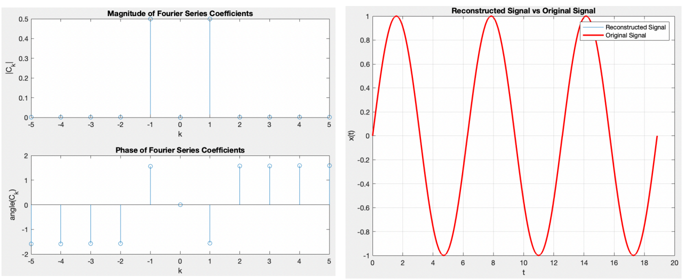
1. During lab 4, we have seen numerical implementation of Fourier Series for periodic signals. As first part of this assignment, you need to write a Matlab function that would take an array representing a single period of a signal (`x`), corresponding time array (`t`), and return the Fourier Series coefficients (`Ck`) in exponential form. The function should also be able to take two (2) optional input arguments: number of Fourier coefficients (`Nk`) and plot option (`p`). Use the template '`fourier_series_exp.m`' for this problem.

Summary:
We define the function to return the Fourier Series coefficients of a signal in exponential form and function as additionally described.

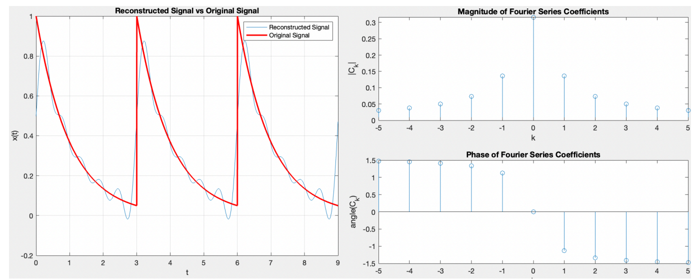
Results:

sin(x), 11 terms, period 2pi




```
>> lab4problem1
1: -0.000003+-0.000101j, 2: -0.000003+-0.000113j, 3: -0.000003+-0.000127j, 4: -0.000003+-0.000146j, 5: -0.000003+-0.000172j
6: -0.000003+-0.000209j, 7: -0.000003+-0.000267j, 8: -0.000004+-0.000375j, 9: -0.000004+-0.000668j, 10: 0.001570+0.499747j
11: 0.000000+0.000000j, 12: 0.001570+-0.499747j, 13: -0.000004+0.000668j, 14: -0.000004+0.000375j, 15: -0.000003+0.000267j
16: -0.000003+0.000209j, 17: -0.000003+0.000172j, 18: -0.000003+0.000146j, 19: -0.000003+0.000127j, 20: -0.000003+0.000113j
21: -0.000003+0.000101j
```

e^-x, 11 terms, period 3




```
>> lab4problem1
1: 0.000672+0.015088j, 2: 0.000841+0.016756j, 3: 0.001076+0.018837j, 4: 0.001419+0.021505j, 5: 0.001946+0.025048j
6: 0.002816+0.029975j, 7: 0.004405+0.037280j, 8: 0.007784+0.049169j, 9: 0.017047+0.071541j, 10: 0.058799+0.123128j
11: 0.316421+0.000000j, 12: 0.058799+-0.123128j, 13: 0.017047+-0.071541j, 14: 0.007784+-0.049169j, 15: 0.004405+-0.037280j
16: 0.002816+-0.029975j, 17: 0.001946+-0.025048j, 18: 0.001419+-0.021505j, 19: 0.001076+-0.018837j, 20: 0.000841+-0.016756j
21: 0.000672+-0.015088j
```

Code:

```
clf; % clear figure
function [Ck]=fourier_series_exp(x,t,Nk,p)
   dT=t(2)-t(1); % calc dT
   T= dT*length(t); % calc T
   w0=2*pi/T;    % calc w0
   if nargin <2 % handle input / default values
       error('Not enough input argument!')
   elseif nargin == 2
       Nk=21;
       p=0;
   elseif nargin ==3
       p=0;
   end
   k=-floor(Nk/2):floor(Nk/2); % calc k
   Ck = zeros(1, length(k)); % Preallocate Ck
   for i = 1:length(k)
       Ck(i) = trapz(t, x .* exp(-1j * k(i) * w0 * t)) * (1/T); % calc fourier
coefficients
   end
   for i = 1:length(k)
           fprintf("%d: %.6f+%fj", i, real(Ck(i)), imag(Ck(i))); % print
formatted
           if mod(i, 5) == 0
               fprintf("\n");
```

