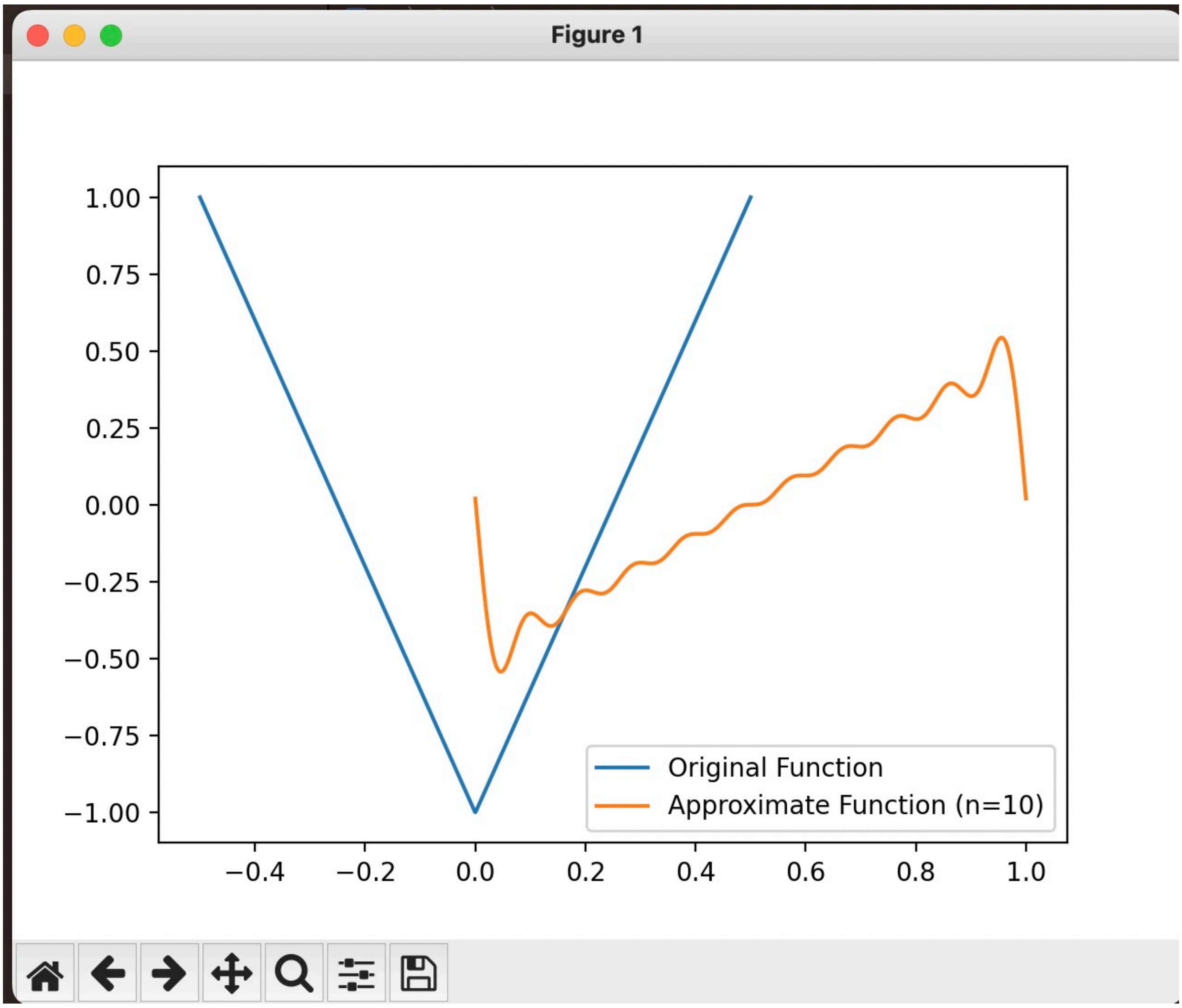
Figure 1
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−0.4
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
Original Function
Approximate Function (n=10)

