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

Qaryouti, Hashem
e2652816@ceng.metu.edu.tr

Muhrabi, Sogol
e2547321@ceng.metu.edu.tr

May 17, 2023

1.

$$x(t) = \sum_{k=-\infty}^{\infty} a_k e^{jkwt} X(t) = \int_{k=-\infty}^{t} x(s)\, ds = \int_{k=-\infty}^{t} \sum_{-\infty}^{\infty} a_k e^{jkws}\, ds = \sum_{k=-\infty}^{\infty} \int_{-\infty}^{t} a_k e^{jkws}\, ds = \sum_{k=-\infty}^{\infty} \frac{a_k}{jwk} e^{jwks}$$

Therefore,the coefficient of X(t) is,

$$\frac{a_k}{jwk} = \frac{1}{jk\frac{2\pi}{T}} a_k$$

. (a) Using the multiplication property of fourier series for CT signals, the coefficient of x(t)*x(t) is a$_k * a_k$ which is convolution operation $\sum_{\forall l} a_l a_{k-l}$

(b)
X$_e(t) = x(t) + x(-t)/2$
$= 0.5 \int_{k=-\infty}^{\infty} a_k e^{jkwt} + a_{-k} e^{jkwt}\, dt$

$$= 0.5 \int_{k=-\infty}^{\infty} (a_k + a_{-k}) e^{jkwt}\, dt$$

So the coefficient of x$_e(t) is (a_k + a_{-k})/2$

(c)
x(t) = a$_k$
$x(t - t_0) = a_k e^{-jkw_0 t_0}$
$x(t + t_0) = a_k e^{+jkw_0 t_0}$
$x(t - t_0) + x(t + t_0) = a_k[e^{+jkw_0 t_0} + e^{-jkw_0 t_0}]$

$x(t - t_0) + x(t + t_0) = 2a_k[\cos kw_o t_o]$

3.
x(t) = $\sum_{-\infty}^{\infty} c_k e^{jkwt}$
$c_k = 1/T \int_0^T x(t) e^{-jkwt}\, dt$

$T = 4s, w = \pi/2 (rad/s)$

c$_k = 0.25 \int_1^5 x(t) e^{-jkwt}\, dt$
$= 0.25[\int_1^2 x(t)e^{-jkwt} + \int_2^3 x(t)e^{-jkwt} + \int_3^4 x(t)e^{-jkwt} + \int_4^5 x(t)e^{-jkwt}]$
$Note(\int_1^2 x(t)e^{-jkwt} = \int_3^4 x(t)e^{-jkwt}) = 0$
$= 0.25[\int_2^3 x(t)e^{-jkwt} + \int_4^5 x(t)e^{-jkwt}]$
$= 0.25[(2/jkw)e^{-jkwt} + (-2/jkw)e^{-jkwt}]$

$c_k = 1/2jkw[e^{-jk3\pi/2} - e^{-jk\pi} - e^{-jk5\pi/2} + e^{-jk2\pi}]$

e$^{-jk3\pi/2} = \cos(-3\pi k/2) + j\sin(-3\pi k/2)$

$\mathrm{e}^{-j\pi k} = \cos(-\pi k) + j\sin(-\pi k)$
$= 1$ when k is even
$=$ -1 when k is odd

$e^{-jk5\pi 2} = \cos(-5\pi k/2) + j\sin(-5\pi k/2)$

$\mathrm{e}^{-jk2\pi} = \cos(-2\pi k) + j\sin(-2\pi k)$
$= 1$ when k is even
$= 1$ when k is odd

$c_k = \frac{2+e^{\frac{-3\pi jk}{2}}+e^{\frac{-jk5\pi}{2}}}{jk\pi}$ k is odd
$c_k = 1/jk\pi[1-1-1+1] = 0$ when k is even
$c_k = 1/jk\pi[j-(-1)-(-j)+1] = (2\pm 2j)/jk\pi$ when k is odd

