Introduction: This lab teaches about the setting up of integral to learn about the Fourier Coefficients for a periodic signal in MATLAB. It also teaches on how to investigate the Gibbs Phenomenon.

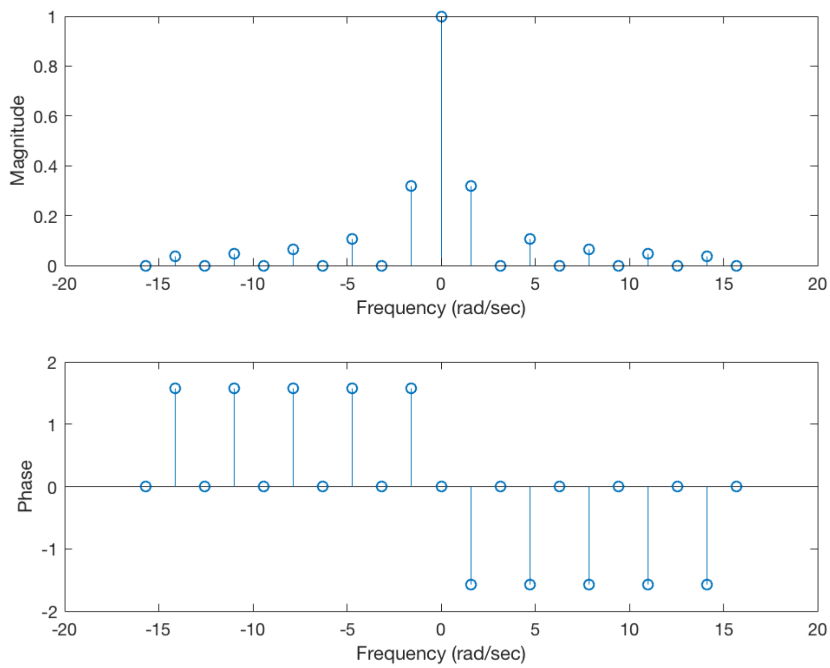
II. Lab Assignment

A.

1.

```
% FSmain.m
% Calculates the Fourier series through symbolic cacluations
clear all
syms t;
% Time Signal parameters
tau = 2; % length of signal
T0 = 4; % 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 = 10; % 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');
```

For this signal parameter T0=4, amp=1, N=10.

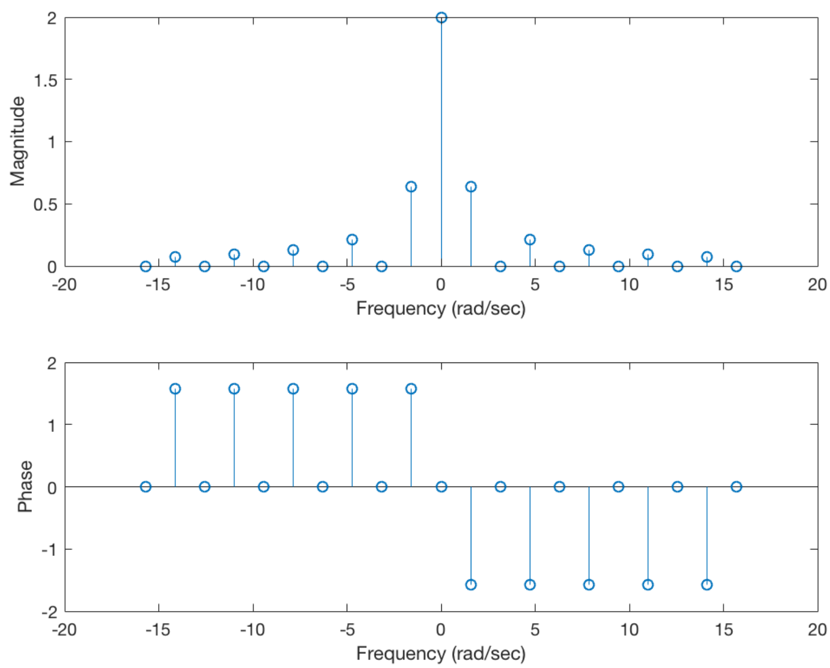


2.

```
% FSmain.m
% Calculates the Fourier series through symbolic cacluations
clear all
syms t;
% Time Signal parameters
tau = 2; % length of signal
T0 = 4; % fundamental period
tshift = 1; % time shift from signal centered at 0
amp = 2; % amplitude of signal
baseline = 1; % DC bias
% !!!IMPORTANT!!!: the signal definition must cover [0 to T0]
% the signal is defined over [-T0, 2T0], which covers [0, T0]
N = 10; % 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');
```

For this signal parameter T0=4, amp=2, baseline=1, N=10.