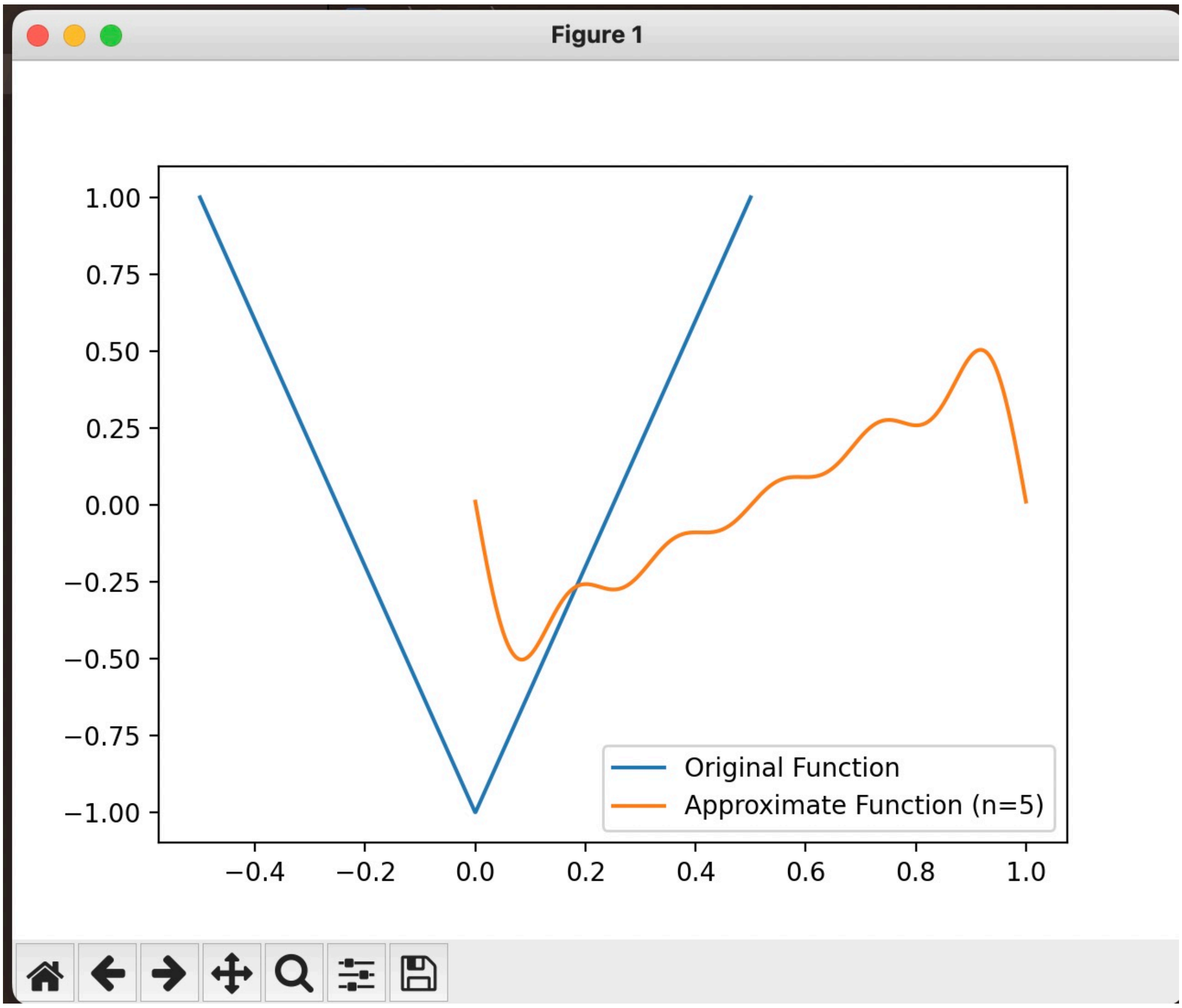
Figure 1
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−0.4
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
Original Function