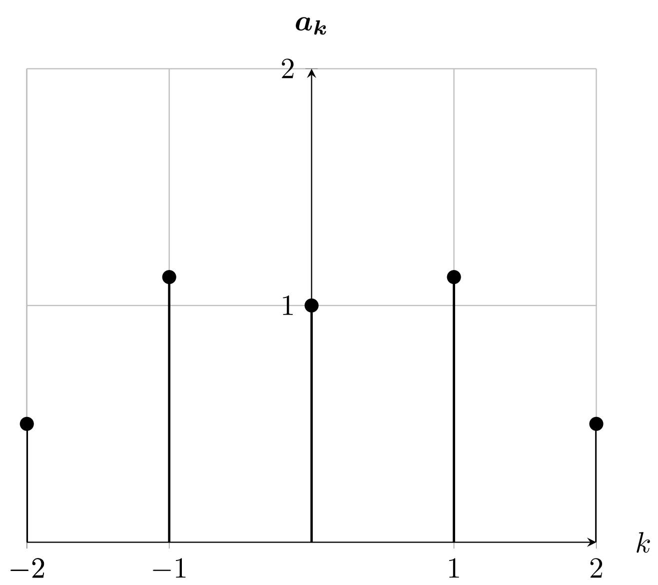
$x(t) = \sum_{-\infty}^{\infty} c_k e^{jkwt}$

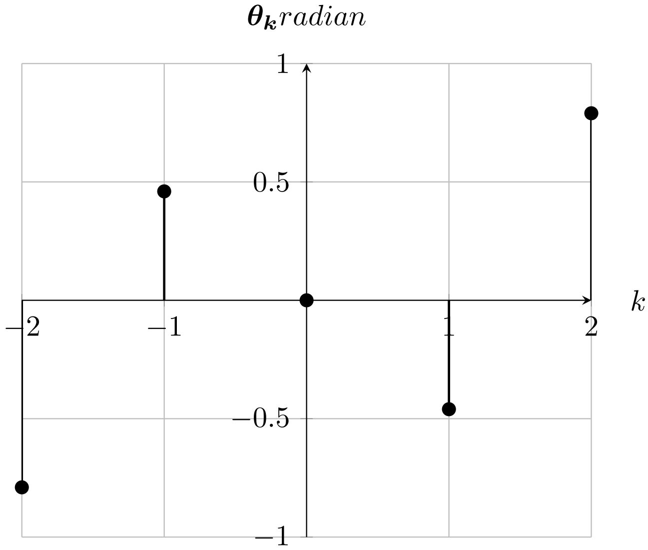
$x(t) = \sum_{-\infty}^{\infty} \frac{2+e^{\frac{-3\pi jk}{2}}+e^{\frac{-jk5\pi}{2}}}{jk\pi} e^{jkwt}$ k is odd

$x(t) = \sum_{-\infty}^{\infty}(2\pm 2j)/jk\pi * e^{jk\pi t/2}$ Complex Exponential Form / k is odd

$|c_k| = 2\sqrt{2}/k\pi$
$\theta_k = tan^{-}(\pm 1) = \pm 45$ k is odd
$c_o = 0$
$x(t) = c_0 + \sum_1^{\infty} 2|c_k|\cos(kwt+\theta_k)$
$x(t) = \sum_1^{\infty} 4\sqrt{2}/k\pi * \cos(\pi kt/2 \pm 45^0)$ Trigonometric Form, when k is odd

4.
(a) x(t) = 1 + sin(wt) + 2 cos(wt) + cos(2wt +π/4)
$= 1 + (e^{jwt} - e^{-jwt})/2j + 2(e^{jwt} + e^{-jwt})/2 + (e^{j(2wt+\pi/4)} + e^{-j(2wt+\pi/4)})/2$
$= 1 - 0.5je^{jwt} + 0.5je^{-jwt} + e^{jwt} + e^{-jwt} + 0.5e^{j\pi/4}e^{j2wt} + 0.5e^{-j\pi/4}e^{-j2wt}$
$x(t) = 1 + (1-0.5j)e^{jwt} + (1+0.5j)e^{-jwt} + 0.5e^{j\pi/4}e^{j2wt} + 0.5e^{-j\pi/4}e^{-j2wt}$
$a_0 = 1 -> |a_0| = 1, \theta_0 = 0$
$a_1 = (1-0.5j) -> |a_1| = \sqrt{5}/2, \theta_1 = tan^{-}(-0.5)$
$a_{-1} = (1+0.5j) -> |a_{-1}| = \sqrt{5}/2, \theta_{-1} = tan^{-}(0.5)$
$a_2 = (0.5e^{j\pi/4}) -> |a_2| = 0.5, \theta_1 = \pi/4$
$a_{-2} = (0.5e^{-j\pi/4}) -> |a_{-2}| = 0.5, \theta_1 = -\pi/4$





