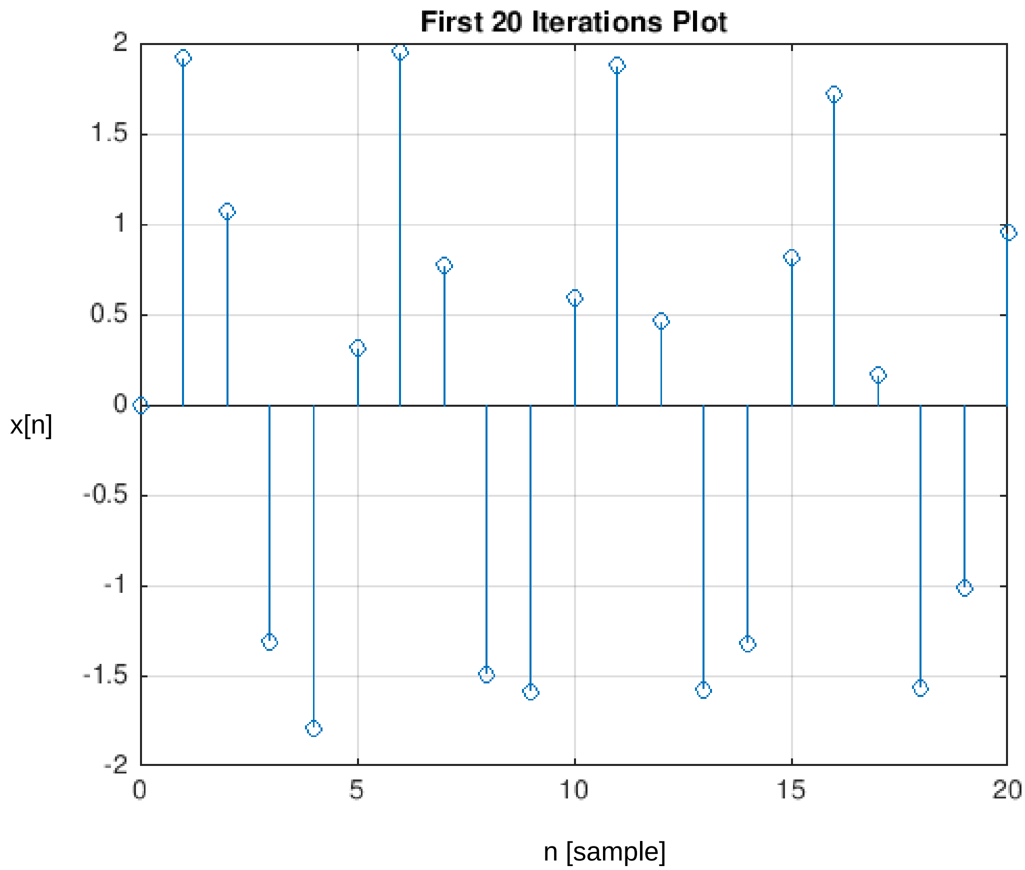
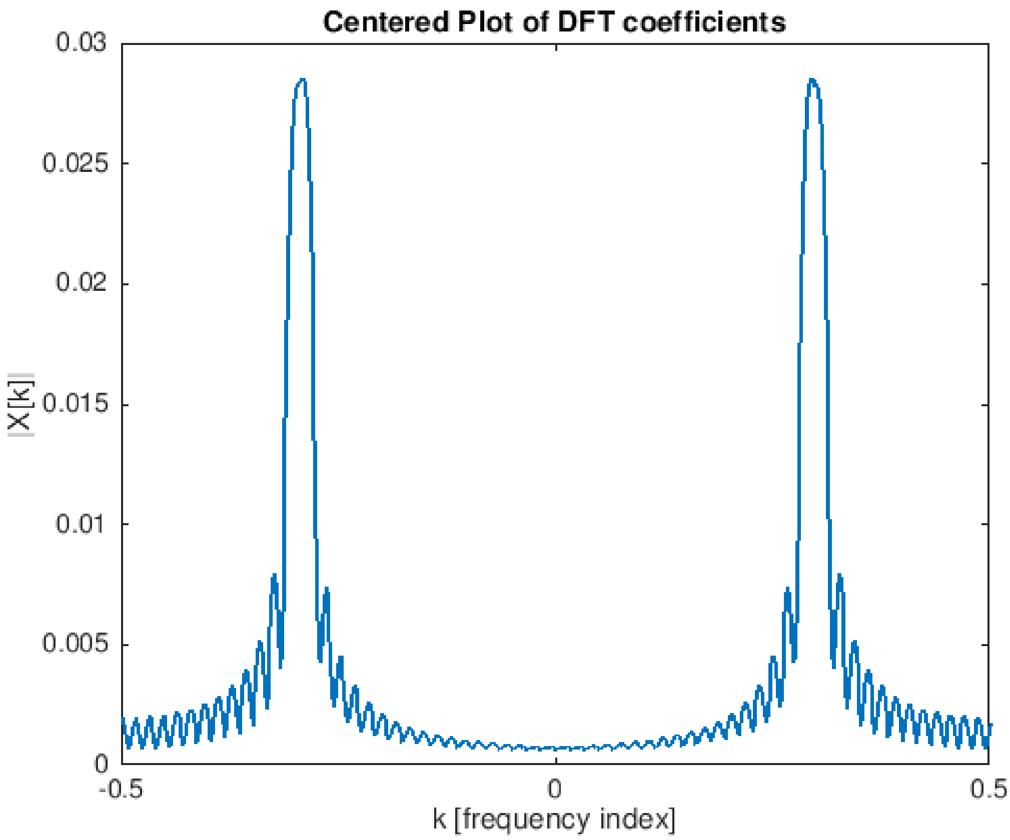
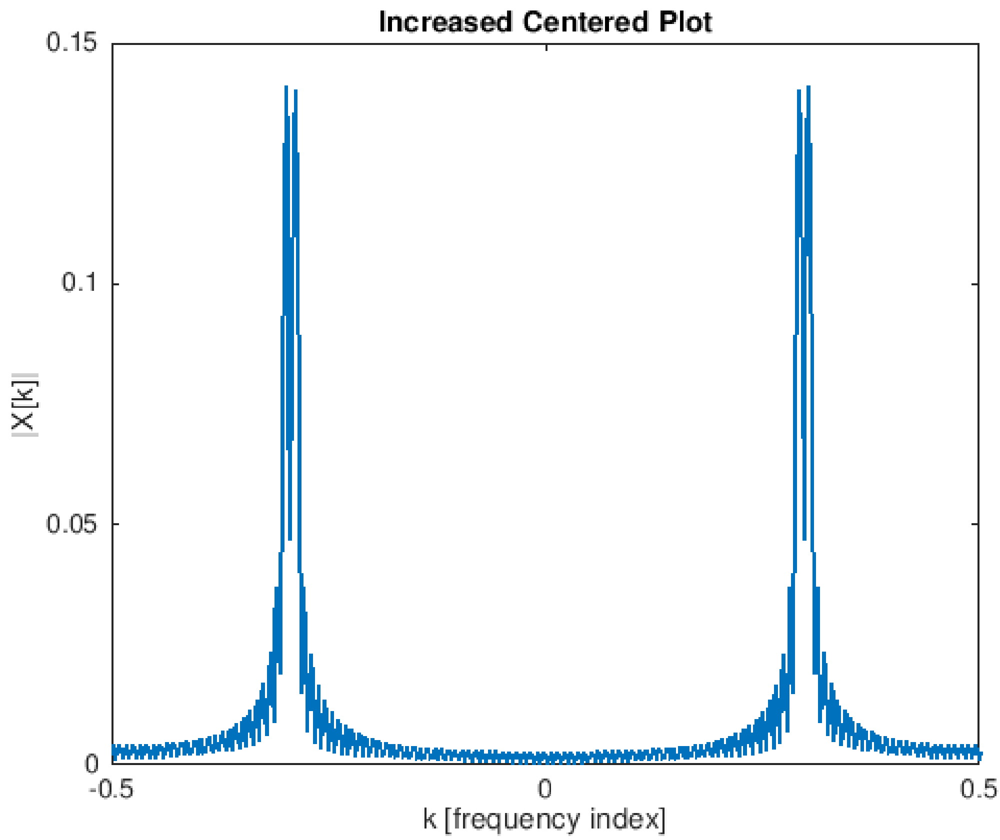
```
%%Part 1: Frequency Resolution

A1 = 1;
A2 = 1;
F1 = .2;
F2 = .21;
L = 64;

n = 0:(L-1);
X1 = A1 * sin(2*pi * F1 * n);
X2 = A2 * sin(2*pi * F2 * n);

XN = X1+X2;

figure;
hp = stem(n,XN);
grid
hold on %put plot on same figure
axis([0 20 -2 2]); %to only plot range [x1 x2 y1 y2]
title('First 20 Iterations Plot')

%Number 2 and 3: shifted frequency, centered plots
NN = 512;
fftPlot = fft(XN,NN);
y = 0:NN-1;
shifter = fftshift(fftPlot);
%m=-floor((NN)/2):1:floor((NN-1)/2); % Shifted frequency index

figure;
plot(linspace(-1/2,1/2,NN),abs(fftPlot/2)/
NN,'marker','none','Linewidth',2);

xlabel('k [frequency index]')
ylabel('|X[k]|')
title('Centered Plot of DFT coefficients')

%4: At L = 64, the two peaks are not distinguishable. Additionally,
% when N was increased from 512 to 1024, even then the two peaks were
% not distinguishabe. This is simply because N is just the order of
 the
% filter and the larger order does not impact the visibility of the
 peaks
% in any way.