Approximate Function (n=5)

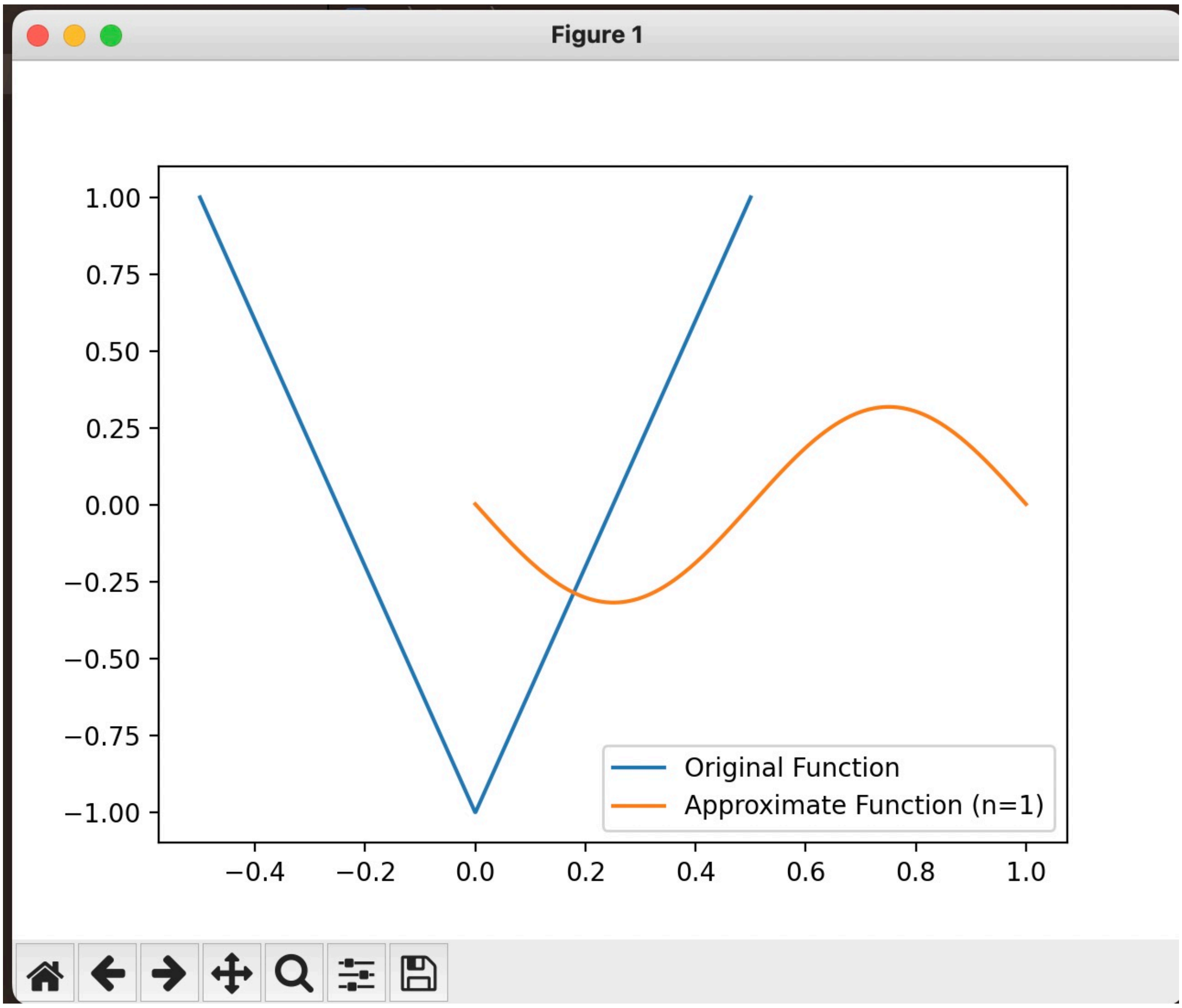
Figure 1
Original Function
Approximate Function (n=1)
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−0.4
−0.2
0.0
0.2
0.4
0.6
0.8
1.0