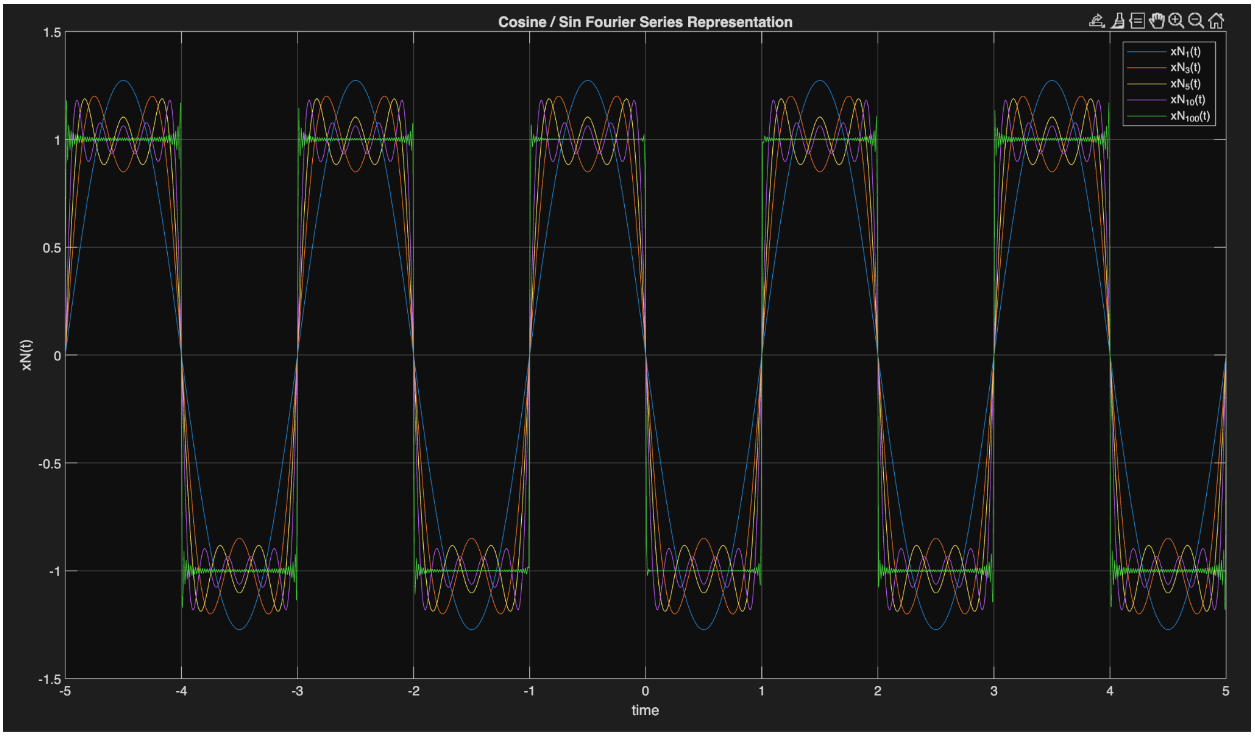
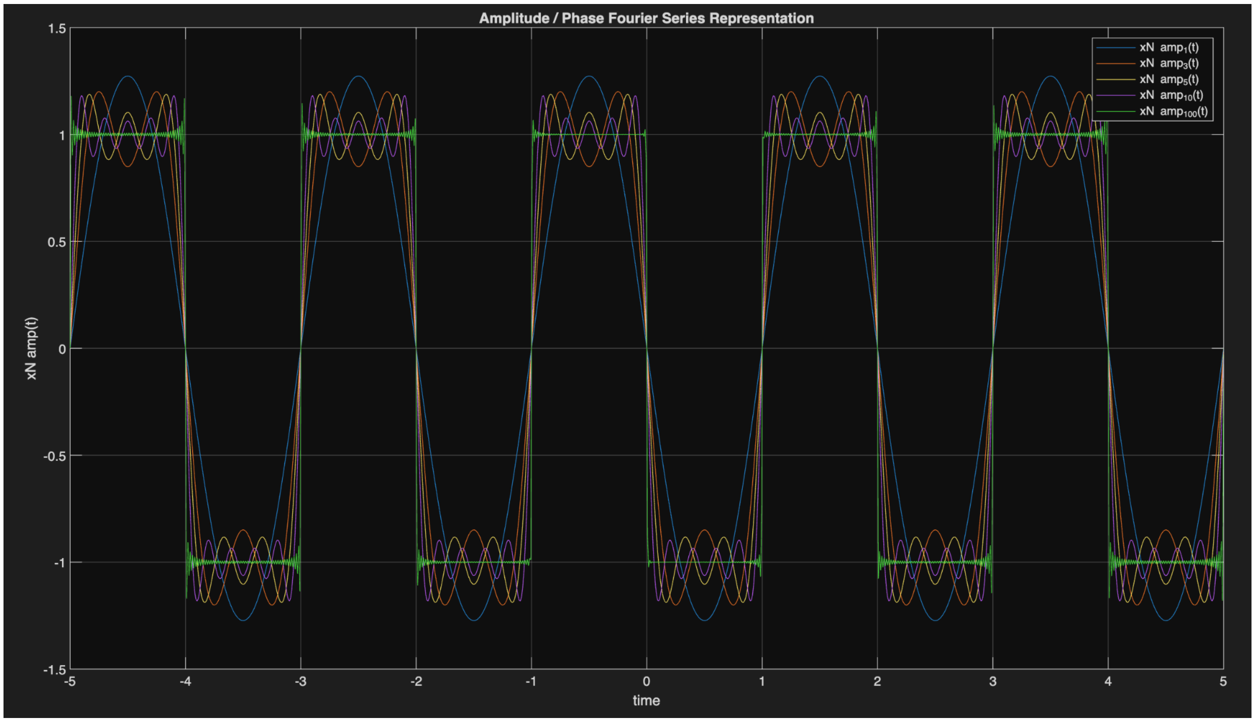
Luis Kligman Computer Assignment 7

