```
% Problem 1
% Section 1.2
% Initialize the time vector
% Sampling frequency in Hz
fs = 100;
% Creating the time vector from 0 to 1
t = 0:(1/fs):1;

% Plot all of the graphs
figure(1);
% Initialize the variables for the fourier calculations
ck = 1;
FO = 1/3;
% Using a for look for the values of k since they are consecutive and
 nice
% Note that there will be 8 total graphs
for k=1:3
    test = fourierTerm(ck, k, FO, t);
    subplot(4,2,(2*k-1));
    plot(t,test);
    title(sprintf('Plot(t,test) with k = %d',k));
    subplot(4,2,2*k);
    plot(test);
    title(sprintf('Plot(test) with k = %d',k))
end
% Comparing the graphs for k = FO = 1
k = 1;
FO = 1;
test = fourierTerm(ck,k,FO,t);
subplot(4,2,7);
plot(t,test);
title('Plot(t,test) with k = Fo = 1');
subplot(4,2,8);
plot(test);
title('Plot(test) with k = Fo = 1');

% Discussion
% Modifying the k term in the exponent modifies the frequency of the
% sinusoids that are being plotted. When you plot the fourier term
 with
% respect to time you ignore the imaginary components of them, where
 just
% using the plot function includes the imaginary part.

% Section 1.3, part 1
% Create the time vector from 0 to 2 sec, sampled at 100Hz
t = 0:(1/fs):2;

% Set up the coefficients for the fourier term
ck = 1;
k = 1;
FO = 2;
```

```
% Call the fourier term function
test = fourierTerm(ck,k,FO,t);

% Plot the resulting data
figure(2);
plotComplexData(t,test);

% Discussion part 1
% The real and imaginary parts represent the real cosine part and the
 sine
% imaginary part of the Fourier term. The absolute value should be
 constant
% because there is no time dependent attenuation on the Fourier term
 and
% the unwrapped phase is a straight line because there is a constant
 time
% term affecting the phase between the real and the imaginary
 components of
% the Fourier term. If the phase were wrapped it would be the unit
 circle.

% Section 1.3, part 2
% Update the coefficients for the fourier term
ck = exp(1j*pi/4);

% Call the fourier term function
test = fourierTerm(ck,k,FO,t);

% Plot the resulting data
figure(3);
plotComplexData(t,test);

% Discussion part 2
% The situation where you have an imaginary Fourier coefficient ck,
 there
% will be a phase shift introduced to the term. This is evident in the
% plots of the real and imaginary components with respect to time. A
% vertical shift in the phase plot corresponds to a horizontal shift
 in the
% time domain.

% Section 1.3, part 3
% Update the coefficients for the fourier term
ck = 1;

% Call the fourier term function
test = fourierTerm(ck,k,FO,t) + fourierTerm(ck,k,-1*FO,t);

% Plot the resulting data
figure(4);
plotComplexData(t,test);

% Discussion part 3
```

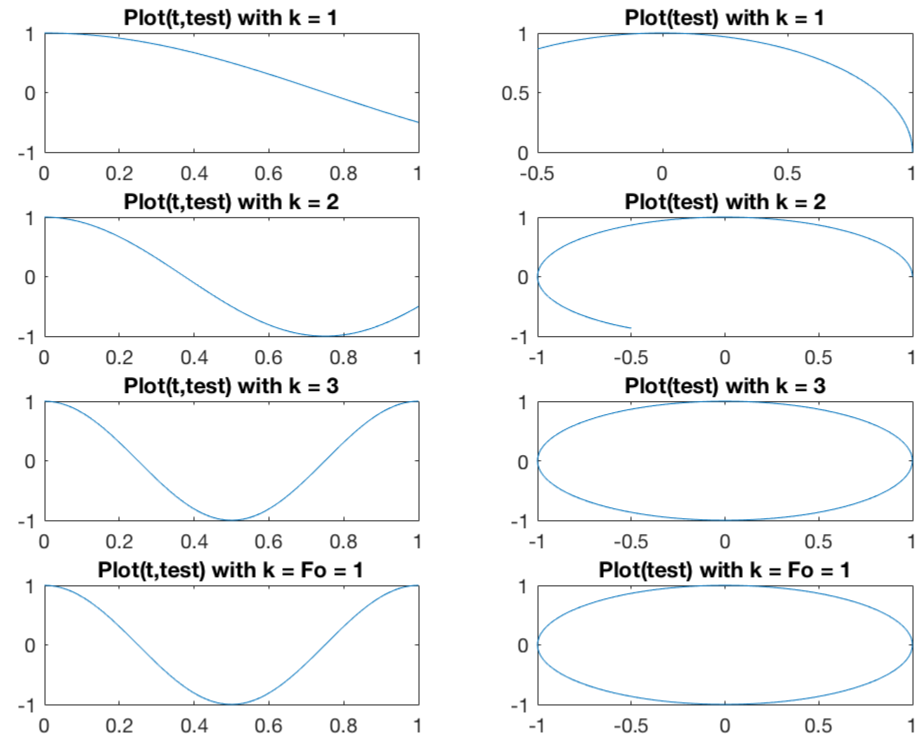
```
% The phases for these Fourier terms are designed to cancel out the
% imaginary components of each other but leave the real components
 intact.
% This translates into the euler expansion because sine is an odd
 function
% and cosine is an even function, so when you make the fundamental
% frequencies opposite of each other and then add the components, the
% cosine parts add and the sine parts become opposites of each other
 and
% cancel out.

% Section 1.3, part 4
% Update the coefficients for the fourier term
ck = 1*j;

% Call the fourier term function
test = fourierTerm(ck,k,FO,t) + fourierTerm(-1*ck,k,FO,t);

% Plot the resulting data
figure(5);
plotComplexData(t,test);

% Discussion part 4
% Similar to the last situation, in the euler expansion the sine
 components
% cancel each other out, but additionally the cosine components cancel
 each
% other, so the magnitude of the signal becomes zero. This is because
 the
% fourier coefficients make the Fourier terms complex coefficients of
 each
% other so they are exactly opposite and cancel out perfectly.

% Section 1.3, part 5
% Update the Fourier term and plot
test = test + fourierTerm(2,0,FO,t);
figure(6);
plotComplexData(t,test);

% Discussion part 5
% Adding another fourier term with co = 2 for k = 0 does nothing to
 the
% fourier term besides add a vertical shift to the real component to
 it.
% This is basically adding a 0 frequency component to the Fourier
 term. In
% this case the fundamental frequency is irrelevant because the k=0
 term
% forces it to be zero regardless of what FO is.

type('fourierTerm.m');
type('plotComplexData.m');

% Problem 2 Synthesizing other waveforms
```