%5:

%shifted frequency, centered plots

A1 = 1;
A2 = 1;
F1 = .2;
F2 = .21;
```

```
L = 128;

n = 0:(L-1);
X1 = A1 * sin(2*pi * F1 * n);
X2 = A2 * sin(2*pi * F2 * n);

XN2 = X1+X2;

NCrit = 512;
fftPlot = fft(XN2,NCrit);
y = 0:NCrit-1;
shifter = fftshift(fftPlot);
%m=-floor((NCrit)/2):1:floor((NCrit-1)/2); % Shifted frequency index

figure;
plot(linspace(-1/2,1/2,NCrit),abs(fftPlot)/
NCrit,'marker','none','Linewidth',2);
xlabel('k [frequency index]')
ylabel('|X[k]|')
title('Increased Centered Plot')

% As described in Figure 3 - Increased Centered Plot, there's a dip in
 the
% plot indicating that there are two cosines. This is clearly visible
 once the
% L value was increased. Increasing the L value means taking more
 samples to check
% the plot against. The lesson learned is that if we can afford to
 sample at a greater
% rate, that is the way to go because it provides more clarity in the
 plots and analysis.
% At L = 128, we see two peaks clearly.
```


First 20 Iterations Plot
x[n]
n [sample]
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
0
5
10
15
20

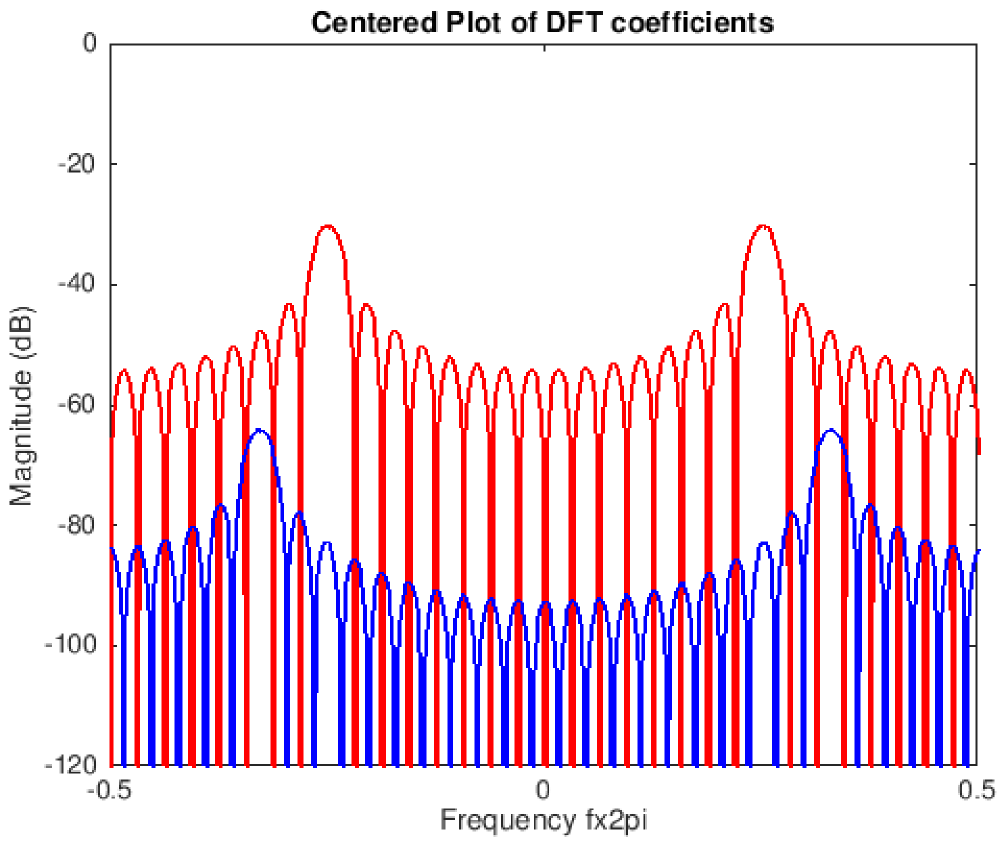

Centered Plot of DFT coefficients
0.03
0.025
0.02
0.015
0.01
0.005
