EXPERIMENT

**Plot for Magnitude and Phase Spectrum for Signum Function**

clear all;

close all;

clc;

% Time and frequency parameters

time\_period = 2 \* pi;

step\_size = 0.001;

t = -time\_period/2 : step\_size : time\_period/2; % Time vector

tt = length(t);

omega\_o = 2 \* pi / time\_period;

no\_of\_forier\_coeff = 50;

n = -no\_of\_forier\_coeff : 1 : no\_of\_forier\_coeff;

% Initialize signal x\_t

x\_t = zeros(1, tt); % Pre-allocate for speed

% Define the signal x\_t

for ii = 1:tt

if (t(ii) >= -pi && t(ii) <= 0)

x\_t(ii) = -1;

elseif t(ii) > 0 && t(ii) <= pi

x\_t(ii) = 1;

else

x\_t(ii) = 0;

end

end

% Plot the signal x\_t

figure;

plot(t, x\_t);

title('Signum x(t)');

xlabel('Time');

ylabel('Amplitude');

% Fourier Coefficients calculation

c\_n = zeros(1, length(n)); % Initialize c\_n for Fourier coefficients

for ii = 1:length(n) % Loop over Fourier coefficients

temp = x\_t .\* exp(-1j \* omega\_o \* n(ii) \* t); % Element-wise multiplication

int\_ans = my\_int\_func(temp, step\_size); % Call the integration function

c\_n(ii) = (1 / time\_period) \* int\_ans; % Calculate the Fourier coefficient

end

% Plot the magnitude and phase of Fourier coefficients

figure;

stem(n, abs(c\_n));

title('Magnitude of Fourier Coefficients');

xlabel('n');

ylabel('|c\_n|');

figure;

stem(n, angle(c\_n));

title('Phase of Fourier Coefficients');

xlabel('n');

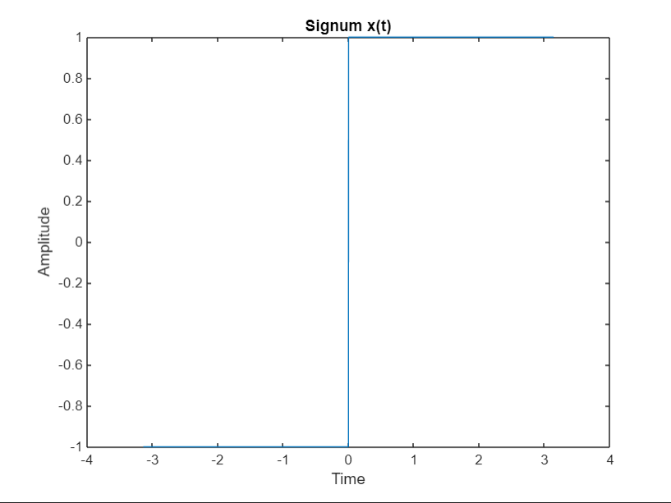
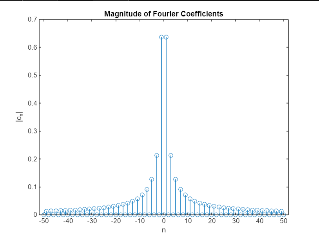
ylabel('Phase of c\_n (radians)');