(b)

H(s) = Y(s) X(s)
dy/dt + y(t) = x(t)
sY(s) - Y(0^-) $+ Y(s) = X(s)$
$Y(s)(s+1) = X(s)$
$H(s) = Y(s)/X(s) = 1/(s+1)$
The eigenvalue of the system is $(1/1 + w_o)$

(c)

$$b_k jwk + b_k = c_k b_k = \frac{c_k}{1 + jwk}$$

(d)
y(t) = $\sum_{-\infty}^{\infty} b_k e^{jkwt}$
$y(t) = \sum_{-\infty}^{\infty} \frac{c_k}{1+jwk} e^{jkwt}$

5.
(a)
x[n] = sin($\pi n/2$)
$x[n] = (e^{j\pi n/2} - e^{-j\pi n/2})/2j$
$x[n] = -0.5je^{j\pi n/2} + 0.5je^{-j\pi n/2}$
$a_1 = -0.5j$
$a_{-1} = 0.5j$

(b)
$y[n] = 1 + cos(\pi n/2)$
$y[n] = 1 + 0.5e^{j\pi n/2} + 0.5e^{-j\pi n/2}$
$a_o = 1$
$a_1 = 0.5$
$a_{-1} = 0.5$

(c)
By multiplication property, the coefficients of the fourier series of x[n]y[n] is $c_k = \sum_{l=-\infty}^{\infty} a_l b_{k-l}$

$$c_k = \sum_{l=0}^{3} a_l b_{k-l} = a_0 b_k + a_1 b_{k-l} + a_2 b_{k-2} + a_3 b_{k-3}$$

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

$$c_0 = a_1 b_{-1} + a_{-1} b_{-3} = -j/4 + j/4 = 0$$

$$c_1 = a_1 b_0 + a_{-1} b_{-2} = -j/2 + 0 = -0.5j$$

$$c_2 = a_1 b_1 + a_{-1} b_{-1} = -j/4 + j/4 = 0$$

$$c_3 = a_{-1} b_{-2} + a_1 b_{-4} = a_1 b_0 = j/2$$

(d)

$$x[n]y[n] = sin(\pi n/2)(1 + cos(\pi n/2)) = sin(\pi n/2) + cos(\pi n/2)sin(\pi n/2)$$
$$= sin(\pi n/2) + 0.5sin(\pi n) = -0.5j[e^{j\pi n/2} - e^{-j\pi n/2} + 0]$$
$$c_0 = 0, c_1 = 0.5j, c_{-1} = -0.5j$$