0
-0.5
0
0.5
|X[k]|
k [frequency index]

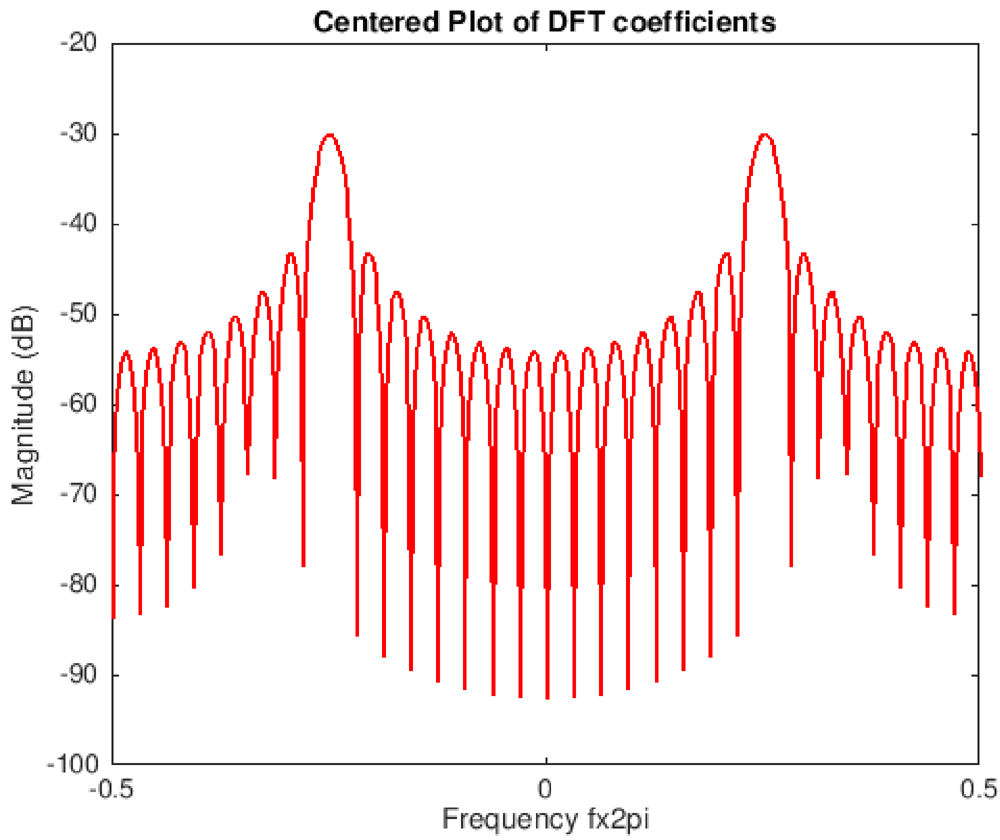


Published with MATLAB® R2017b

```
%%Part 2: Frequency Masking

%Number 1 and 2

A1 = 1;
A2 = 0.02;
L = 32;
F1 = 8/L;
F2 = 10.5/L;

n = 0:L;

X1 = A1 * sin(2*pi * F1 * n);
X2 = A2 * sin(2*pi * F2 * n);


NN = 512;
fftPlot = fft(X1,NN);
fftPlot2 = fft(X2,NN);
%plot together
shifter = fftshift(fftPlot);
shifter2 = fftshift(fftPlot2);
%m=-floor((NN)/2):1:floor((NN-1)/2); % Shifted frequency index


figure;
plot(linspace(-1/2,1/2,NN),mag2db(abs(shifter)/
NN), 'marker','none','Linewidth',2,'color','r');
hold on;
plot(linspace(-1/2,1/2,NN),mag2db(abs(shifter2)/
NN), 'marker','none','Linewidth',2,'color','b');
hold off;
ylim([-120 0])
%ax=axis;
%axis([-floor(NN/2) floor((NN-1)/2) ax(3:4)]);
xlabel('Frequency fx2pi')
ylabel('Magnitude (dB)')
title('Centered Plot of DFT coefficients')

% Using the dB scaling allows us to tell the peaks separately.
 Relative to the data length, there are both
% good and bad frequency separations. Towards the top of the plot,
 there are clear frequency separations, however
% towards the bottom, there is a cluster due to the nature of the
 frequencies being passed.