Problem 1:



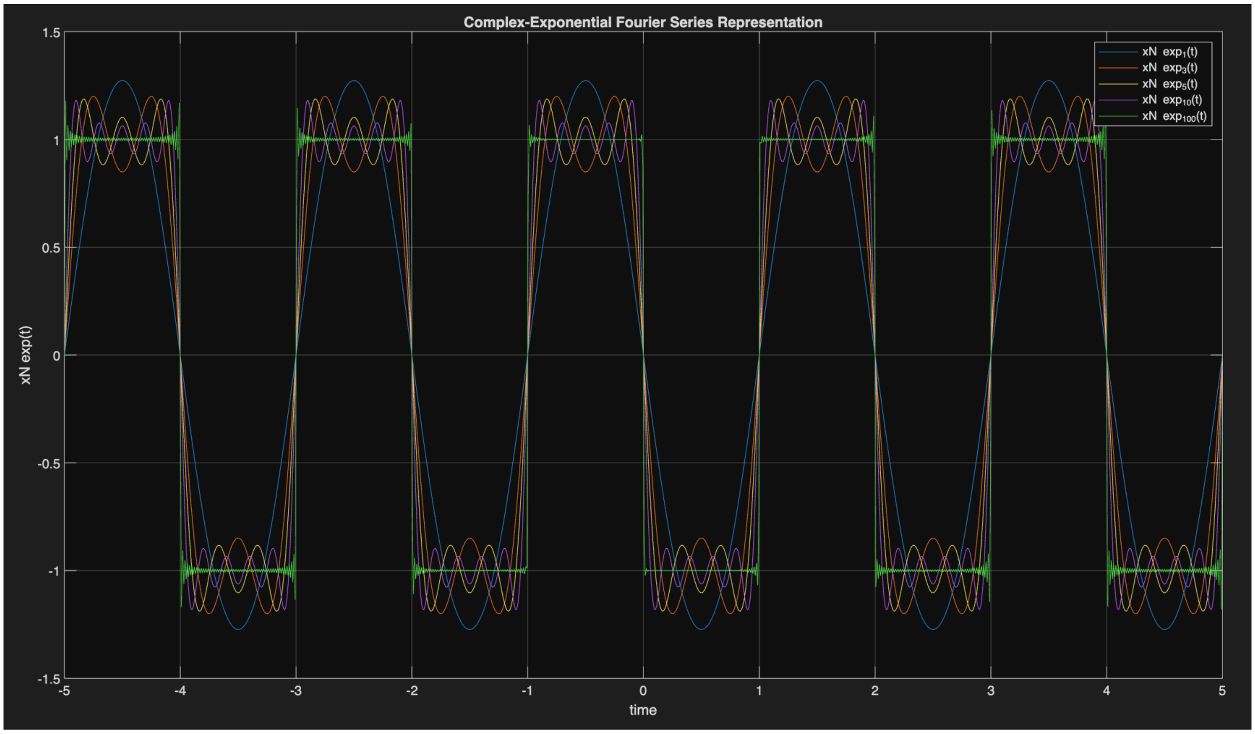
For problem one, we first found a_0, a_n, and b_n on paper due to the complexities of solving for these in MATLAB. When solving these, a_0 and a_n ended up being zero, which greatly simplified our future calculations, while b_n -4/npi for odd n, and 0 for even n. Once I had these, I was able to generate my x(t) function, which took the matrix N that held the values of the harmonics. By using a for loop, we can plot each x(t, N) for each n on a single plot.

