Fourier Series Lab I

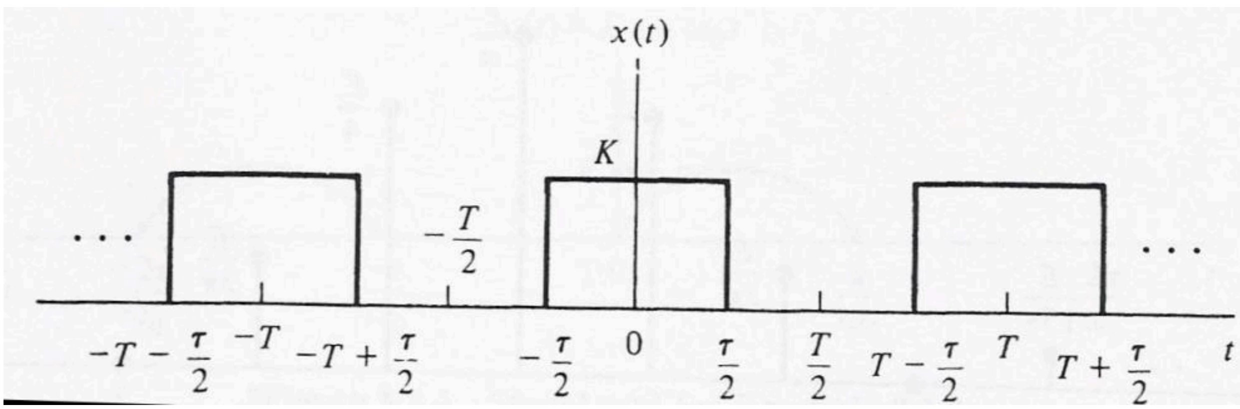
Overview

The objectives of this lab are to learn how to set up the integral to learn the Fourier Coefficients for a periodic signal, investigate the Gibbs Phenomenon that occurs at discontinuities of the signal. The lab uses the Matlab Symbolic Toolbox to evaluate the continuous time integrals from the Fourier coefficient computation.

Pre-Lab

a. Read through the provided code in section I on computing Fourier Series Coefficients and make sure you understand the script.

b. Compute the Fourier Series Coefficients for the square wave given below analytically.



I. Computing Fourier Series Coefficients in Matlab

The following code calculates the Fourier series of the following signal with Matlab symbolic calculation (Please note that the symbolic function for the unit step function u(t) in Matlab is *heaviside*.)

```
function [X, w] = FourierSeries(x, T0, k_vec)
%
% symbolically calculates the Fourier Series, and returns the
% numerical results
%
% x: the time domain signal within one period;
% it must have definition over [0, T0]
% it must be a symbolic function of t
% T0: the period of the signal
% k_vec: the range of Harmonics to be calculated
%
% Outputs
% X: vector of Fourier coefficients evaluated at k_vec*w0 frequency,
%    Note that the numbers go from -N, -N+1, ...,0, 1,2   , N for the
%    indices
%  w: vector of evaluation frequencies
```

```
syms t
for mm = 1:length(k_vec)
    k = k_vec(mm);
    % angular frequency
    w(mm) = k*2*pi/T0;
    % Fourier series coefficients
    X1(mm) = int(x*exp(-j*w(mm)*t), t, 0, T0)/T0;
    % change the symbolic value to numerical value
    X(mm) = subs(X1(mm));
end
```

The code has two pieces a function that computes the Fourier Series coefficients and a main program where the function x(t) is defined, that calls it. The main program also plots the coefficients. T0 =5 and tau = 1;