%Number 3
XN = X1 + X2;
fftPlot3 = fft(XN,NN);
shifter3 = fftshift(fftPlot3);
figure;
```

```
plot(linspace(-1/2,1/2,NN),mag2db(abs(shifter3)/
NN), 'marker','none','Linewidth',2,'color','r');
hold on;
ylim([-100 -20])
%ax=axis;
%axis([-floor(NN/2) floor((NN-1)/2) ax(3:4)]);
xlabel('Frequency fx2pi')
ylabel('Magnitude (dB)')
title('Centered Plot of DFT coefficients')

% In the X[n] signal, only one peak is visible instead of the two.
 This is because the amplitudes are far off,
% therefore, we can see where the lower amplitude should be but cannot
 actually pin point it in the plot.
```





Published with MATLAB® R2017b

```
%%Part 3: Windowing

A1 = 1;
A2 = 0.02;
L = 32;
F1 = 8/L;
F2 = 10.5/L;
N = 512;

%Number 1
%Hanning
n = 0:(L-1);
X1 = (A1*sin(2*pi*F1*n));
X2 = (A2*sin(2*pi*F2*n));
x = X1+X2;

w = hanning(L)'; % Choose Window
S1=sum(w)/L;     % Compute Window Scaling

X=fft(x.*w,L);   % Compute DFT
k=0:L-1;      % frequency index vector (for plots)

% Plot properly centered and scaled coefficients
S=L*S1; % Scaling = DFT * Windows Scaling
figure
Y=fftshift(X)/S; % Spectrum with proper scaling
%m=-floor((N)/2):1:floor((N-1)/2);
plot(linspace(-1/2,1/2,L),mag2db(abs(Y)),'marker','none','Linewidth',2);
hold on;
xlabel('k [frequency index]')
ylabel('|X[k]|')

title('Spectrum of windowed signal - Hanning')



%Haming
n = 0:(L-1);
X1 = (A1*sin(2*pi*F1*n));
X2 = (A2*sin(2*pi*F2*n));
x = X1+X2;

w = hamming(L)'; % Choose Window
S1=sum(w)/L;     % Compute Window Scaling

X=fft(x.*w,L);   % Compute DFT
k=0:L-1;      % frequency index vector (for plots)

% Plot properly centered and scaled coefficients
S=L*S1; % Scaling = DFT * Windows Scaling
figure
Y=fftshift(X)/S; % Spectrum with proper scaling