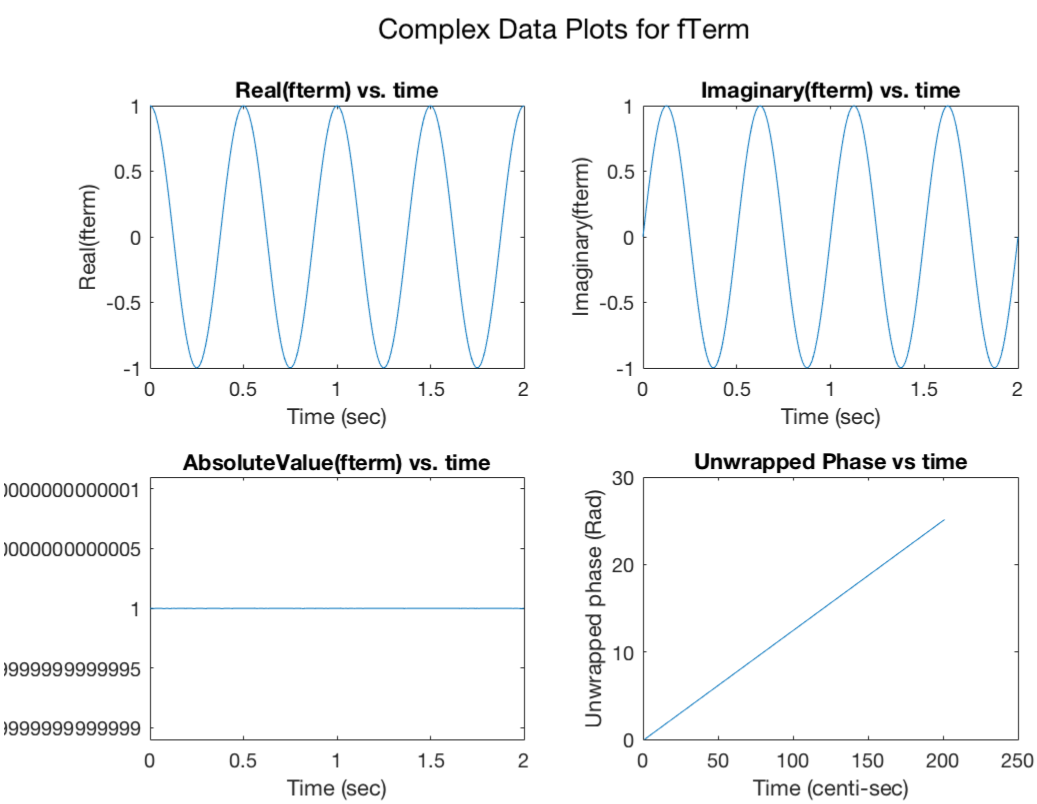
```
% Choosing values for Tp and tau
Tp = 2;
tau = 0.5;

% Compute x(t) for K = 1,2,3,10,50,100,500
K = 1;
X = sumFourierSeries(K,Tp,tau);

figure(7);
plot(real(X));
xlabel('time (centiseconds)');
ylabel('x(t)');
title(sprintf('x(t) vs. time, K = %d',K));

K = 2;
X = sumFourierSeries(K,Tp,tau);

figure(8);
plot(real(X));
xlabel('time (centiseconds)');
ylabel('x(t)');
title(sprintf('x(t) vs. time, K = %d',K));

K = 3;
X = sumFourierSeries(K,Tp,tau);

figure(9);
plot(real(X));
xlabel('time (centiseconds)');
ylabel('x(t)');
title(sprintf('x(t) vs. time, K = %d',K));

K = 10;
X = sumFourierSeries(K,Tp,tau);

figure(10);
plot(real(X));
xlabel('time (centiseconds)');
ylabel('x(t)');
title(sprintf('x(t) vs. time, K = %d',K));

K = 50;
X = sumFourierSeries(K,Tp,tau);

figure(11);
plot(real(X));
xlabel('time (centiseconds)');
ylabel('x(t)');
title(sprintf('x(t) vs. time, K = %d',K));

K = 100;
X = sumFourierSeries(K,Tp,tau);

figure(12);
```