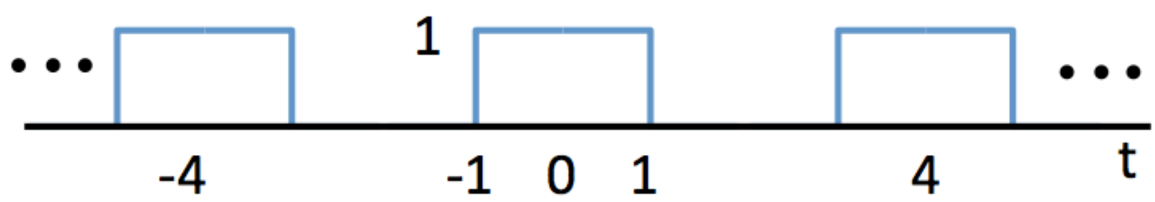
```
% FSmain.m
% Calculates the Fourier series through symbolic cacluations
clear all
syms t;

% Time Signal parameters
tau = 1; % length of signal
T0 = 5; % fundamental period
tshift = 1; % time shift from signal centered at 0
amp = 1; % amplitude of signal
baseline = -0.5; % DC bias

% !!!IMPORTANT!!!: the signal definition must cover [0 to T0]
% the signal is defined over [-T0, 2T0], which covers [0, T0]
N = 7; % number of components +/- to compute
k_vec = [-N:N];
xt = amp*(heaviside(t+tau/2-tshift)-heaviside(t-tau/2-tshift) + heaviside(t-
(T0-tau/2)-tshift) ...
-heaviside(t-(T0+tau/2)-tshift))+ heaviside(t+T0-tshift)*baseline;

% Compute FS coefficients
[X, w] = FourierSeries(xt, T0, k_vec);

% plot the results from Matlab calculation
figure();subplot(211);
% plot magnitude and phase separately
stem(w,abs(X), 'o-');
xlabel('Frequency (rad/sec)');ylabel('Magnitude');
hold on;subplot(212);
stem(w,angle(X), 'o-');
xlabel('Frequency (rad/sec)');ylabel('Phase');
```

II. Lab Assignments

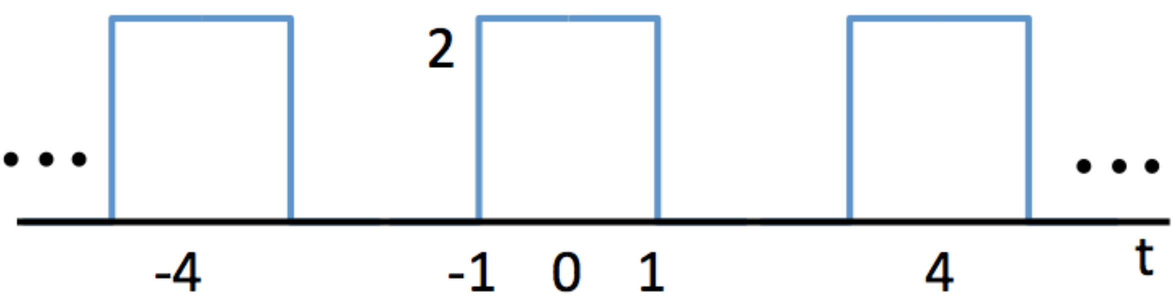
A. Symbolic Fourier Series Calculation

Use Matlab to perform symbolic Fourier series calculation of the following signals for the first 10 harmonics, k goes from -10 to 10. Plot both the amplitude and phase of the harmonics. Save the coefficients for each case as a vector. In each case, identify which Fourier Series property was used to change the sequence. Show which Fourier Series coefficients changed in magnitude and phase and why by providing a plot of the frequency spectrum. Also, give your signal parameters for each case.