The result is in the same as the result in part c

6.
(a)
$T = 4s, w = \pi/2 rad/s, x[n] = \sum_{k<N>} a_k e^{jk\pi n/2}$

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

$$a_k = 0.25[\sum_{n=0}^{3} x[n]e^{-jk\pi n/2}]$$

$$= 0.25[x(1)e^{\frac{jk\pi}{2}} + x(2)e^{jk\pi} + x(3)e^{\frac{jk3\pi}{2}}]$$

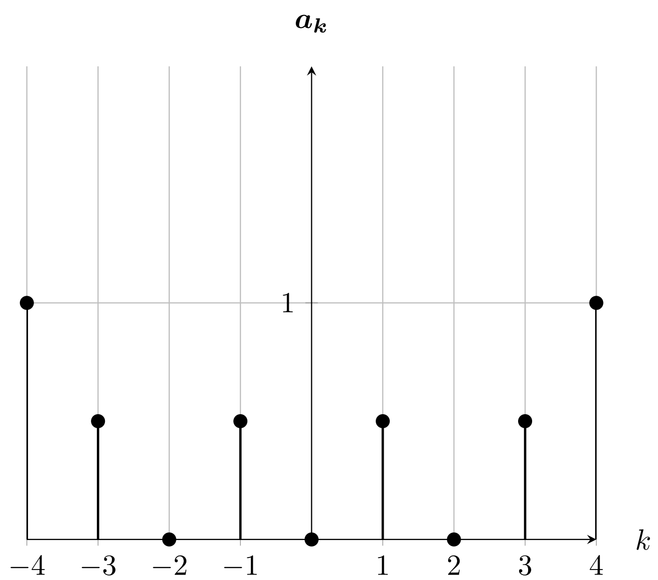
$$= 0.25[e^{\frac{jk\pi}{2}} + 2e^{jk\pi} + e^{\frac{jk3\pi}{2}}]$$

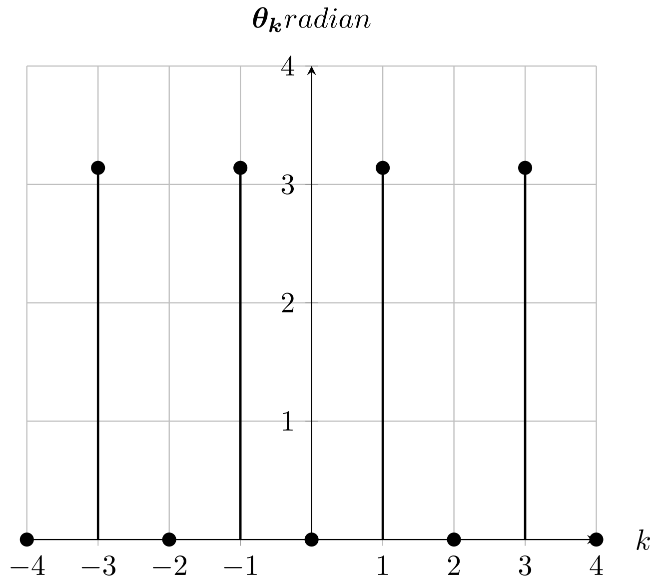
$$e^{\frac{jk\pi}{2} = cos(\frac{k\pi}{2}) + jsin(\frac{k\pi}{2})}$$

$$e^{jk\pi} = cos(k\pi) + jsin(k\pi)$$

$$e^{\frac{jk3\pi}{2} = cos(\frac{k3\pi}{2}) + jsin(\frac{k3\pi}{2})}$$

$$a_0 = 0.25[1 + 2 + 1] = 1, a_0 = 1$$
$$a_1 = 0.25[j - 2 - j] = -0.5, a_1 = -0.5$$
$$a_2 = 0.25[-1 + 2 - 1] = 0, a_2 = 0$$
$$a_3 = 0.25[-j - 2 + j] = -0.5, a_3 = -0.5$$
$$a_k = a_{k+4} \text{ for all k}$$





(b.i)

$$y[n] = x[n] - \delta[n+5] - \delta[n+5] - \delta[n+1] - \delta[n-3] - \delta[n-7] - ... = x[n] - \sum_{k=-\infty}^{\infty} \delta[n+4k+1]$$

