```
%Christopher Morales
```

Problem 1

Implementing a Fourier Series Going to use 24 components

```
% The number of components in the spectrum
N = 1024;

% Sample frequency
fs = 1024;

% Number of component oging to iterate for the fourier series
Nfft = 24;

% Creating the sawtooth data
x1 = 1:N;
x2 = sawtooth(x1);


% Getting the Fast Fourier Transform
Xf = fft(x2,Nfft);

% Magnitude
Xfmag = abs(Xf);

% Phase Shift Angle
Xfphase = angle(Xf);

% Frequency Vector
f = [0:(Nfft-1)] * fs/Nfft;

% Reconstructing the time vector
t = (1:Nfft)/fs;

% Reconstructing the x(t) and need to create and empty vector
L = length(Xf);
y = zeros(1,L);

% Use the fourier transform to replicate the following signal
% If you increase L then the signal will look like a sawtooth
for i = 1 : Nfft
    y = y + Xfmag(i) * cos( 2 * pi * f(i) * t + Xfphase(i));
end
```

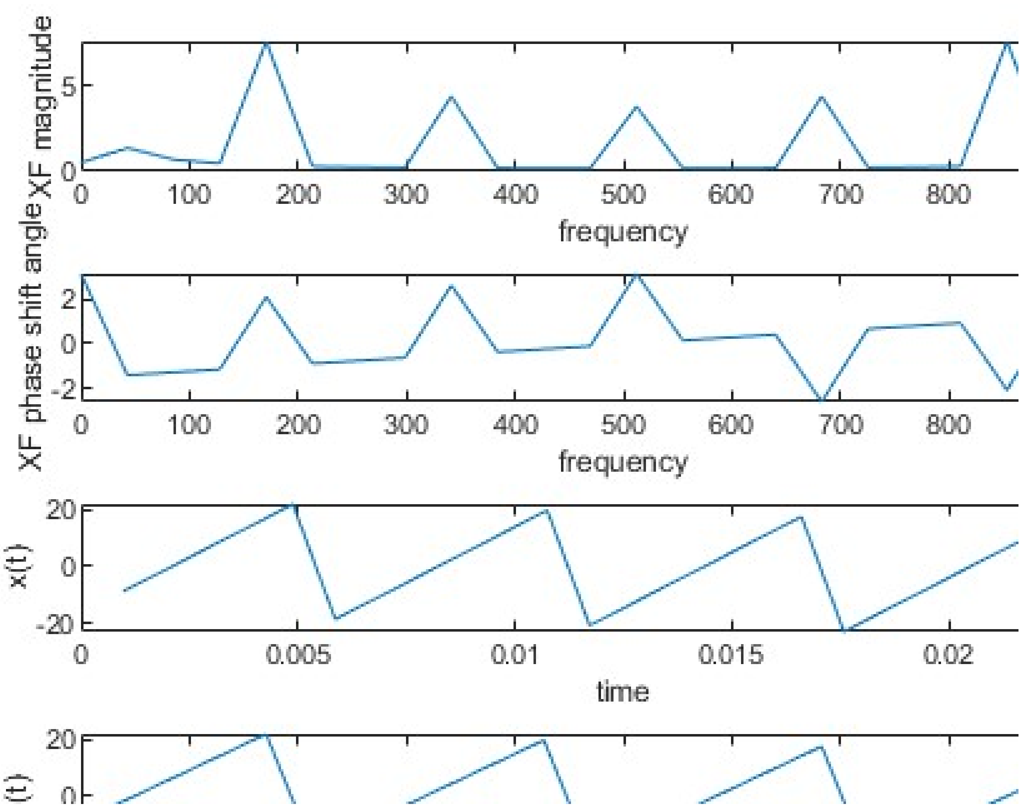
```
% Creating a figure
figure(1);

% Creating a plot for Hz vs Xf magnitude
subplot(4,1,1);
plot(f, Xfmag);
xlabel('frequency');
ylabel('XF magnitude');

subplot(4,1,2);
plot(f, Xfphase);
xlabel('frequency');
ylabel('XF phase shift angle');

subplot(4,1,3);
plot(t, y);
xlabel('time');
ylabel('x(t)');

subplot(4,1,4);
plot(f, y);
xlabel('Frequency');
ylabel('x(t)');
```