Problem 2:



From our calculations in problem one, we know $c_0 = 0$ and b_n is the only useful term. We calculate c_n to be abs((-4./(N*pi)) .* (mod(N,2)==1)) so that way it is only valid for even n. We are also able to easily find c theta through this simplification (theta_n = -tan^-1 (b_n/0) = -tan^-1 (-inf) --> pi/2) to solve that theta = pi/2. With these pieces, we can now build our x_n in the amplitude/phase Fourier representation. Then, as done in problem one, we create a for loop to increment over our different N harmonic values and add each N for t in the range to our graph as a new line. Plotting all five harmonics on the same graph. This graph should match the graph from problem one, as although they are different representations, they both represent the same function.

Problem 3:



For problem three, since my a_0=c_0=0 for n = 0, my x_n was 0. For values greater than zero, x_n is 2j/npi, which I got from simplifying using Euler's formula. Once I had these pieces solved, I could build the x_n function for the complex-exponential Fourier series representation. This function was saved by a variable called xN_exp and was used to plot this function for each N harmonic for all t values in the given range. This was plotted by using a for loop that would calculate all the values using this specific N value and plot to the figure; hold was enabled so MATLAB plotted all harmonics to the same graph, so we could compare as the N value went up. This graph is identical to the graphs obtained in problems one and two, which makes sense because each one of these problems proves a different representation of the same function.

Appendix:

```
clear all;
close all;
clc;
%% Solving for a0, an, bn, and stating xN
T0 = 2;  % Period
w0 = pi;
% x(t) = -1 from 0 to 1
%         1 from 1 to 2
% Calculated a0, an, and bn on paper due to the complexity of solving them
% in MATLAB
a0 = 0;
an = 0;
bN = @(N) (-4./(N*pi)).*(mod(N,2)==1);  % odd n -> -4/npi, even n -> 0
```

```
t = linspace(-5,5,1000);
%% Question 1: cosine/sin Fourier Series Representation
xN = @(t,N) sum(bN(1:N).' .* sin((1:N)'*w0.*t), 1);
for N = [1, 3, 5, 10, 100]
figure(1)
plot(t,xN(t,N),'DisplayName',['xN_{' num2str(N) '}(t)'])
hold on
grid on
xlabel('time')
ylabel('xN(t)')
end
legend('show')
title('Cosine / Sin Fourier Series Representation')
%% Question 2: amplitude/phase Fourier Series Representation
% c_n = sqrt( (b_n)^2 )
% theta_n = -tan^-1 (b_n/0) = -tan^-1 (-inf) --> pi/2
C_n = @(N) abs((-4./(N*pi)) .* (mod(N,2)==1));
theta_n = pi/2;
xN_amp = @(t,N) sum( C_n(1:N).' .* cos((1:N)'*w0.*t + theta_n), 1 );
for N = [1, 3, 5, 10, 100]
figure(2)
plot(t,xN_amp(t,N),'DisplayName',['xN amp_{' num2str(N) '}(t)'])
hold on
grid on
xlabel('time')
ylabel('xN amp(t)')
end
legend('show')
title('Amplitude / Phase Fourier Series Representation')
%% Question 3: complex-exponential Fourier Series Representation
% xn = c0 when n = 0
% xn = cn/2 * e^jtheta_n when n > 0
%x_n = x_n^*
c_of = @(n) (mod(abs(n),2)==1).* (2j./(n*pi));  % 0 for even or n=0, 2j(n*p)
for odd
xN_exp = @(t,N) real( sum( c_of(-N:N).' .* exp(1j* (-N:N)' *w0 .* t), 1));
hold on;
grid on;
for N = [1, 3, 5, 10, 100]
figure(3)
plot(t,xN_exp(t,N),'DisplayName',['xN exp_{' num2str(N) '}(t)'])
hold on
grid on
xlabel('time')
ylabel('xN exp(t)')
end
legend('show')
title('Complex-Exponential Fourier Series Representation')
```