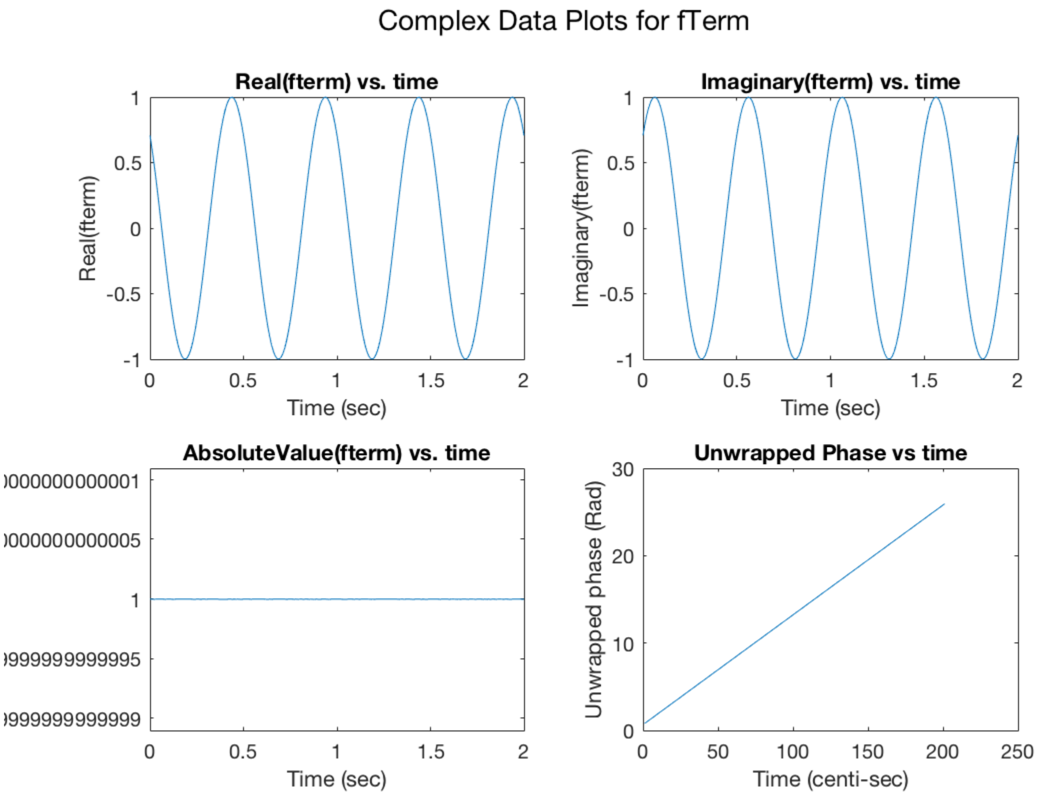
```
plot(real(X));
xlabel('time (centiseconds)');
ylabel('x(t)');
title(sprintf('x(t) vs. time, K = %d',K));

K = 500;
X = sumFourierSeries(K,Tp,tau);

figure(13);
plot(real(X));
xlabel('time (centiseconds)');
ylabel('x(t)');
title(sprintf('x(t) vs. time, K = %d',K));

% Discussion
% You can see as K increases the corner of the boxcar waves become
 more and
% more well defined, and where K is close to 1, is behaves more
% sinusoidally. This is because as more and more sinusoids are
 combined to
% create the function the edges become more defined, but you can still
 see
% the large oscillations that happen near the edge with some of the
 plots
% (i.e. k = 100, k = 10)

type('pulseTrainDFS.m');
type('sumFourierSeries.m');

Warning: Imaginary parts of complex X and/or Y arguments ignored
Warning: Imaginary parts of complex X and/or Y arguments ignored
Warning: Imaginary parts of complex X and/or Y arguments ignored
Warning: Imaginary parts of complex X and/or Y arguments ignored

function fterm = fourierTerm(ck, k, FO, t)
% function fterm = fourierTerm(ck, k, FO, t)
% fourierTerm calculates and returns a single term of the fourier
 series
% Inputs:
%   ck = fourier coefficients
%   k = which fourier coefficient
%   FO = fundamental frequency
%   t = vector of times in seconds
% Outputs:
%   fterm = vector of same length t for fourier coefficient

% Do the calculation
fterm = ck*exp(1j*2*k*pi*FO*t);

% Testing using the cosine and sine expansion of eulers
% fterm = ck*(cos(2*pi*k*FO*t) + 1j*sin(2*pi*k*FO*t));

return
```