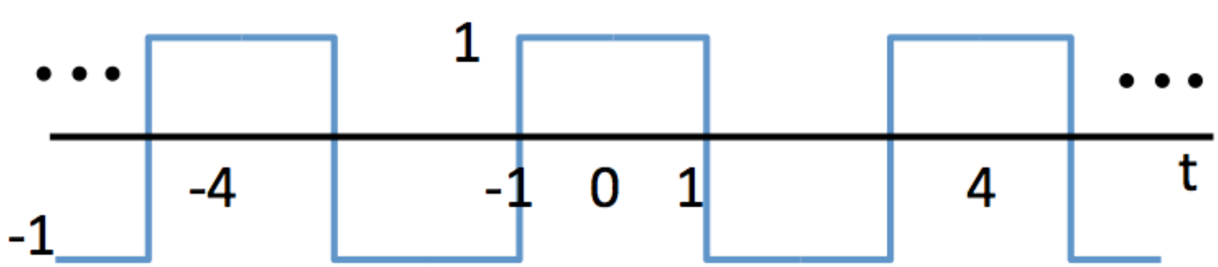
1. The first signal is a 50% duty cycle square wave. You will need to change the signal parameters in the mainFS script to get the right time signal.



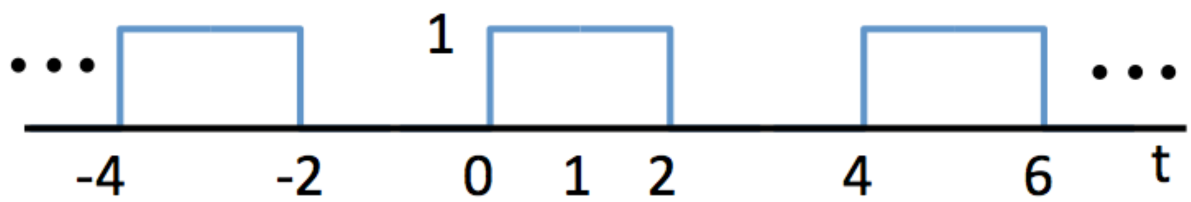
2. The second signal has been scaled by a factor of two. The fundamental period is the same. You will need to change the signal parameters in the mainFS script to get the right time signal.



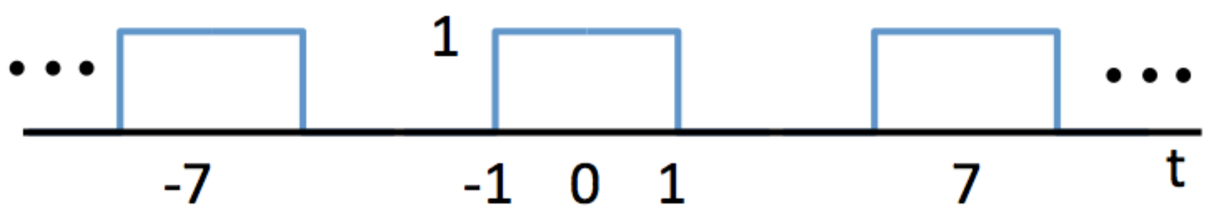
3. The third signal has been scaled by a factor of two and shifted to be centered at zero in amplitude. The fundamental period is the same. You will need to change the signal parameters in the mainFS script to get the right time signal.



4. The fourth signal is delayed in time by one unit. You will need to change the signal parameters in the mainFS script to get the right time signal.



5. The fifth signal has a different fundamental frequency. You will need to change the signal parameters in the mainFS script to get the right time signal.



B. Gibbs Phenomenon

When the Fourier Series is truncated (not all harmonics are included), a phenomenon called the Gibbs oscillation occurs.

From Wikipedia (https://en.wikipedia.org/wiki/Gibbs_phenomenon): "In mathematics, the **Gibbs phenomenon,** discovered by Henry Wilbraham (1848) and rediscovered by J. Willard Gibbs (1899), is the peculiar manner in which the Fourier series of a piecewise continuously differentiable periodic function behaves at a jump discontinuity. The *n*th partial sum of the Fourier series has large oscillations near the jump, which might increase the maximum of the partial sum above that of the function itself. The overshoot does not die out as *n* increases, but approaches a finite limit. This sort of behavior was also observed by experimental physicists, but was believed to be due to imperfections in the measuring apparatuses. This is one cause of **ringing artifacts** in signal processing."