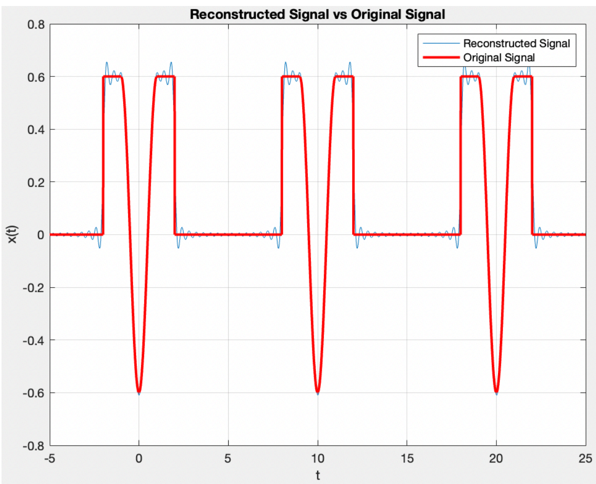
```
            elseif i == length(k)
                fprintf("\n");
            else
                fprintf(", ");
            end
    end
    if p==1
        subplot(2,1,1);
        stem(k,abs(Ck));          % plot abs(Ck) vs k and angle(Ck) vs k
        xlabel('k');
        ylabel('|C_k|');
        title('Magnitude of Fourier Series Coefficients');
        subplot(2,1,2);
        stem(k,angle(Ck)); % plot phase
        xlabel('k');
        ylabel('angle(C_k)');
        title('Phase of Fourier Series Coefficients');
        figure; % plot 3 cycles of the signal 'x' and the reconstructed signal
        t_extended = linspace(min(t),3*(max(t)-min(t))+min(t),3*length(t)); % 3
periods
        x_reconstructed = zeros(1,length(t_extended)); % initialize
reconstruction
        for i = 1:length(k)
            x_reconstructed = x_reconstructed +
Ck(i)*exp(1j*k(i)*w0*t_extended); % reconstruct signal
        end
        plot(t_extended,real(x_reconstructed)); % plot reconstructed
        hold on;
        plot(t_extended, [x, x, x],'r', 'LineWidth', 2); % plot signal
        xlabel('t');
        ylabel('x(t)');
        title('Reconstructed Signal vs Original Signal');
        legend('Reconstructed Signal', 'Original Signal');
        grid on;
    end
end
t = linspace(0, 3, 1000);  % Generate a sample signal
x = exp(-t);                        % Signal
Ck_default = fourier_series_exp(x, t); % Call the function with default
arguments
Ck_custom_Nk = fourier_series_exp(x, t, 10); % specify the number of Fourier
coefficients
Ck_custom_Nk_plot = fourier_series_exp(x, t, 10, 1); % specify the number of
Fourier coefficients and plotting option
```

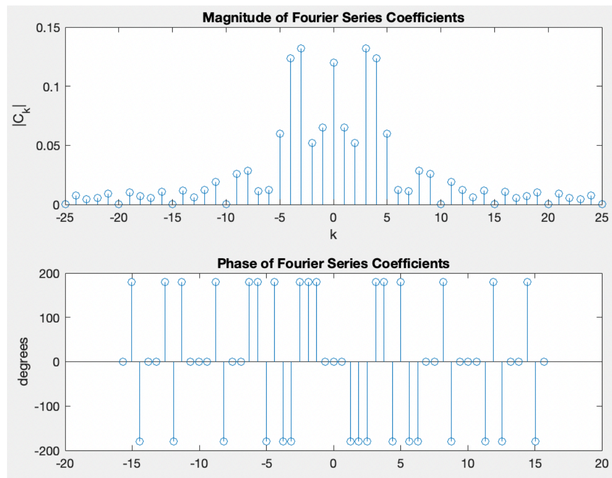
2. A signal $x = 0.6\left\{u(t+2) - (\cos(\pi t) + 1)\left[u(t+1) - u(t-1)\right] - u(t-2)\right\}$ with a period $-5 \le t \le 5$ controls the location of the light source in an optical scanner. Plot the signal for the interval $-5 \le t \le 25$, its spectrum ($|C_k|$ vs ω and $\angle C_k$ vs ω), and reconstructed time domain signal using 51 Fourier Series coefficients. Use the function you have written in problem 1 for solving this problem.