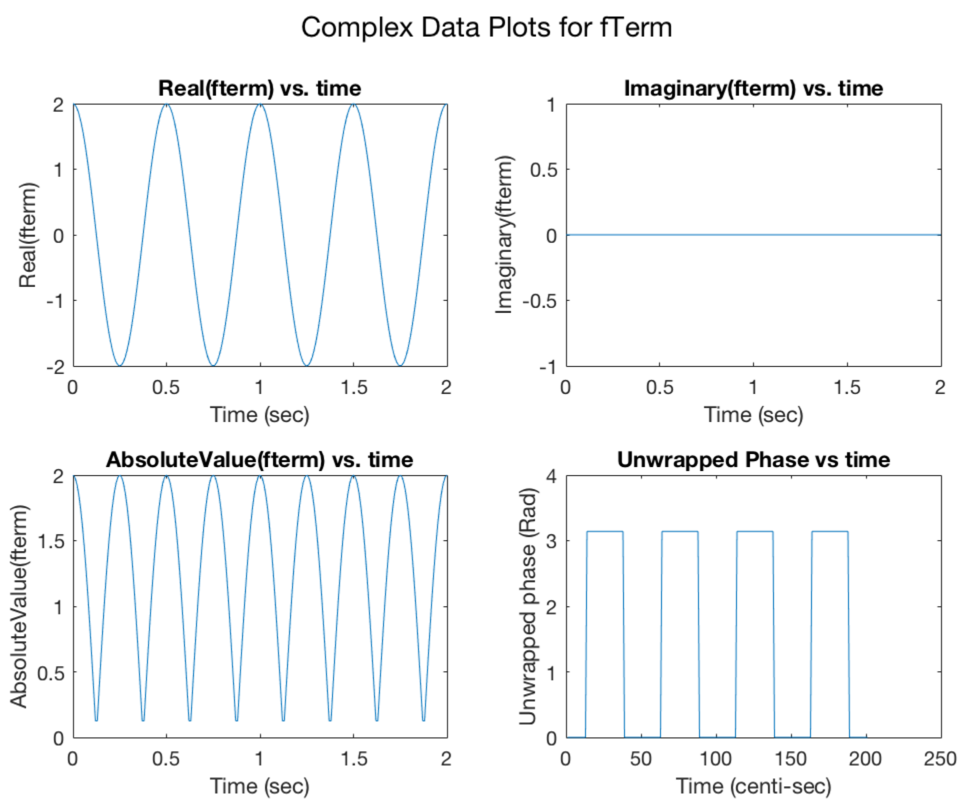
```
function plotComplexData(t,fterm)
% function plotComplexData(t,fterm)
% This function takes as an input a time vector and a calculated
 fourier
% vector, and plots them. It creates 4 plots:
%   - the real part of fterm vs. time
%   - the imaginary part of fterm vs. time
%   - absolute value vs. time
%   - unwrapped phase

% Plot the real vs time
subplot(2,2,1);
plot(t,real(fterm));
title('Real(fterm) vs. time');
ylabel('Real(fterm)');
xlabel('Time (sec)');
% Plot the imaginary vs time
subplot(2,2,2);
plot(t,imag(fterm));
title('Imaginary(fterm) vs. time');
ylabel('Imaginary(fterm)');
xlabel('Time (sec)');
% Plot the absolute value vs time
subplot(2,2,3);
plot(t,abs(fterm));
title('AbsoluteValue(fterm) vs. time');
ylabel('AbsoluteValue(fterm)');
xlabel('Time (sec)');
% Plot the unwrapped phase
subplot(2,2,4);
plot(unwrap(angle(fterm)));
title('Unwrapped Phase vs time');
ylabel('Unwrapped phase (Rad)');
xlabel('Time (centi-sec)');
suptitle('Complex Data Plots for fTerm');

function ck = pulseTrainDFS(k,Tp,tau)
% function ck = pulseTrainDFS(Tp,tau)
% implements Fourier series as calculated in example 4.1.1 of P&M, 4th
% edition
% Inputs:
%    k, the coefficient index. This function will work if you enter
%    EITHER a single value for k (i.e., '4') or a vector (i.e.,'-5:5')
%    Tp, the fundamental period of the pulse train (see Fig 4.1.3)
%    tau, the width of the each pulse
%
% ASSUMES the amplitude A = 1

% formula assumes both Tp and tau aren't zero; check
assert(Tp*tau~=0,'Tp and tau must both be nonzero')

F0 = 1/Tp;
% above:if the pulse train is repeating at times Tp (fig 4.1.3),
% the fundamental period is Tp, so the fundamental freq is 1/Tp
```

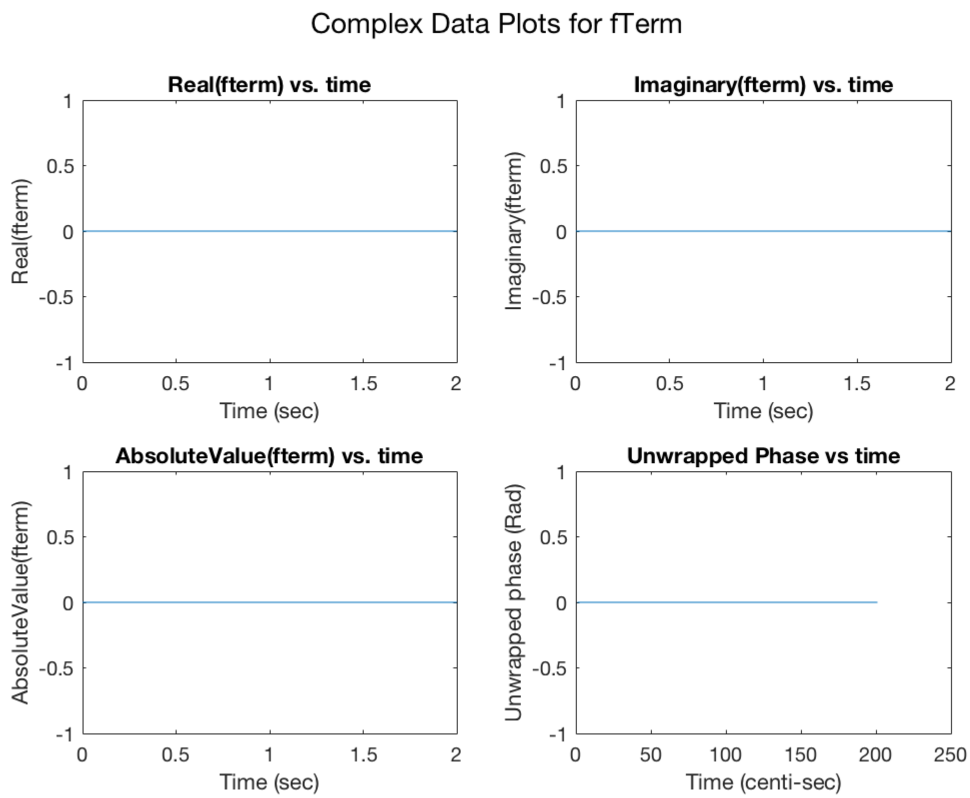
```
% do the calculation in Eq 4.1.18.
% Note, this gives the WRONG ANSWER for k=0
ck = tau/Tp *sin(pi*k*F0*tau)./(pi*k*F0*tau);

% go back and correct the k=0 term
ck(k==0) =  tau/Tp;  % eq 4.1.17 in book


return

function vecSum = sumFourierSeries(K,Tp,tau)
% function vecSum = sumFourierSeries(K,Tp,tau)

vecSum = 0;

% set up a time vector t
t = -4:(1/100):4;

% now, loop over the coefficients k, from -K to K
% Inside the loop, do:
%    calculate the value of ck, from pulseTrainDFS.m
%    calculate the fourier term from 'fourierTerm.m'
%    add up the new term to a running sum 'vecSum'
for k =-K:K
    ck = pulseTrainDFS(k,Tp,tau);
    test = fourierTerm(ck,k,Tp,t);
    vecSum = vecSum + test;
end

return
```

