Code for plotting square function and its Fourier Series Expansion

```
T = 0.001; % period of signal
A = 1; % amplitude of signal

dt = T/100;     % step in time domain
t  = -2*T:dt:2*T;% define the time axis - up to 4 time periods
%
% define the function s(t) - one period is used
s = A*(1/2 + 1/2*square(2*pi*(t+T/4)./T));


%
% Fourier Series expansion is given by:
%
%                infinity
%  s(t) = a_0/2 + sum [ a_k * cos(2*pi*k*f*t) + b_k * sin(2*pi*k*f*t)]
%                 k=1
%


aa_0            = A;
max_k           = 100;
k               = 1:max_k;
aa_k            = 2*A./(pi*k) .* (-1).^((k-1)/2);
aa_k(2:2:max_k) = 0;

bb_k            = zeros(size(k));

f  = 1/T;
ss = A/2 *ones(size(t));
for i=1:6
   ss = ss + aa_k(i)*cos(2*pi*i*f*t) + bb_k(i)*sin(2*pi*f*t);
end

figure(1);
subplot(2,1,1);
plot(t, s);
title('s(t) - square function');
xlabel('time (t) - seconds');
ylabel('Amplitude - volts');
axis([-2*T 2*T -0.2 1.2]);
grid

subplot(2,1,2);
plot(t, ss);
title('s(t) - using Fourier Series expansion (6 terms)');
xlabel('time (t) - seconds');
ylabel('Amplitude - volts');
grid
```

Code for plotting function and its Spectrum Function using FFT routine

```
clear all
%
% N defines the numbe of samples in the time domain - and the number of
% FFT points
% Since N has be even, N is made a multiple of 2
M = 10;  % The higher the value of M (and consequently N) the better
N = 2^M;% the approximation of the FFT to the Fourier Series expansion

T = 0.001; % period of signal
A = 1; % amplitude of signal

n  = 0:1:N-1;
dt = T/N; % step in time domain
t  = dt*n; % define the time axis
%
% define the function s(t) - one period is used
s = A*(1/2 + 1/2*square(2*pi*(t+T/4)./T));

%
% fft definition includes averaging over N

S = fft(s)/N;

figure(1);
subplot(2,1,1);
plot(t, s);
title('s(t) - one period)');
xlabel('t - time');
ylabel('s(t)');
axis([0 T -0.2 1.2]);
grid
subplot(2,1,2);
f = n./T;
stem(f(1:N/2), abs(T*S(1:N/2)),'');
title('Frequency Spectrum for s(t)');
xlabel('frequency (Hz)');
ylabel('Magnitude of S(f)');
axis([0 20./T 0 1.2*max(abs(T*S))]);
grid
```