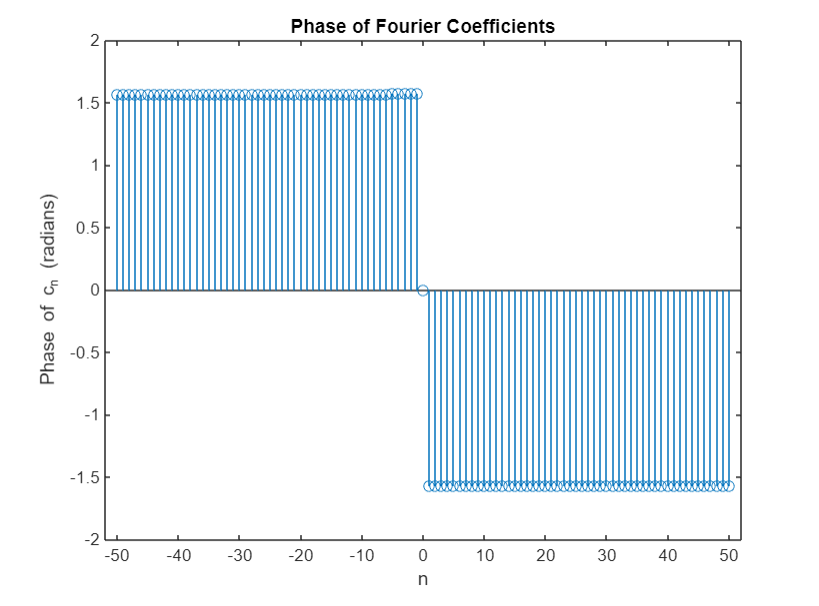
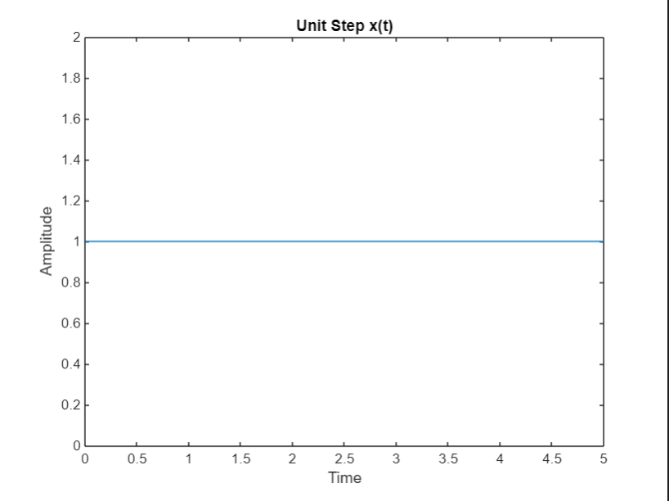
% Integration function for trapezoidal rule

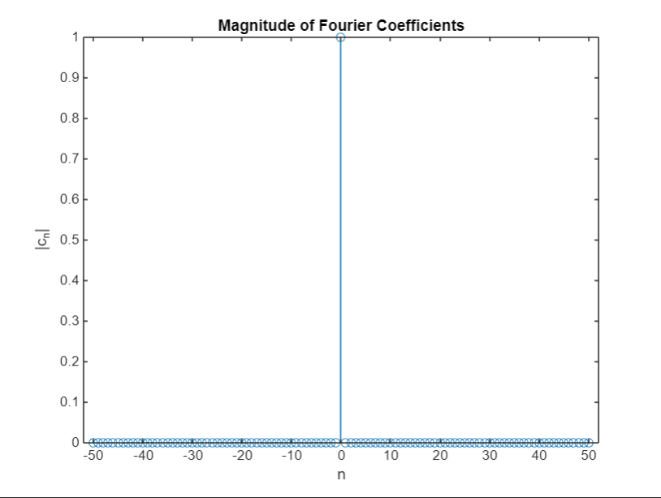
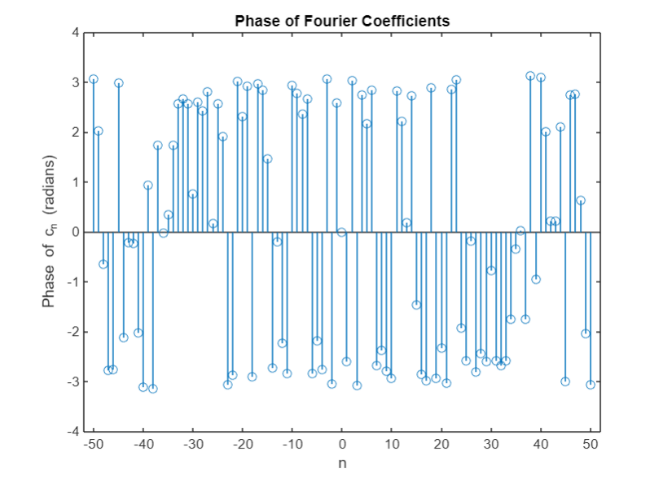
function[int\_ans] = my\_int\_func(Data, step\_size)

int\_ans = (step\_size / 2) \* (Data(1) + Data(end) + 2 \* sum(Data(2:end-1)));

end



**Plot for Magnitude and Phase Spectrum for Unit Step Function**

clear all;

close all;

clc;