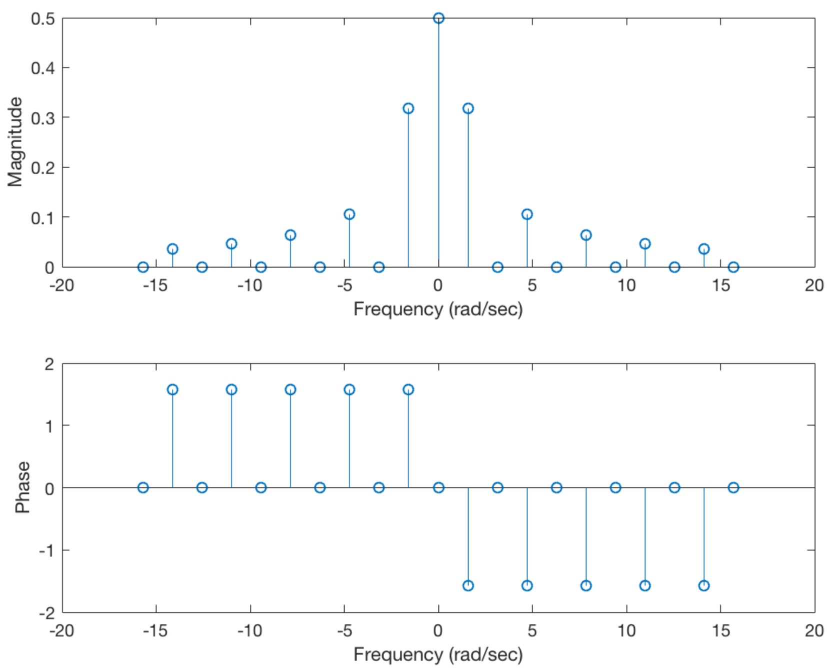


3.

```
% FSmain.m
% Calculates the Fourier series through symbolic cacluations
clear all
syms t;
% Time Signal parameters
tau = 2; % length of signal
T0 = 4; % fundamental period
tshift = 1; % time shift from signal centered at 0
amp = 1; % amplitude of signal
baseline = 0; % DC bias
% !!!IMPORTANT!!!: the signal definition must cover [0 to T0]
% the signal is defined over [-T0, 2T0], which covers [0, T0]
N = 10; % 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');
```

For this signal parameter T0=4, amp=1, baseline=0, N=10.

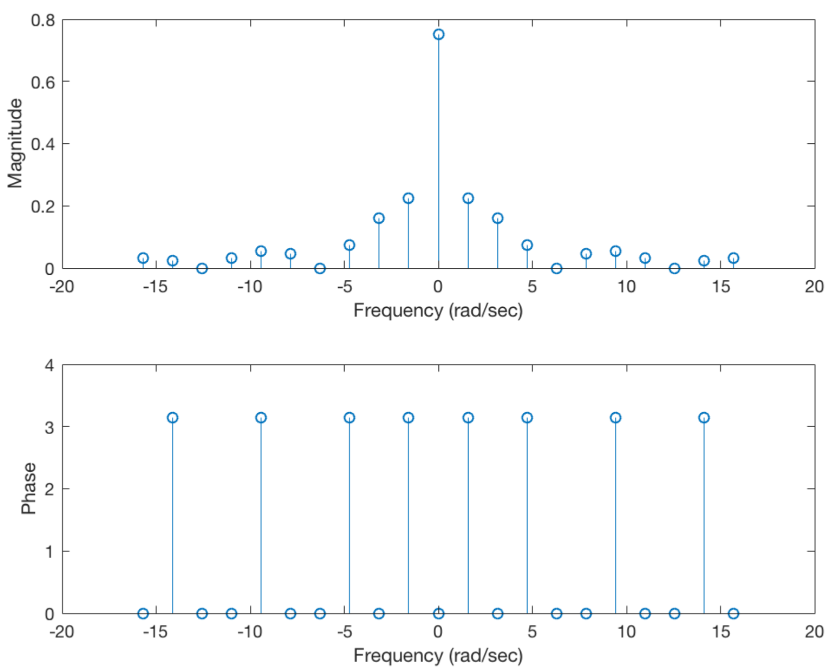


4.

```
% FSmain.m
% Calculates the Fourier series through symbolic cacluations
clear all
syms t;
% Time Signal parameters
tau = 1; % length of signal
T0 = 4; % fundamental period
tshift = 2; % 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 = 10; % 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');
```

For this signal parameter T0=4,tshift=2, amp=1, N=10.