```

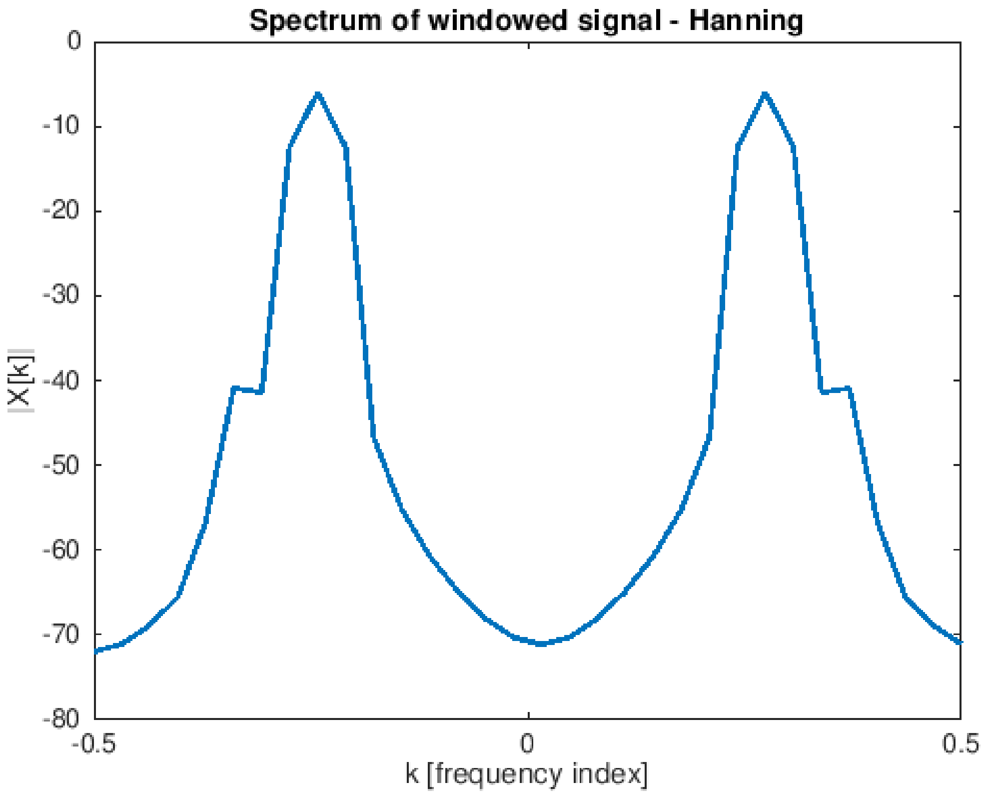
```
%m=-floor((N)/2):1:floor((N-1)/2);
plot(linspace(-1/2,1/2,L),mag2db(abs(Y)),'marker','none','Linewidth',2);
hold on;
xlabel('k [frequency index]')
ylabel('|X[k]|')

title('Spectrum of windowed signal - Hamming')




%Triangular
n = 0:(L-1);
X1 = (A1*sin(2*pi*F1*n));
X2 = (A2*sin(2*pi*F2*n));
x = X1+X2;

w = triang(L)'; % Choose Window
S1=sum(w)/L;     % Compute Window Scaling

X=fft(x.*w,L);   % Compute DFT
k=0:L-1;      % frequency index vector (for plots)

% Plot properly centered and scaled coefficients
S=L*S1; % Scaling = DFT * Windows Scaling
figure
Y=fftshift(X)/S; % Spectrum with proper scaling
%m=-floor((N)/2):1:floor((N-1)/2);
plot(linspace(-1/2,1/2,L),mag2db(abs(Y)),'marker','none','Linewidth',2);
hold on;
xlabel('k [frequency index]')
ylabel('|X[k]|')

title('Spectrum of windowed signal - Triangular')


%Chebwin
n = 0:(L-1);
X1 = (A1*sin(2*pi*F1*n));
X2 = (A2*sin(2*pi*F2*n));
x = X1+X2;

w = chebwin(L)'; % Choose Window
S1=sum(w)/L;     % Compute Window Scaling

X=fft(x.*w,L);   % Compute DFT
k=0:L-1;      % frequency index vector (for plots)

% Plot properly centered and scaled coefficients
S=L*S1; % Scaling = DFT * Windows Scaling
figure
Y=fftshift(X)/S; % Spectrum with proper scaling
%m=-floor((N)/2):1:floor((N-1)/2);
plot(linspace(-1/2,1/2,L),mag2db(abs(Y)),'marker','none','Linewidth',2);
```

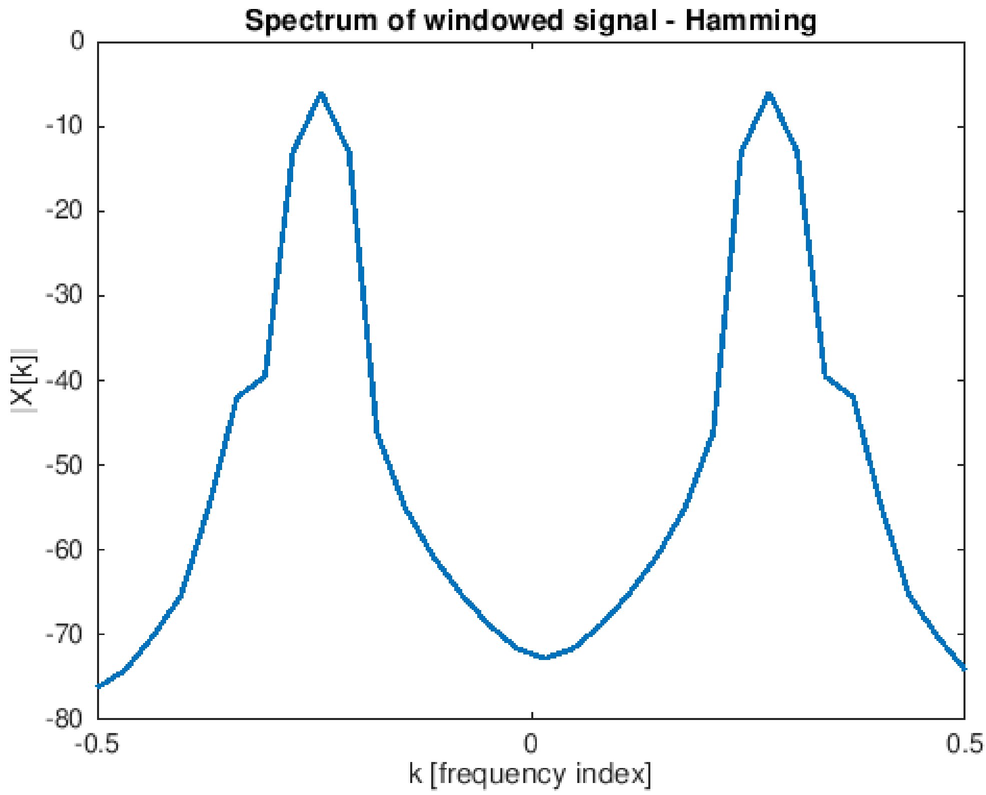
```
hold on;
xlabel('k [frequency index]')
ylabel('|X[k]|')
title('Spectrum of windowed signal - Chebwin')

% The advantage of using dB scale for the vertical axis. This allows
 for the peaks
% to be visible otherwise the plots will stick to the top and or
 bottom of the graphs.