% Time and frequency parameters

time\_period = 5; % Define the time period

step\_size = 0.001; % Step size

t = 0:step\_size:time\_period; % Time vector

tt = length(t);

omega\_o = 2 \* pi / time\_period;

no\_of\_forier\_coeff = 50; % Number of Fourier coefficients

n = -no\_of\_forier\_coeff : 1 : no\_of\_forier\_coeff; % Range of Fourier coefficients

% Initialize signal x\_t

x\_t = zeros(1, tt); % Pre-allocate for speed

% Define the signal x\_t

for ii = 1:tt

if (t(ii) >=0)

x\_t(ii) = 1;

else

x\_t(ii) = 0;

end

end

% Plot the signal x\_t

figure;

plot(t, x\_t);

title('Unit Step x(t)');

xlabel('Time');

ylabel('Amplitude');

% Fourier Coefficients calculation

c\_n = zeros(1, length(n)); % Initialize c\_n for Fourier coefficients

for ii = 1:length(n) % Loop over Fourier coefficients

temp = x\_t .\* exp(-1j \* omega\_o \* n(ii) \* t); % Element-wise multiplication

int\_ans = my\_int\_func(temp, step\_size); % Call the integration function

c\_n(ii) = (1 / time\_period) \* int\_ans; % Calculate the Fourier coefficient

end

% Plot the magnitude and phase of Fourier coefficients

figure;

stem(n, abs(c\_n)); % Plot magnitude

title('Magnitude of Fourier Coefficients');

xlabel('n');

ylabel('|c\_n|');

figure;

stem(n, angle(c\_n)); % Plot phase

title('Phase of Fourier Coefficients');

xlabel('n');

ylabel('Phase of c\_n (radians)');

% Integration function for trapezoidal rule

function[int\_ans] = my\_int\_func(Data, step\_size)

int\_ans = (step\_size / 2) \* (Data(1) + Data(end) + 2 \* sum(Data(2:end-1)));

end

**Plot for Magnitude and Phase Spectrum for Square Wave Function**

% Fourier Transform

clear all;

close all;

clc;

% Parameters

time\_period = 1; % Set the time period

step\_size\_t = 0.0001; % Step size for time vector

t = -time\_period/2:step\_size\_t:time\_period/2; % Time vector

tt = length(t); % Length of the time vector

omega\_o = 2 \* pi / time\_period; % Fundamental frequency

no\_of\_fourier\_coeff = 50; % Number of Fourier coefficients

n = -no\_of\_fourier\_coeff:1:no\_of\_fourier\_coeff; % Coefficient index

% Define the square wave signal

x = zeros(size(t)); % Preallocate x for efficiency

for ii = 1:tt

if (t(ii) < -0.25)

x(ii) = -1; % Set value for t < -0.25

elseif (t(ii) >= -0.25 && t(ii) < 0.25)

x(ii) = 1; % Set value for -0.25 <= t < 0.25

else

x(ii) = -1; % Set value for t >= 0.25

end

end

% Plot the square wave signal

figure;

plot(t, x)

xlabel("Time (t)");

ylabel("x(t)");

title("Square Wave Signal");

grid on; % Add grid for better readability

% Fourier series coefficients calculation using my\_int\_fun

c = zeros(size(n)); % Preallocate c for Fourier coefficients

for ii = 1:length(n) % Loop over the range of n

temp1 = x .\* exp(-1j \* omega\_o \* n(ii) \* t); % Calculate the product

int\_ans = my\_int\_fun(temp1, step\_size\_t); % Use custom integration function

c(ii) = (1 / time\_period) \* int\_ans; % Compute Fourier coefficient

end

% Plot the magnitude of Fourier coefficients

figure;

stem(n, abs(c)); % Plot the magnitude

xlabel('n');

ylabel('|C\_n|');

title('Magnitude of Fourier Coefficients');

grid on;

% Plot the phase of Fourier coefficients

figure;

stem(n, angle(c)); % Plot the phase

xlabel('n');

ylabel('Phase of C\_n (radians)');

title('Phase of Fourier Coefficients');