(b.ii)
$T = 4s, w = \pi/2 rad/s, y[n] = \sum_{k<N>} a_k e^{jk\pi n/2}$

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

$$b_k = 0.25[\sum_{n=0}^{3} y[n] e^{-jk\pi n/2}]$$

$$= 0.25[y(1)e^{\frac{jk\pi}{2}} + y(2)e^{jk\pi}]$$

$$= 0.25[e^{\frac{jk\pi}{2}} + 2e^{jk\pi}]$$

$$e^{\frac{jk\pi}{2} = cos(\frac{k\pi}{2}) + jsin(\frac{k\pi}{2})}$$

$$e^{jk\pi} = cos(k\pi) + jsin(k\pi)$$

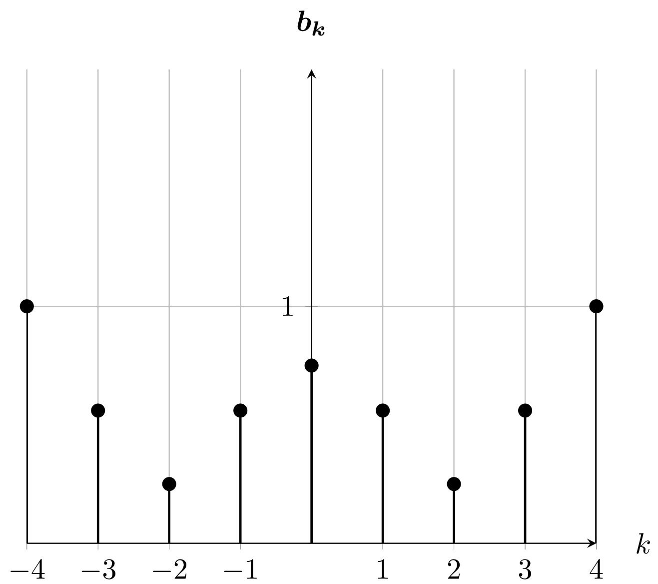
$$b_0 = 0.25[1+2] = 3/4, b_0 = 3/4$$

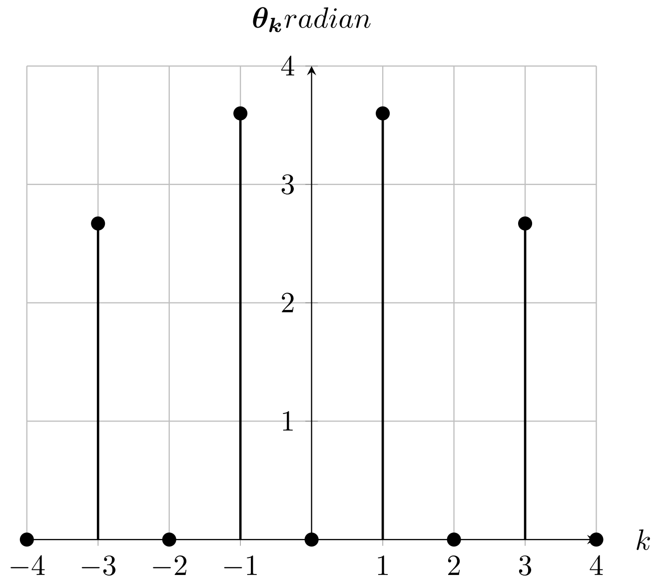
$$b_1 = 0.25[-j-2] = \frac{-2-j}{4}, b_1 = \frac{-2-j}{4}$$

$$b_2 = 0.25[-1+2] = 0.25, b_2 = 0.25$$

$$b_3 = 0.25[j-2] = \frac{-2+j}{4}, b_3 = \frac{-2+j}{4}$$

$$b_k = b_{k+4} \text{ for all k}$$





7.

(a)

if y(t) = x(t) it means that the system does not change the input signal, so the coefficients $a_k of x(t)$ are non zero only for frequencies $|w_k| <= 80$.

(b)

If $y(t) \neq x(t)$ we conclude that the coefficients a_k are non-zero for both inside and outside the range $[-80, 80]$ and for those outside the range x(t) is being affected.

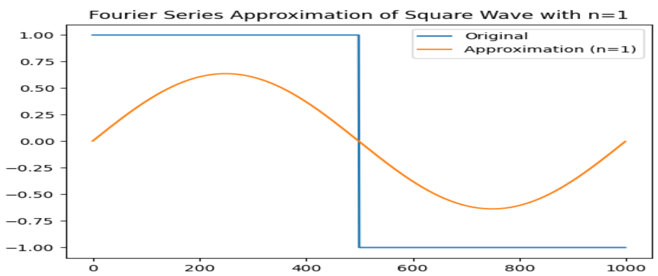
8.

```
import numpy as np
import matplotlib.pyplot as plt
from scipy import signal

# (a)
def compute_fourier_coeffs(signal, T, n):
    N = len(signal)
    t = np.array([(-T/2) + (i * T/N) for i in range(N)])
    coeffs = []

    # DC component equal to the average value of the signal over a period
    a_0 = np.sum(signal)/N
    coeffs.append(a_0)

    # cosine and sine coefficients
    for k in range(1, n+1):
        cos_terms = signal * np.cos(2*np.pi*k*t/T)
        sin_terms = signal * np.sin(2*np.pi*k*t/T)

        a_k = np.sum(cos_terms)/N
        b_k = np.sum(sin_terms)/N

        coeffs.append((a_k, b_k))

    return coeffs

# (b)
def approx_fourier(coeffs, T, n, N):
    t = np.array([(-T/2) + (i * T/N) for i in range(N)])
    approx = np.zeros(N) + coeffs[0]

    # add cosine and sine terms for each coefficient
    for k in range(1, n+1):
        a_k, b_k = coeffs[k]
        cos_terms = a_k * np.cos(2*np.pi*k*t/T)
        sin_terms = b_k * np.sin(2*np.pi*k*t/T)
```