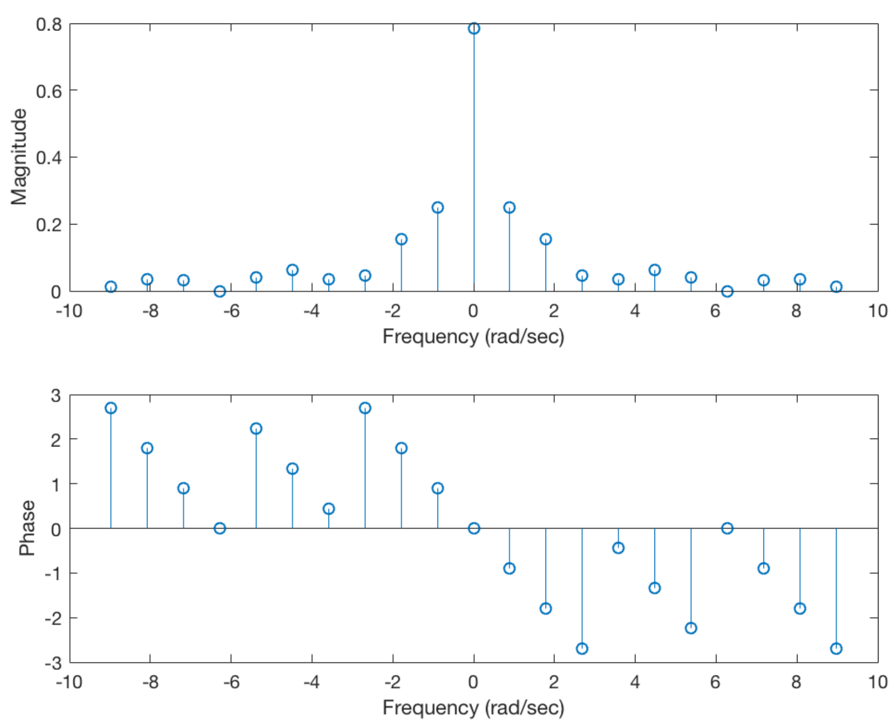


5.

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

```
% Time Signal parameters
tau = 2; % length of signal
T0 = 7; % 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 = 10; % 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');
```

For this signal parameter T0=7, tshift=1, amp=1, N=10.

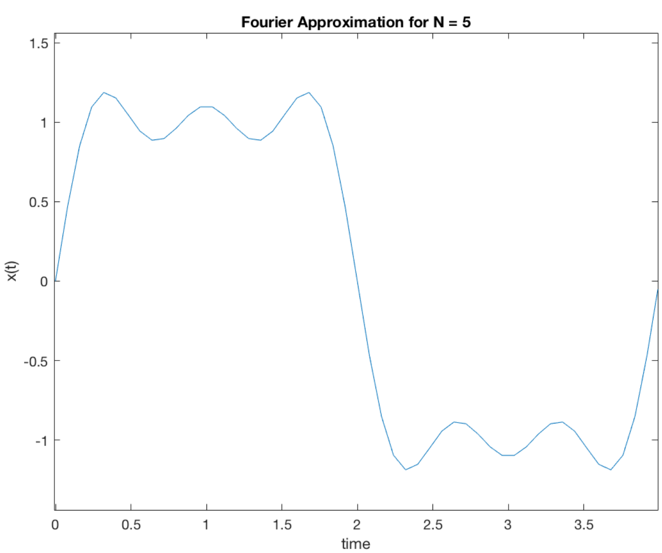


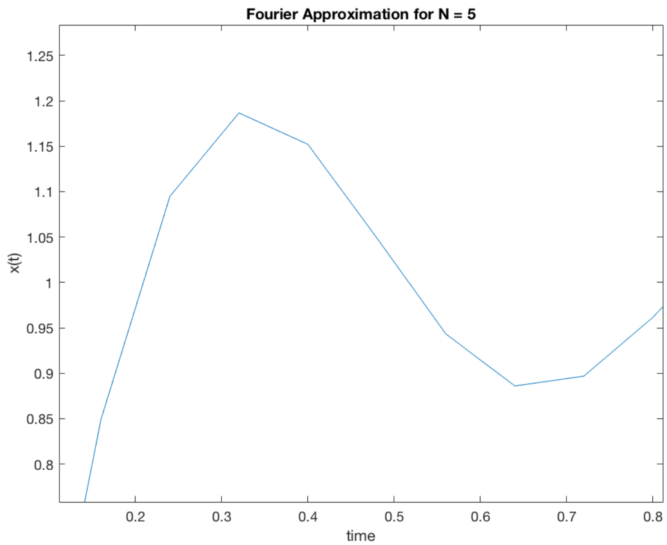
B.