Summary:

We use the code we wrote in #1, and create the function and interval above to pass into it. We user-define a step function to do this. We also change `stem(k,angle(Ck));` to `w0k = w0*k; stem(w0k,angle(Ck)*180/pi);`.

Results:



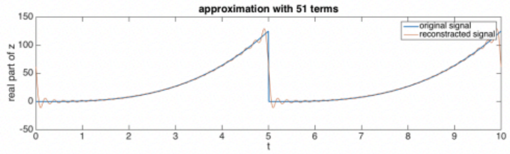


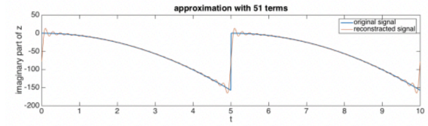

```
>> lab4problem2
1: 0.000005+0.000000j, 2: -0.007358+0.000000j, 3: -0.004491+-0.000000j, 4: 0.005555+-0.000000j, 5: 0.008972+0.000000j
6: -0.000008+0.000000j, 7: -0.010013+-0.000000j, 8: -0.007089+0.000000j, 9: 0.005596+0.000000j, 10: 0.010607+0.000000j
11: 0.000015+0.000000j, 12: -0.011789+0.000000j, 13: -0.006200+-0.000000j, 14: 0.012537+-0.000000j, 15: 0.019155+0.000000j
16: -0.000040+0.000000j, 17: -0.025813+0.000000j, 18: -0.028660+-0.000000j, 19: -0.011063+0.000000j, 20: -0.012305+-0.000000j
21: -0.060030+-0.000000j, 22: -0.123361+0.000000j, 23: -0.131975+0.000000j, 24: -0.051903+0.000000j, 25: 0.064813+-0.000000j
26: 0.120120+0.000000j, 27: 0.064813+0.000000j, 28: -0.051903+-0.000000j, 29: -0.131975+-0.000000j, 30: -0.123361+-0.000000j
31: -0.060030+0.000000j, 32: -0.012305+0.000000j, 33: -0.011063+-0.000000j, 34: -0.028660+0.000000j, 35: -0.025813+-0.000000j
36: -0.000040+-0.000000j, 37: 0.019155+-0.000000j, 38: 0.012537+0.000000j, 39: -0.006200+0.000000j, 40: -0.011789+-0.000000j
41: 0.000015+-0.000000j, 42: 0.010607+-0.000000j, 43: 0.005596+-0.000000j, 44: -0.007089+-0.000000j, 45: -0.010013+0.000000j
46: -0.000008+-0.000000j, 47: 0.008972+-0.000000j, 48: 0.005555+0.000000j, 49: -0.004491+0.000000j, 50: -0.007358+-0.000000j
51: 0.000005+-0.000000j
```

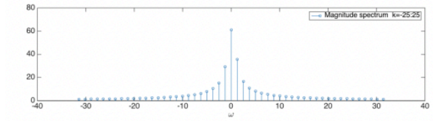
Code:

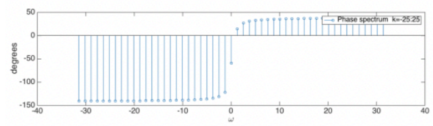
```
function y = u(t) % define step function
   y = (t >= 0);
end
t = linspace(-5, 5, 1000); % define range
x = 0.6*(u(t+2)-(cos(pi*t)+1).*(u(t+1)-u(t-1))-u(t-2)); % define x
fourier_series_exp(x, t, 51, 1); % call function
```

3. So far all the signals we have handled in this course are real signals. However, we can also use complex numbers to represent signals (complex signals). Let's consider a single period of a periodic signal $z(t) = t^3 - j2\pi t^2$, $0 < t \le 5$. Calculate 51 Fourier Series coefficients (C_k) for this signal and reconstruct the time domain signal $\hat{z}(t)$ using these Fourier Series coefficients. Plot the spectrum ($|C_k|$ vs ω and $\angle C_k$ vs ω) and the real and imaginary part of $z(t)$ and $\hat{z}(t)$ for an interval of $0 \le t \le 10$. You can modify the Matlab file '`fs_numerical.m`' which was used during the lab for solving this problem. Following are the sample plots for your reference.