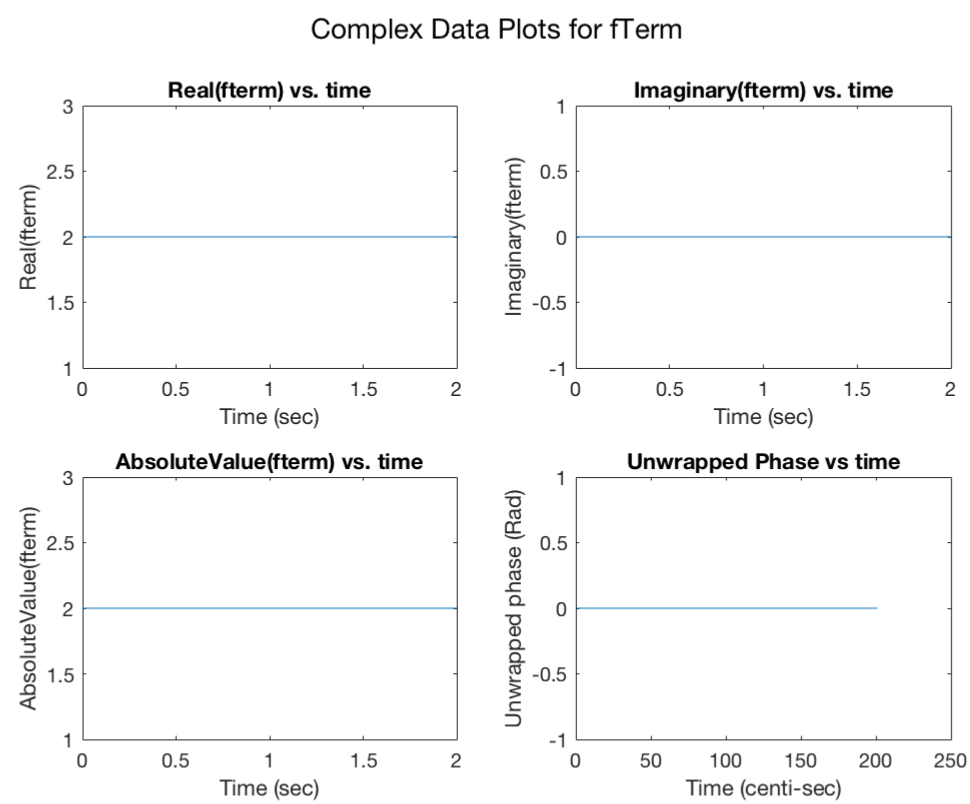

Plot(t,test) with k = 1
Plot(test) with k = 1
Plot(t,test) with k = 2
Plot(test) with k = 2
Plot(t,test) with k = 3
Plot(test) with k = 3
Plot(t,test) with k = Fo = 1
Plot(test) with k = Fo = 1

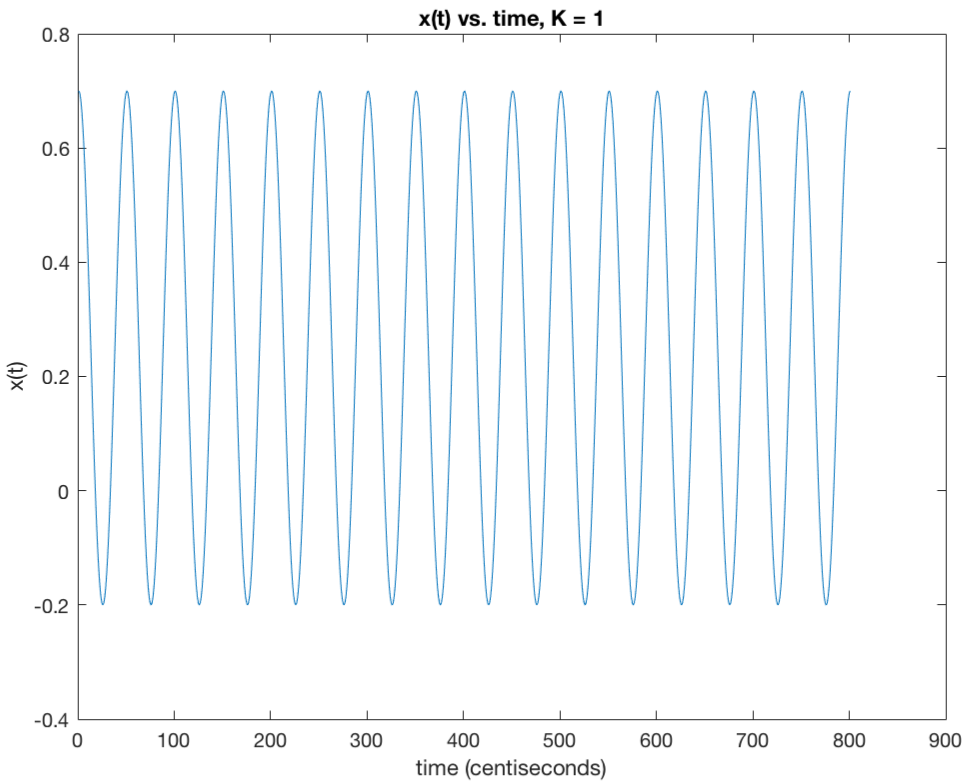

Complex Data Plots for fTerm
Real(fterm) vs. time
Real(fterm)
Time (sec)
Imaginary(fterm) vs. time
Imaginary(fterm)
Time (sec)
AbsoluteValue(fterm) vs. time
Time (sec)
Unwrapped Phase vs time
Unwrapped phase (Rad)
Time (centi-sec)