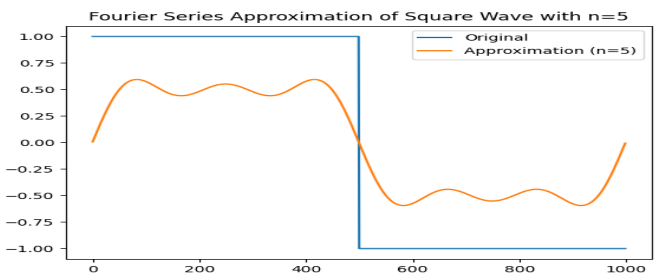
```

        approx += cos_terms + sin_terms
    return approx

# (c)
N = 1000
T = 1
t_arr = np.array([(-T/2) + (i * T) for i in range(N)])
s = np.array([1 if i < N/2 else -1 for i in range(N)])
for n in [1, 5, 10, 50, 100]:
    coeffs = compute_fourier_coeffs(s, T, n)
    approx = approx_fourier(coeffs, T, n, N)
    plt.plot(t_arr, s, label='Original')
    plt.plot(t_arr, approx, label=f'Approximation (n={n})')
    plt.title(f'Fourier Series Approximation of Square Wave with n={n}')
    plt.legend()
    plt.show()

# (d)
sawtooth_wave = signal.sawtooth(2 * np.pi * t_arr)
for n in [1, 5, 10, 50, 100]:
    coeffs = compute_fourier_coeffs(sawtooth_wave, T, n)
    approx = approx_fourier(coeffs, T, n, N)
    plt.plot(t_arr, sawtooth_wave, label='Original')
    plt.plot(t_arr, approx, label=f'Approximation (n={n})')
    plt.title(f'Fourier Series Approximation of Sawtooth Wave with n={n}')
    plt.legend()
    plt.show()

# When we increase the number of Fourier series coefficients,
# the approximation becomes more accurate, and the reconstructed
# signal will be closer to the original signal.
```

C.