Let $\{c_k\}$ be the Fourier Series coefficients of x(t). Define

$$x_N(t) = \sum_{k=-N}^{N} c_k e^{jk\omega_0 t}$$

This is the synthesis equation for Fourier Series with a finite number of sinusoids.

```
function [xt,dt] = FSsynthesis_Square(N, T0, a0, Range);

% N number of components to use
```

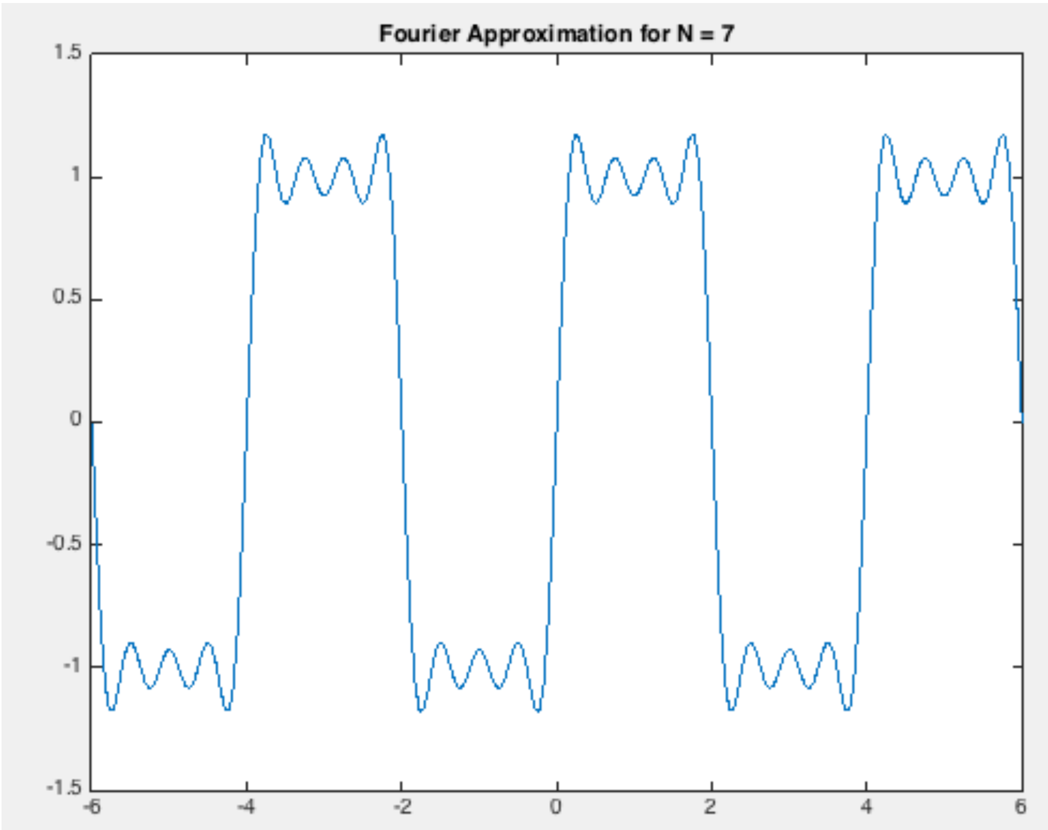
```
% T0 fundamental period in seconds of the square wave
% a0 DC component
% Range - plotting range for signal, 2x1 vector

n_vec = [1:N];
Omega0 = 2*pi/T0;f0 = 1/T0;

% Compute the Fourier series coefficients for a square wave
a_k = zeros(size(n_vec));

% odd indexed components only have non-zero values
a_k(1:2:end) = 1./(j*n_vec(1:2:end)*Omega0);

dt = 1/(N*f0*10);
t = [Range(1):dt:Range(2)];
xt = zeros(size(t));
for m = 1:length(n_vec)
    xt = xt + a_k(m)*exp(j*n_vec(m)*Omega0*t)+conj(a_k(m))*exp(-
j*n_vec(m)*Omega0*t);
end
xt = xt + a0*ones(size(t));

figure();
plot(t,xt)
title(['Fourier Approximation for N = ', num2str(N)]);
xlabel([‘time’]); ylabel([‘x(t)’]);
```