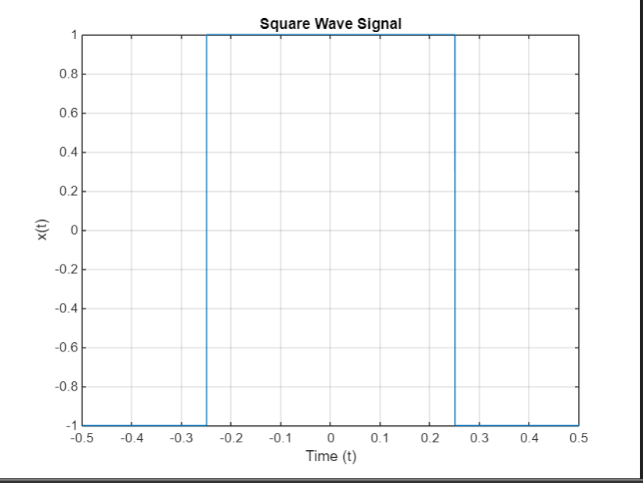
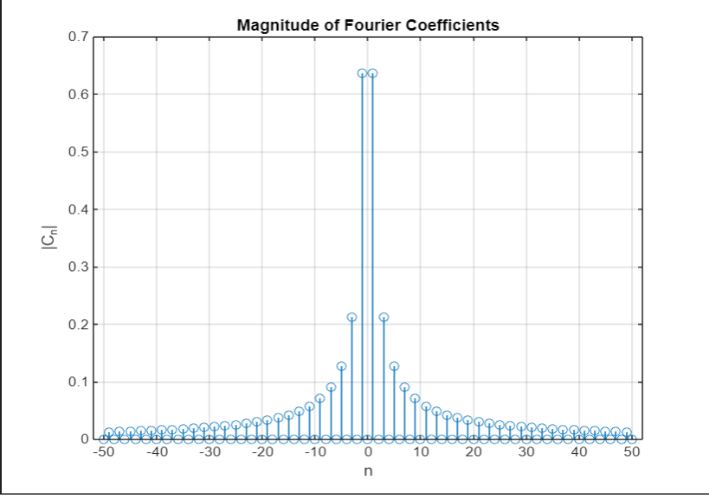
grid on;

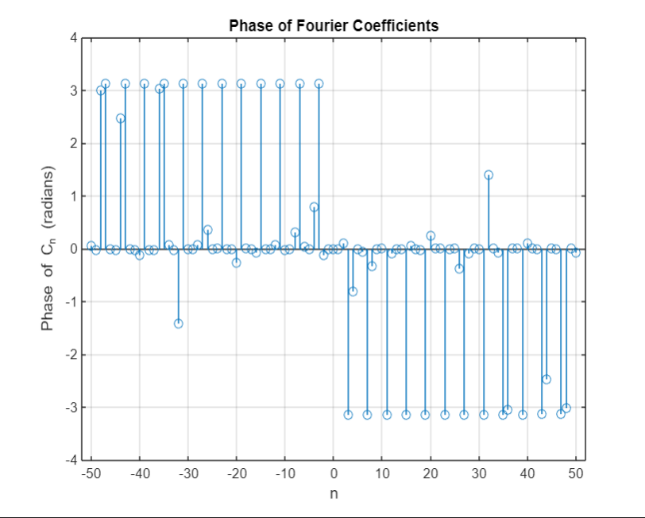
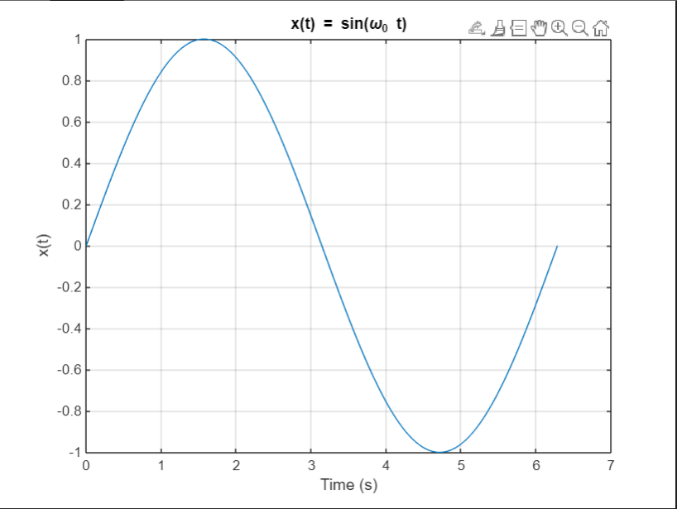
% Custom integration function