For N=5:

```
function [xt,dt] = FSsynthesis_Square(N, T0, a0, Range);
N=5;% N number of components to use

T0=4;% T0 fundamental period in seconds of the square wave
a0=0;% 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 = [0:dt:4];
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)']);
```

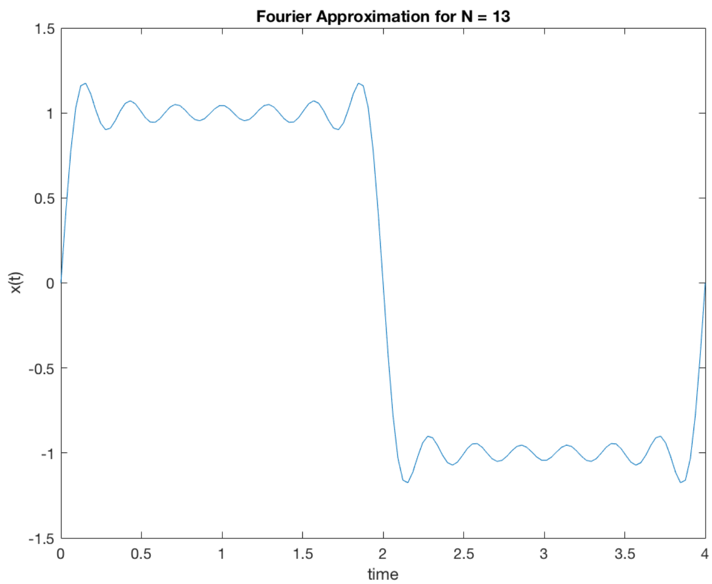


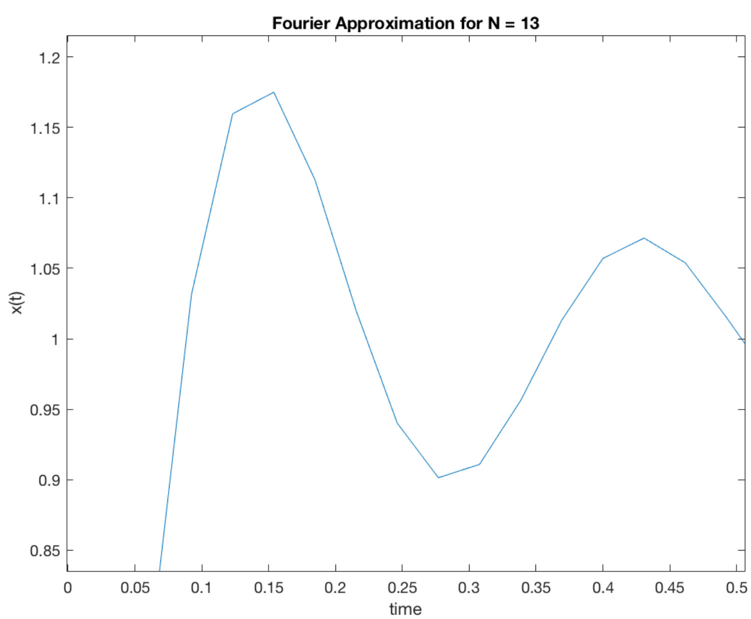


For N = 13:

```
function [xt,dt] = FSsynthesis_Square(N, T0, a0, Range);
N=13;% N number of components to use

T0=4;% T0 fundamental period in seconds of the square wave
a0=0;% 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 = [0:dt:4];
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)']);
```