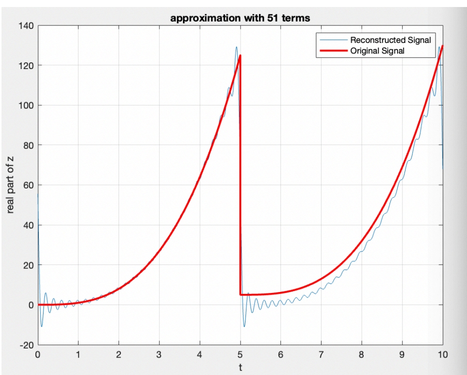


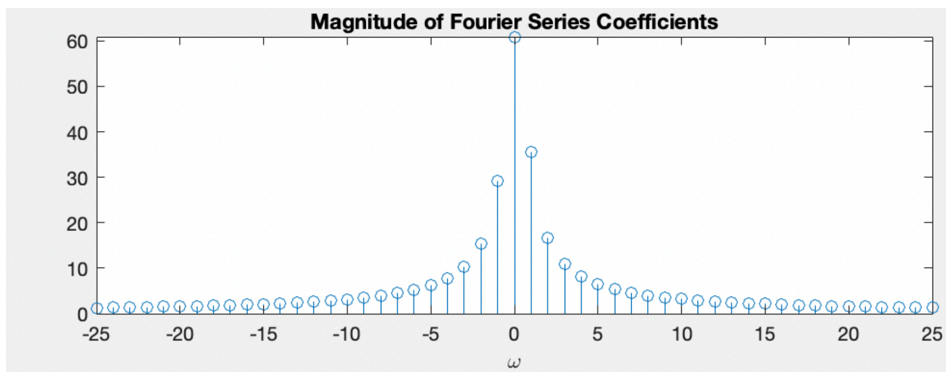


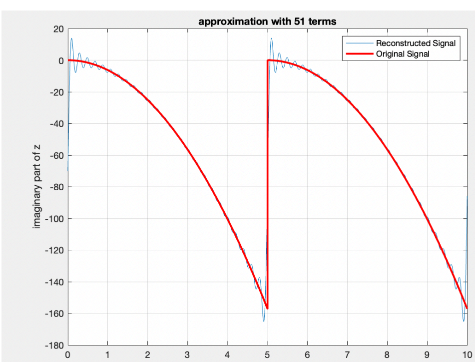


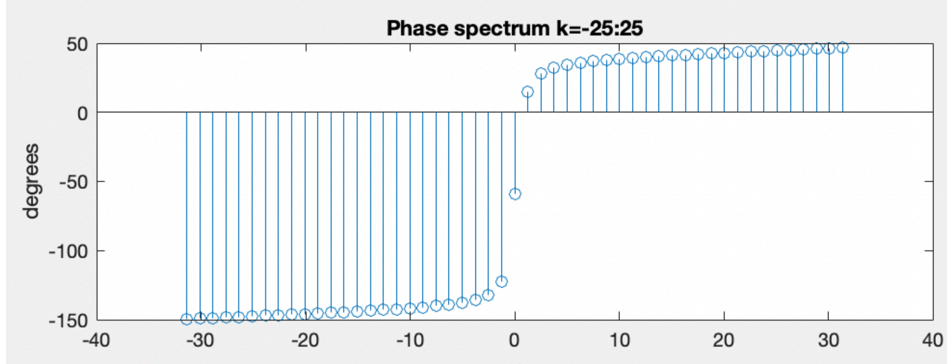
Summary:
We define the function z and time domains and make modifications to our function from #1 on top of the previous from #2. We call the function.