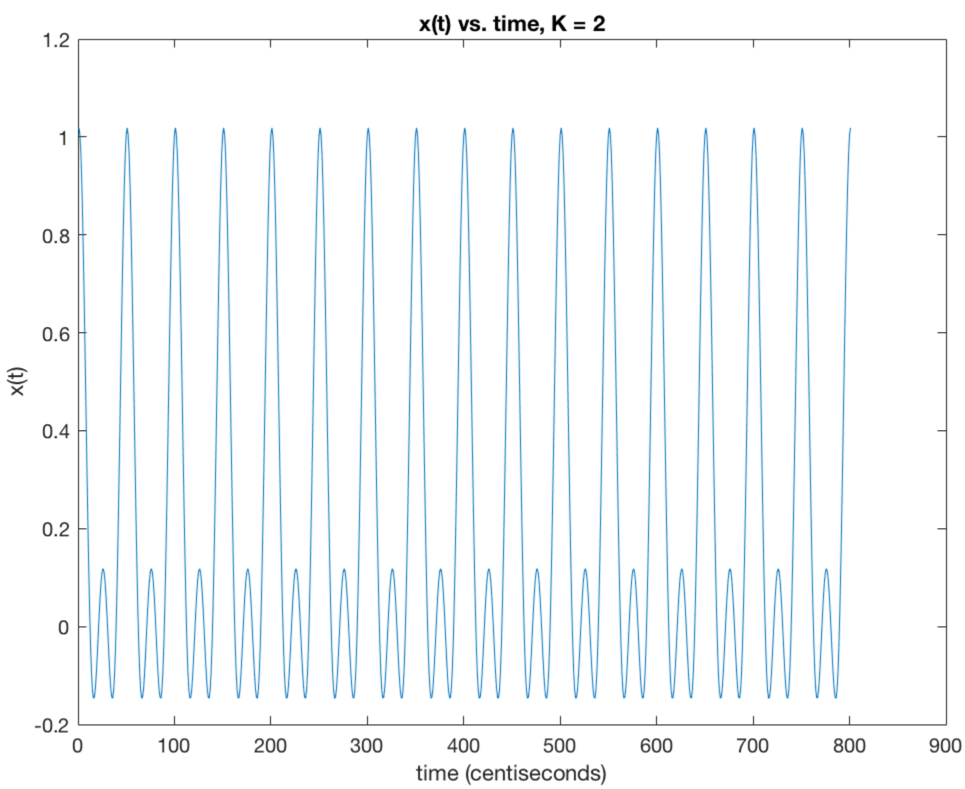

Complex Data Plots for fTerm
Real(fterm) vs. time
Real(fterm)
Time (sec)
Imaginary(fterm) vs. time
Imaginary(fterm)
Time (sec)
AbsoluteValue(fterm) vs. time
Time (sec)
Unwrapped Phase vs time
Unwrapped phase (Rad)
Time (centi-sec)

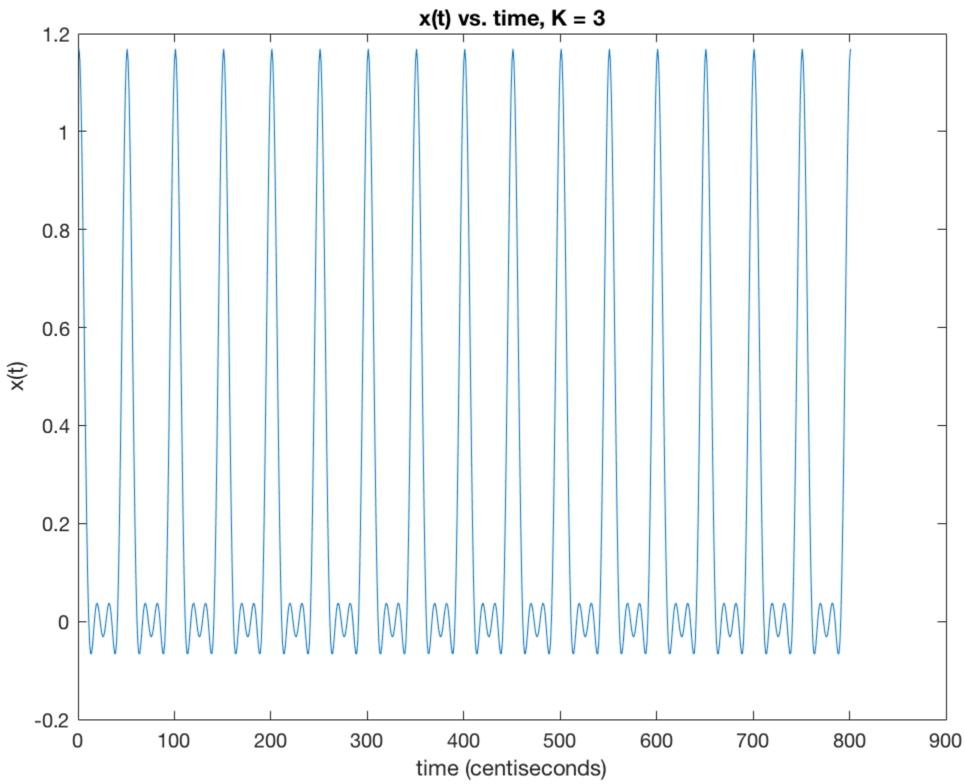

Complex Data Plots for fTerm
Real(fterm) vs. time
Real(fterm)
Time (sec)
Imaginary(fterm) vs. time
Imaginary(fterm)
Time (sec)
AbsoluteValue(fterm) vs. time
AbsoluteValue(fterm)
Time (sec)
Unwrapped Phase vs time
Unwrapped phase (Rad)
Time (centi-sec)