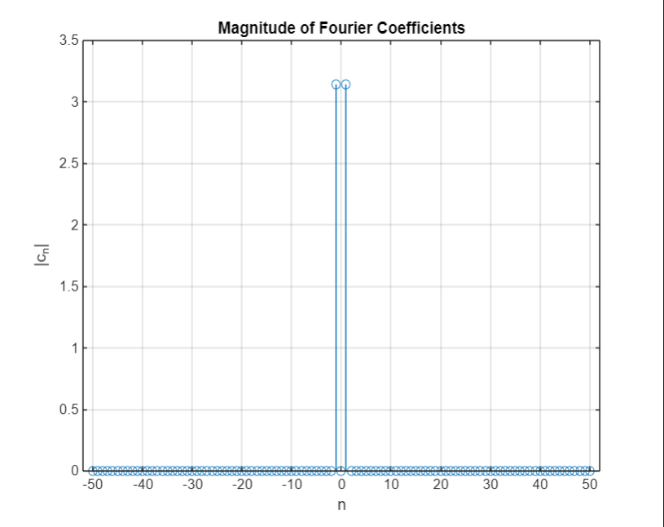
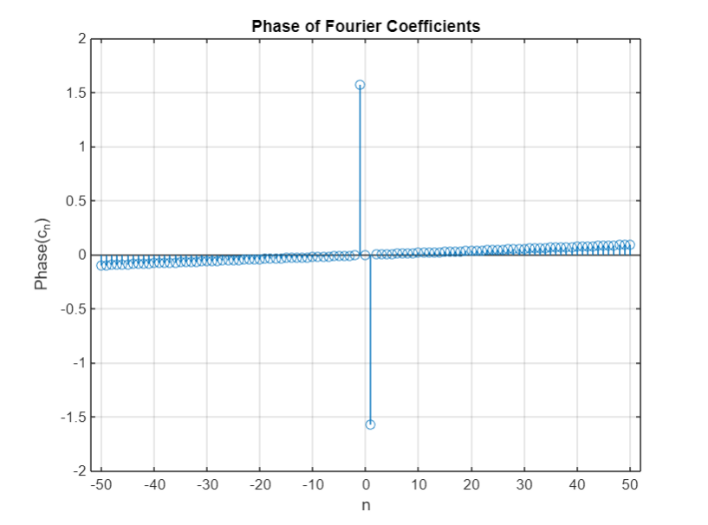
function [int\_ans] = my\_int\_fun(data, step\_size)

int\_ans = (step\_size / 2) \* (data(1) + data(end) + 2 \* sum(data(2:end-1))); % Corrected sum range

end

**Plot for Magnitude and Phase Spectrum for Sine Wave Function**

clear all;

close all;

clc;

% Define parameters

time\_period = 2 \* pi; % Set the time period to 2\*pi

step\_size = 0.001; % Step size for time vector

t = 0:step\_size:time\_period; % Time vector from 0 to time\_period

tt = length(t); % Length of the time vector

omega\_o = 2 \* pi / time\_period; % Fundamental frequency

no\_of\_fourier\_coeff = 50; % Number of Fourier coefficients

n = -no\_of\_fourier\_coeff:1:no\_of\_fourier\_coeff; % Coefficient index

% Define the function x\_t

x\_t = sin(omega\_o \* t); % x\_t is the sine wave based on the fundamental frequency

% Plot x\_t

figure;

plot(t, x\_t);

title('x(t) = sin(\omega\_0 t)');

xlabel('Time (s)');

ylabel('x(t)');

grid on;

% Initialize Fourier coefficients array

c\_n = zeros(size(n));