Problem 2

```
load('ECG_1min.mat');
whos

% Contains 1 min of ECG data
% Get Fourier Transform, plot magnitude and phase
% Get the average beats per min (not seconds)
% No DC offset

% Sample Frequency
fs = 250;

% Fast Fourier Transform
Xf = fft(ecg);

% FFT magnitude
Xfmag = abs(Xf);

% FFT phase shift angle
XFphase = angle(Xf);

% Domain in degrees
omega = [0:(length(ecg)-1)]/ length(ecg) * (2 * pi);

% Frequency
f = [0:(length(ecg)-1)]/ length(ecg) * fs;

% Creating a figure
figure(1);

% Creating a sub plot
subplot(2,1,1);

% Plot Frequency vs Magnitude
plot(f, Xfmag);
xlabel('Hz');
ylabel('Xf magnitude');
title('Christopher Morales (Frequency vs Magnitude) Full Spectrum');

% Creating a subplot
subplot(2,1,2);

% Plot Frequency vs Phase angle shift
plot(f, XFphase);
xlabel('Hz');
ylabel('Xf Phase Shift Angle');
title('Christopher Morales (Frequency vs Phase Shift Angle) Full Spectrum');

% Getting the Max array and the Index of the max value
[m, imax] = max(Xfmag);

% max from the full spectrum
```

```
fmax = f(imax);

% f = 1/T, divide the fmax to get the interbeat interval
interbeat_interval = 1/fmax;

% Muliply 60 seconds to get min then we get BPM
Heartbeats_per_minute = 60 * interbeat_interval;

fprintf('imax = %.4f \n',imax);
fprintf('Fmax is: %.4f \n', fmax);
fprintf('beats per second is %.4f \n',interbeat_interval );
fprintf('Heart beats per mins is: %.4f \n',Heartbeats_per_minute);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Know do the same but up to 20 hz and not the full spectrum
% Will repeat the previous steps used but limited to the spectrum
% alpha will be the new frequency domain that is limited to 20 samples
% As well with the updated magnitude and phse angle shift
XFsample = fft(ecg,length(ecg));
Xfmagsample = abs(XFsample);
XFphasesample = angle(XFsample);

% Degrees
alpha = [ 0 : (20-1) ] / 20 * (2 * pi);

% Frequency
f2 = [0:(length(ecg)-1)]/ length(ecg) * 20;

% Creating a second figure, up to 20 Hz
figure(2);
hold ON;

% Plotting Frequency vs magnitude
subplot(2,1,1);
plot(f2, Xfmagsample);
xlabel('Frequency');
ylabel('Xf magnitude');
title('Christopher Morales (Frequency vs Magnitude)');

% Plotting Frequency vs phase shift angle
hold ON;
subplot(2,1,2);
plot(f2, XFphasesample);
xlabel('Frequency');
ylabel('Xf Phase Shift Angle');
title('Christopher Morales (Frequency vs Phase Shift Angle)');

% Everything from here on is not important but curious if the result will
% be different form 250hz (like drastic change or minimum)
% Getting the max array and the max index from the array
[m2, imax2] = max(XFsample);
```

```
% max from the 20 spectrum
f2max = f2(imax2);
```

```
  Name                Size                Bytes  Class     Attributes

  Discrete_Nfft       1x3                    24  double
  Discrete_fs         1x1                     8  double
  L                   1x1                     8  double
  N                   1x1                     8  double
  Nfft                1x1                     8  double
  Xf                  1x24                  384  double    complex
  Xfmag               1x24                  192  double
  Xfphase             1x24                  192  double
  ecg                 1x15000            120000  double
  f                   1x24                  192  double
  fs                  1x1                     8  double
  i                   1x1                     8  double
  j                   1x1                     8  double
  t                   1x24                  192  double
  x                   1x32                  256  double
  x1                  1x1024               8192  double
  x2                  1x1024               8192  double
  y                   1x24                  192  double
  ysave               1x3                    24  double

imax = 77.0000
Fmax is: 1.2667
beats per second is 0.7895
Heart beats per mins is: 47.3684
```