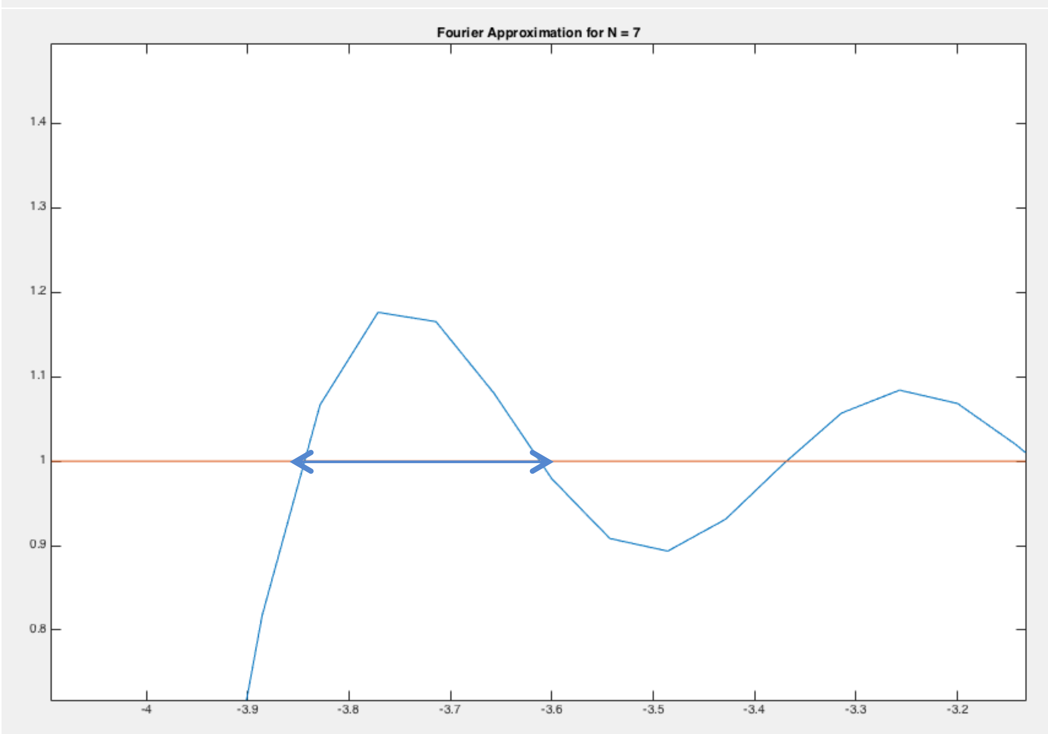
The code above generates the signal for a square wave with a period of 4 with a single period taking the form:

$$x(t)=\begin{cases}1 & 0\leq t<2\\ -1 & 2\leq t<4\end{cases}$$

for the input parameters T0 = 4; a0 = 0.

Investigate the Gibbs Phenomenon for N = 5,13,23 and 99. Plot $x_N(t)$ as a function of time. Use the zoom tool to measure the height and width of the overshoot for each case. Measure the width of the overshoot as shown below. Add a table to your report with the values and figures. Comment on the height of the overshoot and the width of the overshoot as N increases.


Fourier Approximation for N = 7
1.5
1
0.5
0
-0.5
-1
-1.5
-6
-4
-2
0
2
4
6


Fourier Approximation for N = 7
1.4
1.3
1.2
1.1
1
0.9
0.8
-4
-3.9
-3.8
-3.7
-3.6
-3.5
-3.4
-3.3
-3.2