% Calculate Fourier coefficients

for k = 1:length(n)

omega = omega\_o \* n(k); % Angular frequency for the Fourier term

% Use the integration function for trapezoidal rule

c\_n(k) = my\_int\_func(x\_t .\* exp(-1i \* omega \* t), step\_size); % Compute Fourier coefficient

end

% Plot magnitude of Fourier coefficients

figure;

stem(n, abs(c\_n));

title('Magnitude of Fourier Coefficients');

xlabel('n');

ylabel('|c\_n|');

grid on;

% Plot phase of Fourier coefficients

figure;

stem(n, angle(c\_n));

title('Phase of Fourier Coefficients');

xlabel('n');

ylabel('Phase(c\_n)');

grid on;

% Integration function for trapezoidal rule

function[int\_ans] = my\_int\_func(Data, step\_size)

int\_ans = (step\_size / 2) \* (Data(1) + Data(end) + 2 \* sum(Data(2:end-1)));

end

**ASSIGNMENT**

**Plot for Magnitude and Phase Spectrum for Cosine Wave Function**

clear all;

close all;

clc;

% Define parameters

time\_period = 2 \* pi; % Set the time period to 2\*pi

step\_size = 0.001; % Step size for time vector

t = 0:step\_size:time\_period; % Time vector from 0 to time\_period

tt = length(t); % Length of the time vector

omega\_o = 2 \* pi / time\_period; % Fundamental frequency (omega\_0)

no\_of\_fourier\_coeff = 50; % Number of Fourier coefficients

n = -no\_of\_fourier\_coeff:1:no\_of\_fourier\_coeff; % Coefficient index

% Define the cosine wave function x\_t

x\_t = cos(omega\_o \* t); % x\_t is the cosine wave based on the fundamental frequency

% Plot x\_t (cosine wave)

figure;

plot(t, x\_t, 'LineWidth', 1.5);

title('x(t) = cos(\omega\_0 t)');

xlabel('Time (s)');

ylabel('x(t)');

grid on;

% Initialize Fourier coefficients array

c\_n = zeros(size(n));

% Calculate Fourier coefficients

for k = 1:length(n)

omega = omega\_o \* n(k); % Angular frequency for the Fourier term

% Use the integration function for trapezoidal rule

c\_n(k) = my\_int\_func(x\_t .\* exp(-1i \* omega \* t), step\_size); % Compute Fourier coefficient

end

% Plot magnitude of Fourier coefficients

figure;

stem(n, abs(c\_n), 'LineWidth', 1.5);

title('Magnitude of Fourier Coefficients');

xlabel('n');

ylabel('|c\_n|');

grid on;

% Plot phase of Fourier coefficients

figure;

stem(n, angle(c\_n), 'LineWidth', 1.5);

title('Phase of Fourier Coefficients');

xlabel('n');

ylabel('Phase(c\_n)');

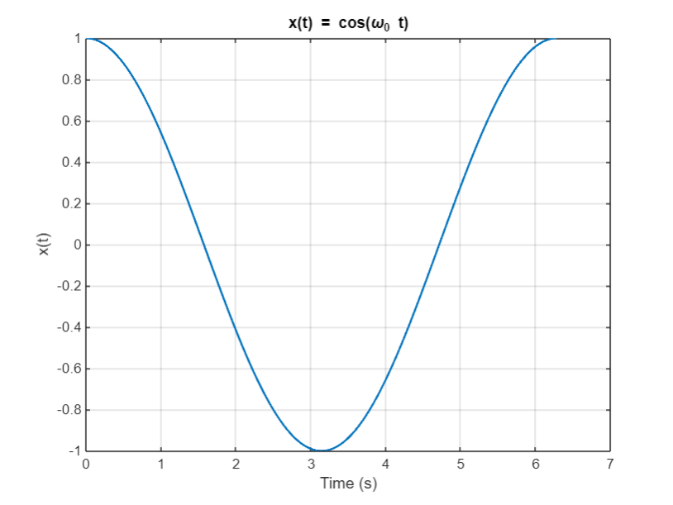
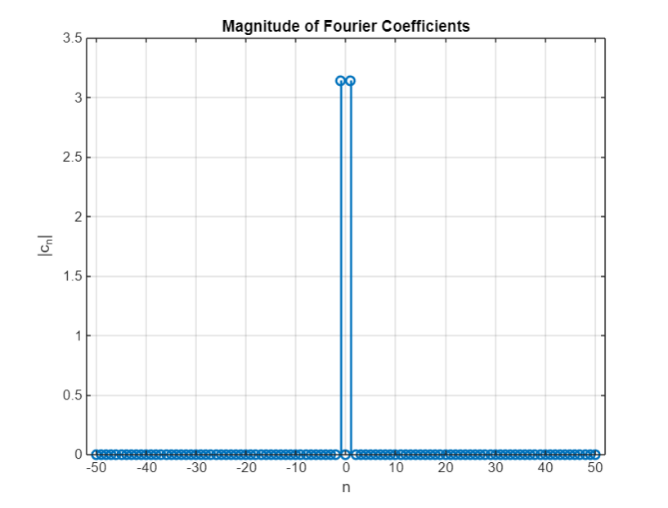
grid on;