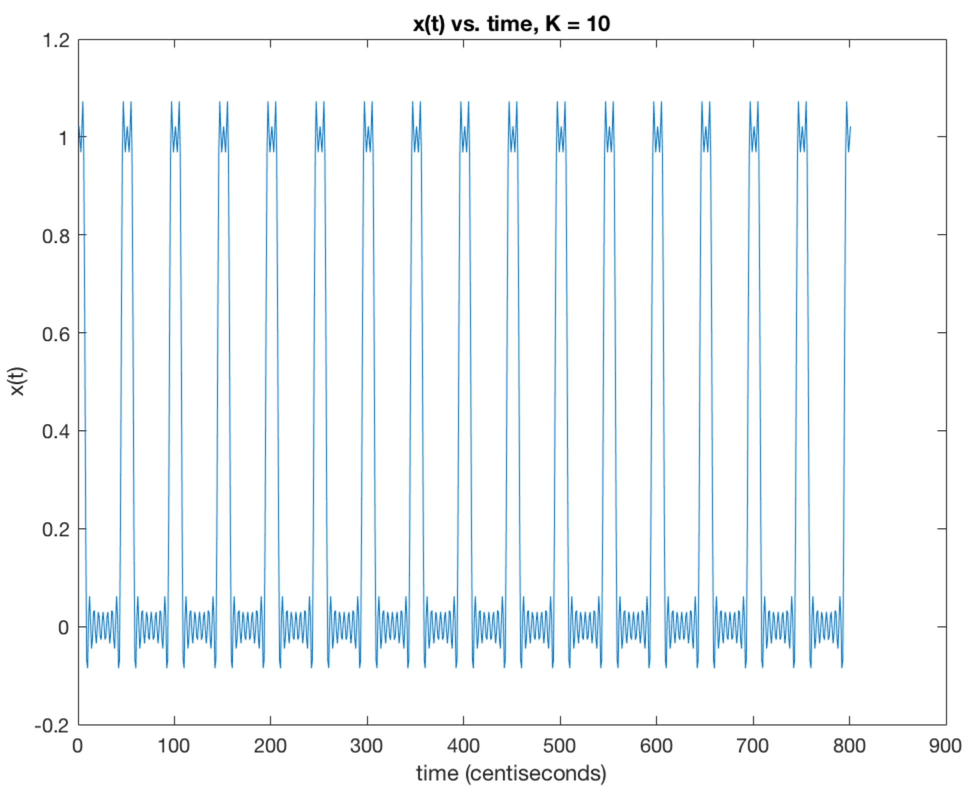

Complex Data Plots for fTerm
Real(fterm) vs. time
Real(fterm)
Time (sec)
Imaginary(fterm) vs. time
Imaginary(fterm)
Time (sec)
AbsoluteValue(fterm) vs. time
AbsoluteValue(fterm)
Time (sec)
Unwrapped Phase vs time
Unwrapped phase (Rad)
Time (centi-sec)

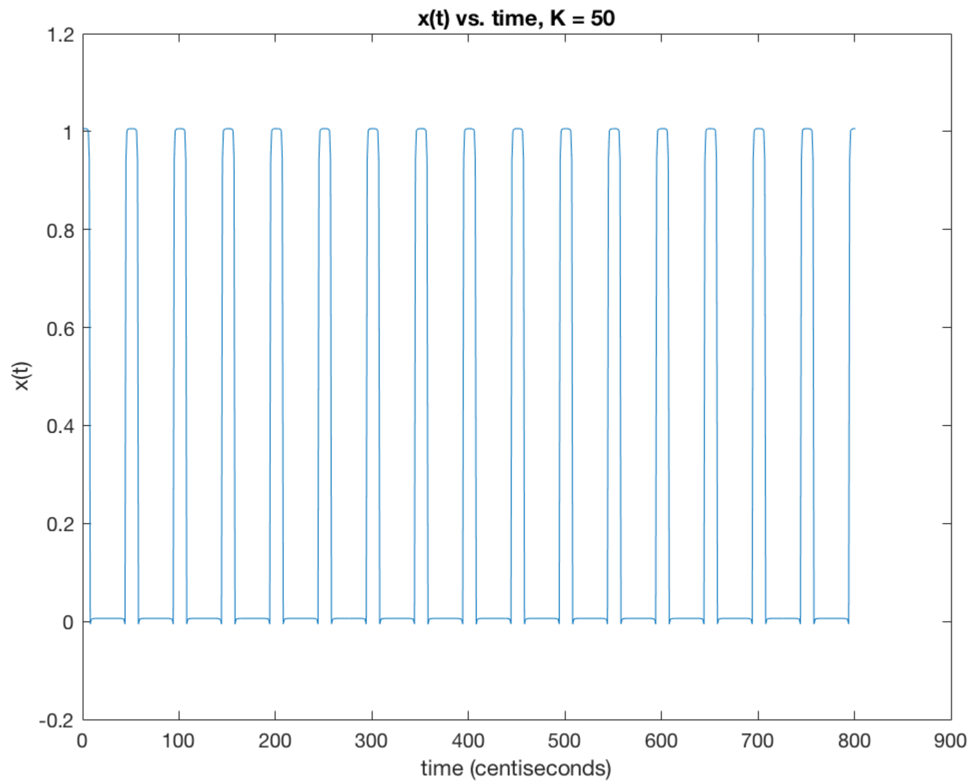

Complex Data Plots for fTerm
Real(fterm) vs. time
Real(fterm)
Time (sec)
Imaginary(fterm) vs. time
Imaginary(fterm)
Time (sec)
AbsoluteValue(fterm) vs. time
AbsoluteValue(fterm)
Time (sec)
Unwrapped Phase vs time
Unwrapped phase (Rad)
Time (centi-sec)