Results:










```
>> lab4problem3
1: -1.096799+-0.643068j, 2: -1.137882+-0.677763j, 3: -1.182422+-0.715493j, 4: -1.230880+-0.756679j, 5: -1.283806+-0.801822j
6: -1.341857+-0.851526j, 7: -1.405825+-0.906522j, 8: -1.476675+-0.967711j, 9: -1.555593+-1.036206j, 10: -1.644053+-1.113405j
11: -1.743912+-1.201089j, 12: -1.857543+-1.301558j, 13: -1.988022+-1.417836j, 14: -2.139417+-1.553982j, 15: -2.317217+-1.715574j
16: -2.529011+-1.910485j, 17: -2.785595+-2.150202j, 18: -3.102866+-2.452160j, 19: -3.505211+-2.844140j, 20: -4.032020+-3.373242j
21: -4.751165+-4.126057j, 22: -5.789889+-5.280493j, 23: -7.414841+-7.266463j, 24: -10.267184+-11.433302j, 25: -15.647240+-24.737519j
26: 31.218781+-52.307544j, 27: 34.451759+9.104565j, 28: 14.780588+7.760868j, 29: 9.281754+5.808926j, 30: 6.730542+4.598154j
31: 5.263107+3.802506j, 32: 4.311096+3.244569j, 33: 3.643893+2.832953j, 34: 3.150442+2.517205j, 35: 2.770727+2.267491j
36: 2.469493+2.065122j, 37: 2.224681+1.897824j, 38: 2.021785+1.757212j, 39: 1.850876+1.637370j, 40: 1.704932+1.534008j
41: 1.578842+1.443936j, 42: 1.468804+1.364739j, 43: 1.371926+1.294551j, 44: 1.285972+1.231908j, 45: 1.209185+1.175650j
46: 1.140166+1.124840j, 47: 1.077787+1.078716j, 48: 1.021129+1.036653j, 49: 0.969432+0.998131j, 50: 0.922069+0.962715j
51: 0.878512+0.930040j
```

Code:

```
clf; % clear figure
```

```matlab
function [Ck]=fourier_series_exp(x,t,Nk,p)
   dT=t(2)-t(1); % calc dT
   T= dT*length(t); % calc T
   w0=2*pi/T;   % calc w0
   if nargin <2 % handle input / default values
       error('Not enough input argument!')
   elseif nargin == 2
       Nk=21;
       p=0;
   elseif nargin ==3
       p=0;
   end
   k=-floor(Nk/2):floor(Nk/2); % calc k
   Ck = zeros(1, length(k)); % Preallocate Ck
   for i = 1:length(k)
       Ck(i) = trapz(t, x .* exp(-1j * k(i) * w0 * t)) * (1/T); % calc fourier
coefficients
   end
   for i = 1:length(k)
           fprintf("%d: %.6f+%fj", i, real(Ck(i)), imag(Ck(i))); % print
formatted
           if mod(i, 5) == 0
               fprintf("\n");
           elseif i == length(k)
               fprintf("\n");
           else
               fprintf(", ");
           end
   end
   if p==1
       subplot(2,1,1);
       stem(k,abs(Ck));         % plot abs(Ck) vs k and angle(Ck) vs k
       xlabel('\omega');
       title('Magnitude of Fourier Series Coefficients');
       subplot(2,1,2);
       w0k = w0*k; % harmonic angular frequencies
       stem(w0k,angle(Ck)*180/pi); % plot phase
       xlabel('\omega');
       ylabel('degrees');
       title('Phase spectrum k=-25:25');
       figure; % plot 3 cycles of the signal 'x' and the reconstructed signal
       t_extended = linspace(min(t),2*max(t),2*length(t)); % 2 periods
       x_reconstructed = zeros(1,length(t_extended)); % initialize
reconstruction
       for i = 1:length(k)
           x_reconstructed = x_reconstructed +
Ck(i)*exp(1j*k(i)*w0*t_extended); % reconstruct signal
       end
       plot(t_extended,real(x_reconstructed)); % plot reconstructed
```

```
        hold on;
        plot([t, t+5], [real(x), real(x+5)],'r', 'LineWidth', 2); % plot real
signal
        xlabel('t');
        ylabel('real part of z');
        title('approximation with 51 terms');
        legend('Reconstructed Signal', 'Original Signal');
        grid on;
        figure;
        plot(t_extended,imag(x_reconstructed)); % plot reconstructed
        hold on;
        plot([t, t+5], [imag(x), imag(x+5)],'r', 'LineWidth', 2); % plot imag
signal
        xlabel('t');
        ylabel('imaginary part of z');
        title('approximation with 51 terms');
        legend('Reconstructed Signal', 'Original Signal');
        grid on;
   end
end
t = linspace(0, 5, 1000); % define t
z = t.^3-1j*2*pi*t.^2; % define z
fourier_series_exp(z, t, 51, 1); % call func
```