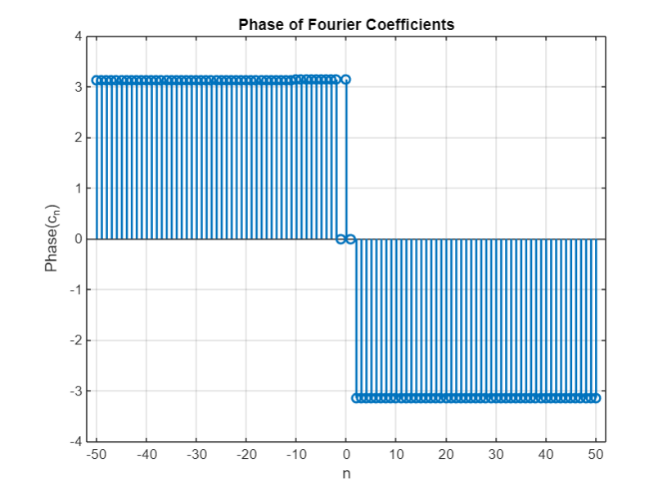
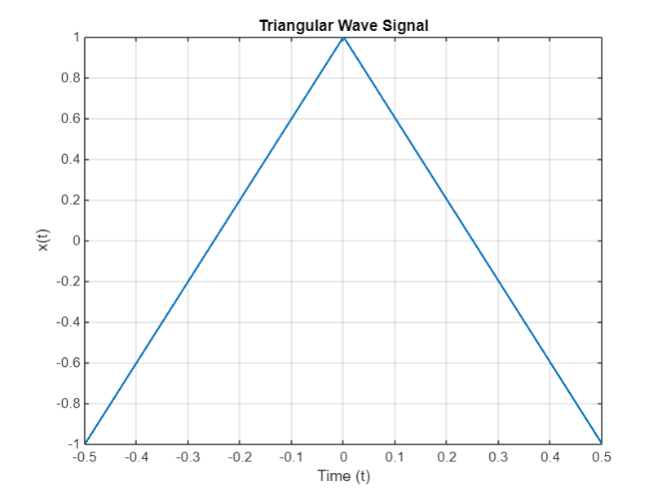
% Integration function for trapezoidal rule