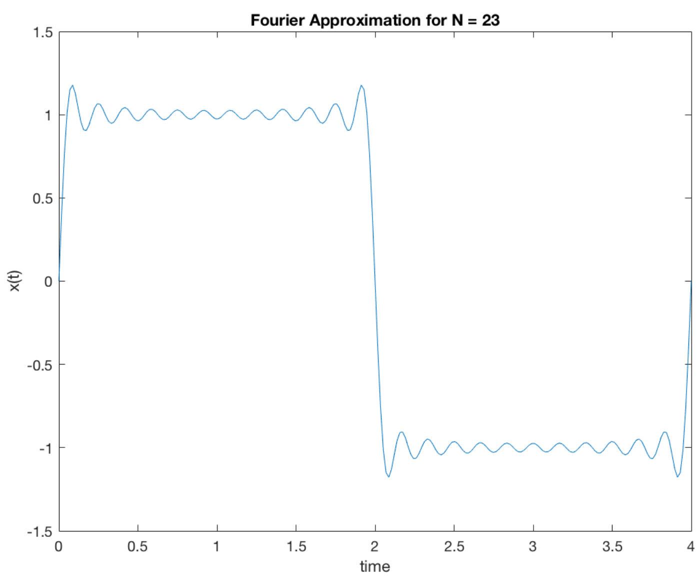


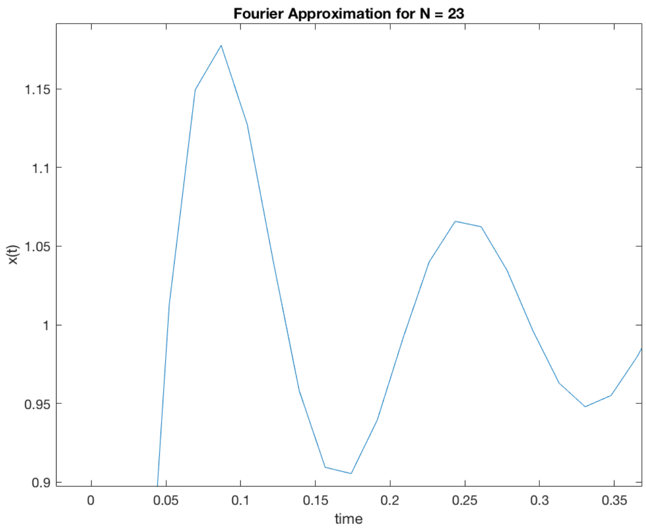
For N = 23:

```
function [xt,dt] = FSsynthesis_Square(N, T0, a0, Range);
N=23;% N number of components to use

T0=4;% T0 fundamental period in seconds of the square wave
```

```
a0=0;% 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 = [0:dt:4];
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)']);
```



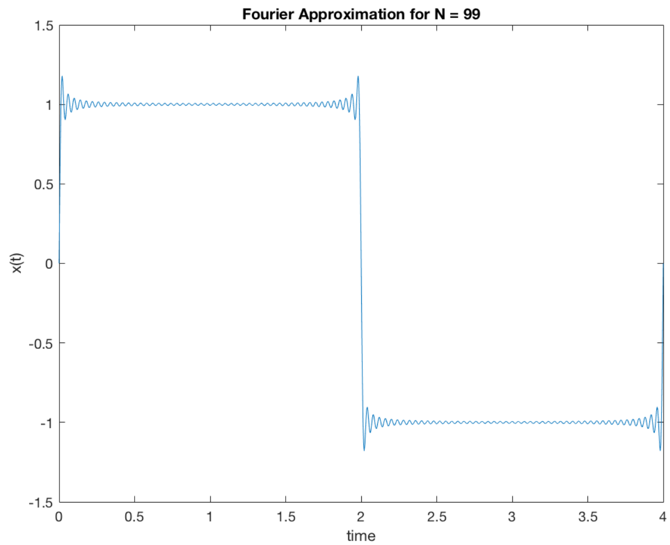


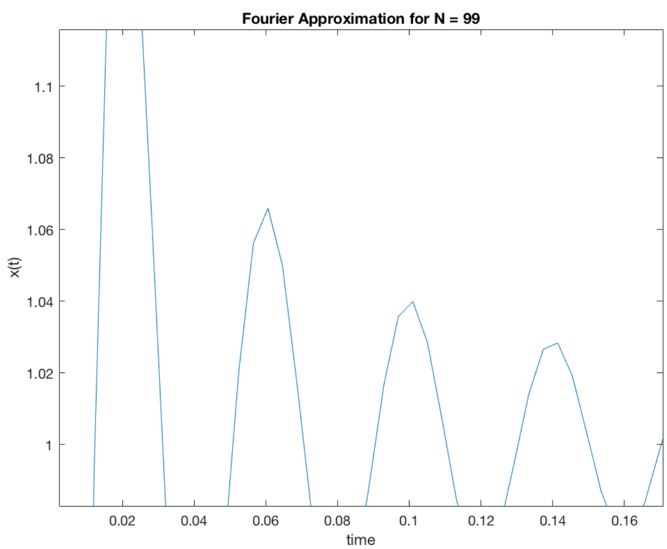
For N=99:

```
function [xt,dt] = FSsynthesis_Square(N, T0, a0, Range);
N=99;% N number of components to use

T0=4;% T0 fundamental period in seconds of the square wave
a0=0;% 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 = [0:dt:4];
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)']);
```

Agarwal Aashwatth LAB REPORT #3





Value of N	Width of overshoot
N=5	0.30
N=13	0.16

N=23	0.10
N=99	0.02

So, as N increases, the height of the overshoot tends to almost remains the same, but the width of overshoot keeps on decreasing.