Problem 3

Generate a 512 point waveform Consist of two sinsusiod Get the Fourier Transform Plot both Magnitude and Waveform

```
% 512 waveform points
N = 512;

% Frequency Sample
fs = 1000;

% Frequency for there repective function
f1 = [200 400];
f2 = [200 400 900];

% Contructing the time vector
t = (1:N)/ (fs) * N;

% Constructing the signal
% Going to use a for loop to implement every data plot signal
% X1 will be to the respect of one frequency
% while x2 will be to the respect of another frequency
```

```
for i = 1 : N
    x1 = sin( 2 * pi * f1(1) * t) + sin(2 * pi * f2(2) * t);
end

for j = 1 : N
    x2 = sin( 2 * pi * f1(1) * t) + sin(2 * pi * f2(3) * t);
end

% Fourier Transform
Xf1 = fft(x1);
Xf2 = fft(x2);

% Magnitude
Xf1mag = abs(Xf1)/2;
Xf2mag = abs(Xf2)/2;

% Creating the frequency vector
f = (1:N) * fs / N;

% Creating a figure
figure(1);

% Creating a plot and plotting the signal
plot(f, Xf1mag, 'g', f, Xf2mag, '--r' );
xlabel('Frequency');
ylabel('Xf Magnitude');
title('Christopher Morales: Superimposed');
legend('200 Hz and 400 Hz', '200 Hz and 900 Hz');
```

Problem 4

```
load('x_fftlab-1.mat');
whos

% Frequency Sample
Discrete_fs = 3000;

% Discrete signal spectrum
Discrete_Nfft = [32 128 512];

% Fast Fourier Transform
Xf = fft(x);

% Magnitude
Xfmag = abs(Xf);

% Phase angle shift but in radians
Xfphase = unwrap(angle(Xf));

% We are going to create an empty array to save all the data from the for
% loop
ysave = [0 0 0];
```

```
% Creating a loop where it will iterate each dicrete nfft value
for i = 1 : 3

    % Reconstructing the x(t) and need to create and empty vector
    L = length(Discrete_Nfft(i));
    y = zeros(1,L);

    f = [0:(Discrete_Nfft(i)-1)] * Discrete_fs/Discrete_Nfft(i);

    t = (1:Discrete_Nfft(i))/Discrete_fs;

    for j = 1 : L
            y = y + Xfmag(j) * cos( 2 * pi * f(j) * t + Xfphase(j));
            ysave(i) = y;
    end

end

figure(1)

subplot(5,1,1);
plot(f, Xfmag );
xlabel('Frequency');
ylabel('magnitude');

subplot(5,1,2);
plot(f, Xfmag );
xlabel('Frequency');
ylabel('Phase Shift Angle');

subplot(5,1,3);
plot(f, ysave(1) );
xlabel('time');
ylabel('x(t)');

subplot(5,1,4);
plot(f, ysave(2) );
xlabel('time');
ylabel('x(t)');

subplot(5,1,5);
plot(f, ysave(3) );
xlabel('time');
ylabel('x(t)');
```

```
  Name                       Size            Bytes  Class     Attributes

  Discrete_Nfft              1x3                24  double
  Discrete_fs                1x1                 8  double
  Heartbeats_per_minute      1x1                 8  double
  L                          1x1                 8  double
  N                          1x1                 8  double
```

```
  Nfft                      1x1                    8  double
  XFphase                   1x15000           120000  double
  XFphasesample             1x15000           120000  double
  XFsample                  1x15000           240000  double    complex
  Xf                        1x15000           240000  double    complex
  Xf1                       1x512               8192  double    complex
  Xf1mag                    1x512               4096  double
  Xf2                       1x512               8192  double    complex
  Xf2mag                    1x512               4096  double
  Xfmag                     1x15000           120000  double
  Xfmagsample               1x15000           120000  double
  Xfphase                   1x24                 192  double
  alpha                     1x20                 160  double
  ecg                       1x15000           120000  double
  f                         1x512               4096  double
  f1                        1x2                   16  double
  f2                        1x3                   24  double
  f2max                     1x1                    8  double
  fmax                      1x1                    8  double
  fs                        1x1                    8  double
  i                         1x1                    8  double
  imax                      1x1                    8  double
  imax2                     1x1                    8  double
  interbeat_interval        1x1                    8  double
  j                         1x1                    8  double
  m                         1x1                    8  double
  m2                        1x1                   16  double    complex
  omega                     1x15000           120000  double
  t                         1x512               4096  double
  x                         1x32                 256  double
  x1                        1x512               4096  double
  x2                        1x512               4096  double
  y                         1x24                 192  double
  ysave                     1x3                   24  double
```

```
Unable to perform assignment because the left and right sides have a different number of
elements.

Error in homework (line 271)
            ysave(i) = y;
```

Published with MATLAB® R2021b