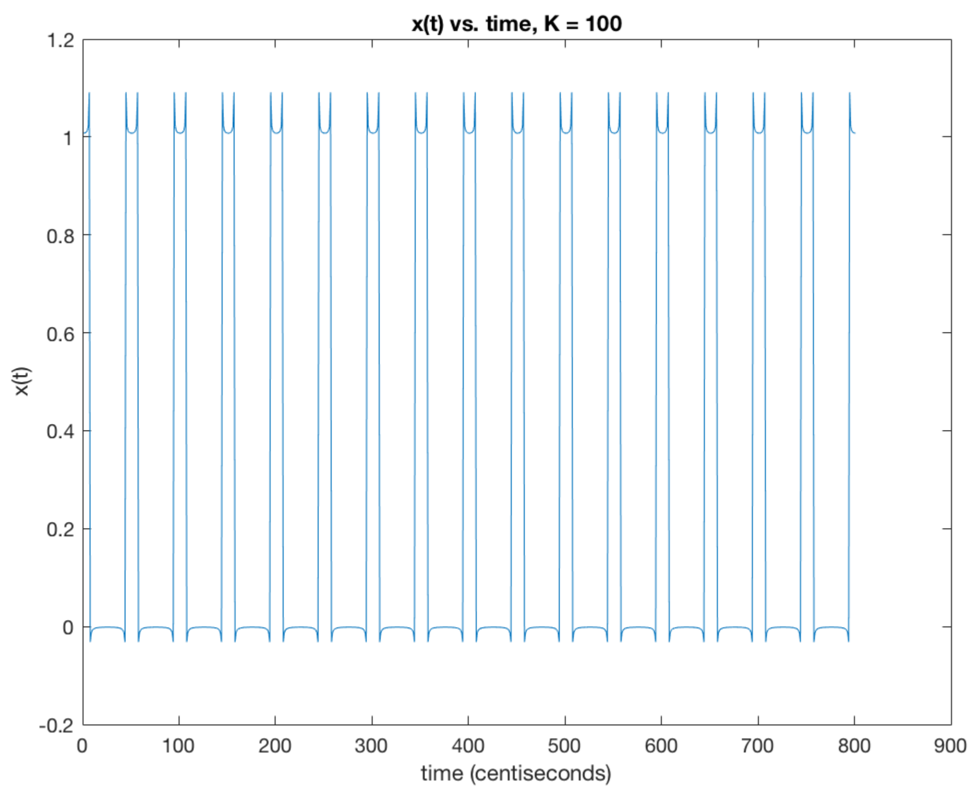

x(t) vs. time, K = 1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
x(t)
0
100
200
300
400
500
600
700
800
900
time (centiseconds)

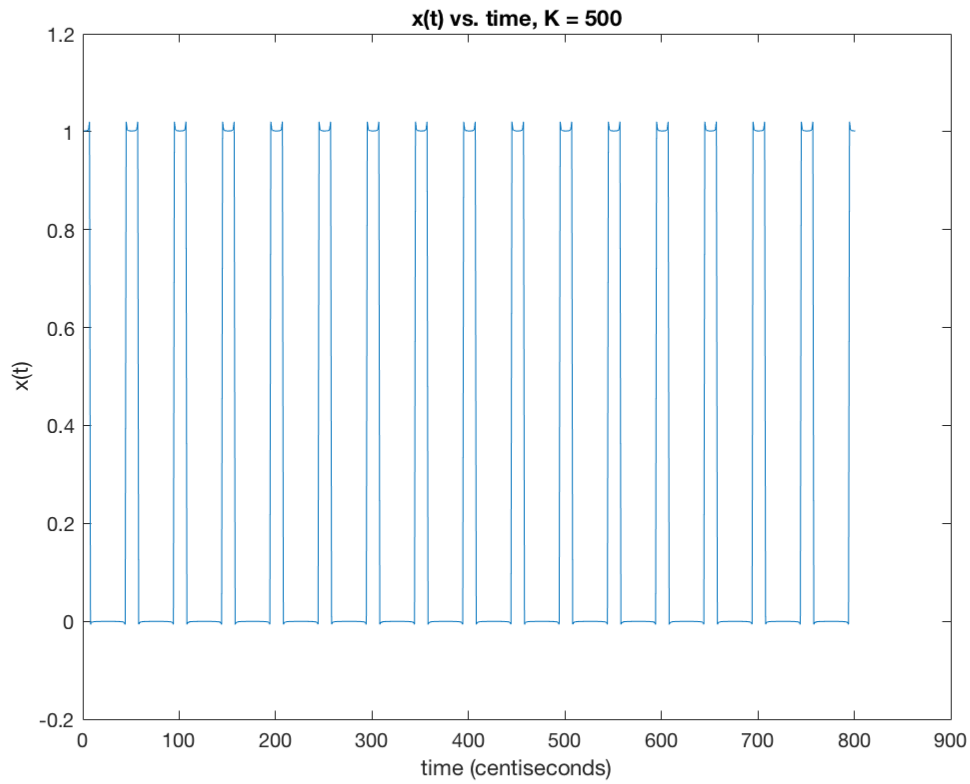

x(t) vs. time, K = 2
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
x(t)
0
100
200
300
400
500
600
700
800
900
time (centiseconds)


x(t) vs. time, K = 3
x(t)
time (centiseconds)
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
0
100
200
300
400
500
600
700
800
900


x(t) vs. time, K = 10
x(t)
time (centiseconds)
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
0
100
200
300
400
500
600
700
800
900


x(t) vs. time, K = 50
x(t)
time (centiseconds)
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
0
100
200
300
400
500
600
700
800
900


x(t) vs. time, K = 100
x(t)
time (centiseconds)
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
0
100
200
300
400
500
600
700
800
900



Published with MATLAB® R2016b