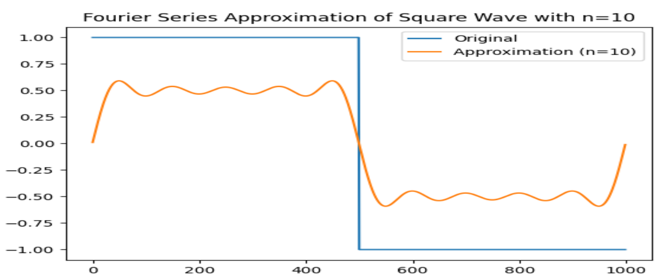
when n = 1



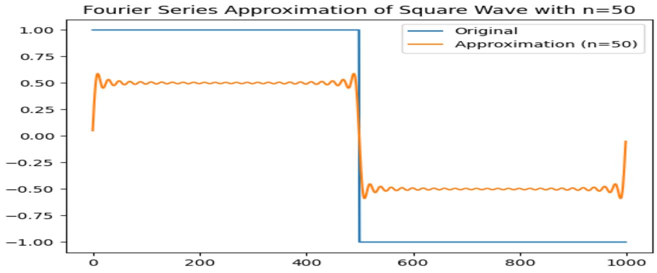
when n = 5



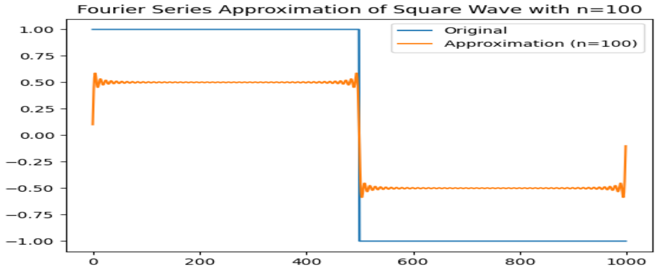
when n = 10



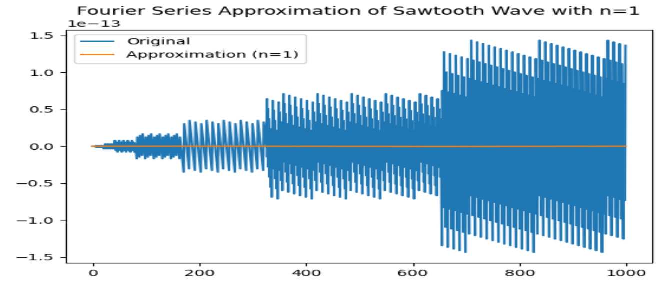
when n = 50



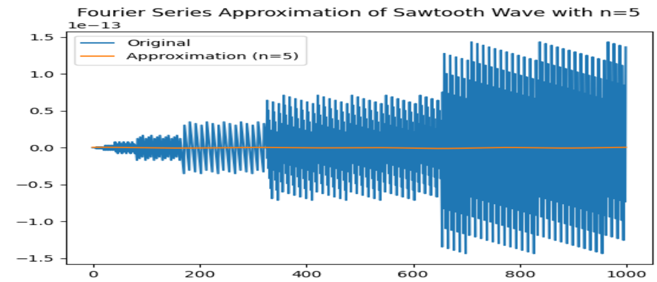
when n = 100



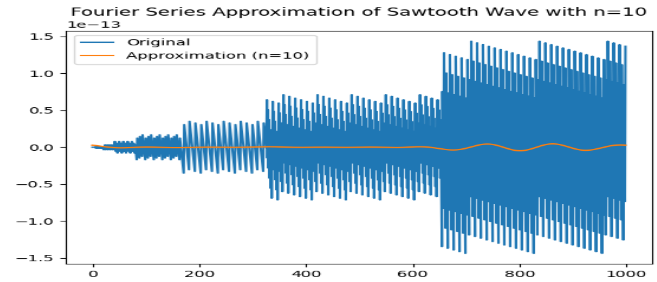
d.
when n = 1



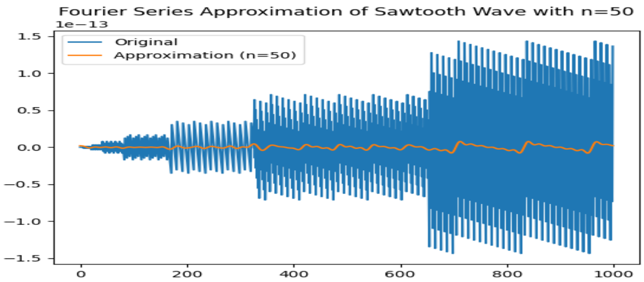
when n = 5



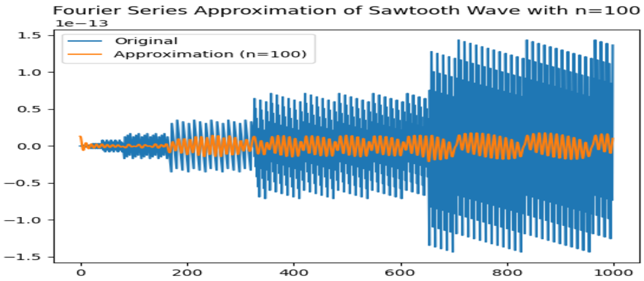
when n = 10



when n = 50


Fourier Series Approximation of Sawtooth Wave with n=50
1e−13
Original
Approximation (n=50)
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0
200
400
600
800
1000

when n = 100


Fourier Series Approximation of Sawtooth Wave with n=100
1e−13
Original
Approximation (n=100)
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0
200
400
600
800
1000