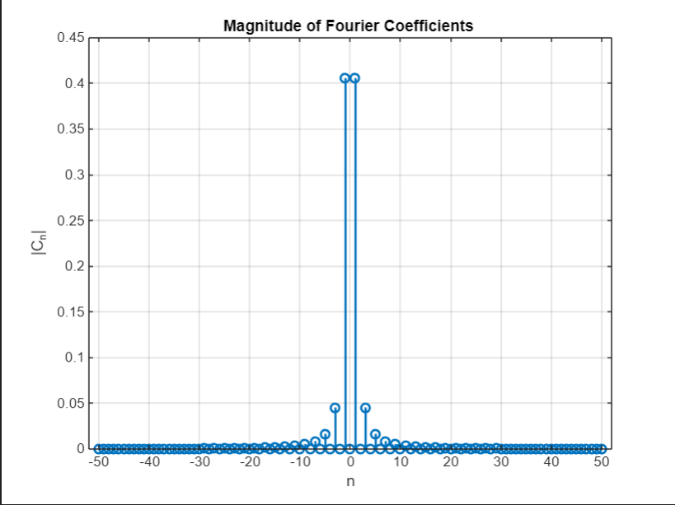
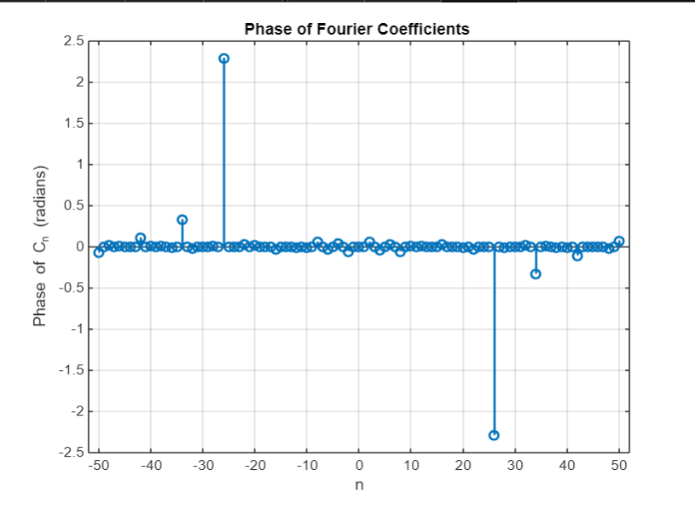
function[int\_ans] = my\_int\_func(Data, step\_size)

int\_ans = (step\_size / 2) \* (Data(1) + Data(end) + 2 \* sum(Data(2:end-1)));

end

**Plot for Magnitude and Phase Spectrum for Triangular Wave Function**

clear all;

close all;

clc;

% Parameters

time\_period = 1; % Set the time period (T)

step\_size\_t = 0.0001; % Step size for time vector

t = -time\_period/2:step\_size\_t:time\_period/2; % Time vector

tt = length(t); % Length of the time vector

omega\_o = 2 \* pi / time\_period; % Fundamental frequency (omega\_0)

no\_of\_fourier\_coeff = 50; % Number of Fourier coefficients

n = -no\_of\_fourier\_coeff:1:no\_of\_fourier\_coeff; % Coefficient index (n)

% Define the triangular wave signal

x = zeros(size(t)); % Preallocate x for efficiency

for ii = 1:tt

if (t(ii) < 0)

x(ii) = 1 + (4 / time\_period) \* t(ii); % Slope for t < 0

else

x(ii) = 1 - (4 / time\_period) \* t(ii); % Slope for t >= 0

end

end

% Plot the triangular wave signal

figure;

plot(t, x, 'LineWidth', 1.5);

xlabel("Time (t)");

ylabel("x(t)");

title("Triangular Wave Signal");

grid on;

% Fourier series coefficients calculation using custom integration function

c = zeros(size(n)); % Preallocate c for Fourier coefficients

for ii = 1:length(n) % Loop over the range of n

temp1 = x .\* exp(-1j \* omega\_o \* n(ii) \* t); % Calculate the product x(t)\*exp(-j\*n\*omega\_0\*t)

int\_ans = my\_int\_fun(temp1, step\_size\_t); % Use custom integration function

c(ii) = (1 / time\_period) \* int\_ans; % Compute Fourier coefficient

end

% Plot the magnitude of Fourier coefficients

figure;

stem(n, abs(c), 'LineWidth', 1.5); % Plot the magnitude

xlabel('n');

ylabel('|C\_n|');

title('Magnitude of Fourier Coefficients ');

grid on;

% Plot the phase of Fourier coefficients

figure;

stem(n, angle(c), 'LineWidth', 1.5); % Plot the phase

xlabel('n');

ylabel('Phase of C\_n (radians)');

title('Phase of Fourier Coefficients ');

grid on;

% Custom integration function (Trapezoidal Rule)

function [int\_ans] = my\_int\_fun(data, step\_size)

int\_ans = (step\_size / 2) \* (data(1) + data(end) + 2 \* sum(data(2:end-1))); % Corrected sum range for trapezoidal integration

end