%Number 2

% Yes, windowing provides an optimal view of the plots highlighting
 both peaks in our case
% very evidently. More importantly, windowing in this case took a
 smaller subset of the larger
% existing dataset to process the plots and show the two peaks
 evidently. In each of the respective
% windowing cases, the contents were modified and the respectively
 plots were exposed. Additionally,
% it is important to note that one of the main things solved by
 windowing is reducing of frequency
% masking which is what makes it so useful.

% In this event, Chebyshev looks the cleanest in the case that it gets
 rid of a lot of noise to clarify
% the showcasing of the plot. On the other hand, the others did not
 eliminate the noise completely but it
% allowed for a better ability to view other frequencies or signals at
 a lower amplitude. Regardless, in windowing
% we are expected to lose lower signals anyway which is why Chebyshev
 seems to be the cleanest.

% Number 3

% Since in windowing, we're only taking a small subset of a larger
 dataset, we find that the price paid is that some smaller
% signals are lost in plotting the graph. On the other hand, the
 process reduces frequency masking which is the flip side advantange
% in doing windowing.

% Number 4

A1 = 1;
A2 = 0.02;
L = 32;
F1 = 8/L;
F2 = 9.5/L;
N = 512;


%Hanning
n = 0:(L-1);
X1 = (A1*sin(2*pi*F1*n));
```

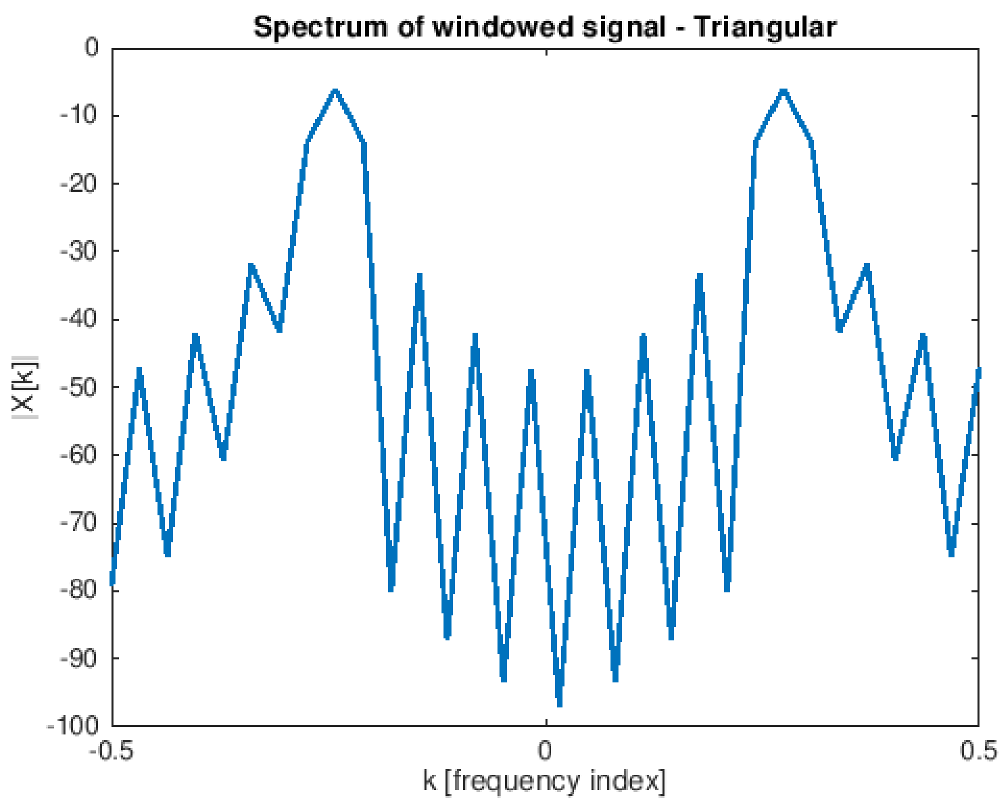
```
X2 = (A2*sin(2*pi*F2*n));
x = X1+X2;

w = hanning(L)'; % Choose Window
S1=sum(w)/L;     % Compute Window Scaling

X=fft(x.*w,L);   % Compute DFT
k=0:L-1;      % frequency index vector (for plots)

% Plot properly centered and scaled coefficients
S=L*S1; % Scaling = DFT * Windows Scaling
figure
Y=fftshift(X)/S; % Spectrum with proper scaling
%m=-floor((N)/2):1:floor((N-1)/2);
plot(linspace(-1/2,1/2,L),mag2db(abs(Y)),'marker','none','Linewidth',2);
hold on;
xlabel('k [frequency index]')
ylabel('|X[k]|')

title('Spectrum of windowed signal - Hanning')



%Haming
n = 0:(L-1);
X1 = (A1*sin(2*pi*F1*n));
X2 = (A2*sin(2*pi*F2*n));
x = X1+X2;

w = hamming(L)'; % Choose Window
S1=sum(w)/L;     % Compute Window Scaling

X=fft(x.*w,L);   % Compute DFT
k=0:L-1;      % frequency index vector (for plots)

% Plot properly centered and scaled coefficients
S=L*S1; % Scaling = DFT * Windows Scaling
figure
Y=fftshift(X)/S; % Spectrum with proper scaling
%m=-floor((N)/2):1:floor((N-1)/2);
plot(linspace(-1/2,1/2,L),mag2db(abs(Y)),'marker','none','Linewidth',2);
hold on;
xlabel('k [frequency index]')
ylabel('|X[k]|')

title('Spectrum of windowed signal - Hamming')




%Triangular
n = 0:(L-1);
X1 = (A1*sin(2*pi*F1*n));
X2 = (A2*sin(2*pi*F2*n));
```

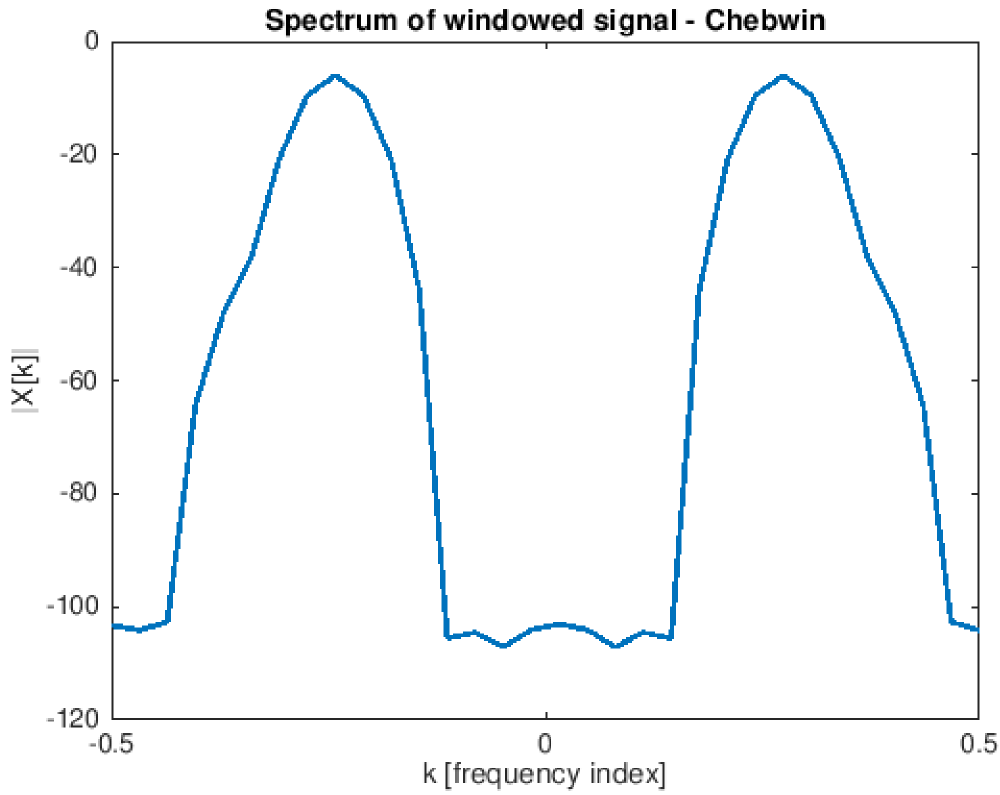
```
x = X1+X2;

w = triang(L)'; % Choose Window
S1=sum(w)/L;     % Compute Window Scaling

X=fft(x.*w,L);   % Compute DFT
k=0:L-1;      % frequency index vector (for plots)

% Plot properly centered and scaled coefficients
S=L*S1; % Scaling = DFT * Windows Scaling
figure
Y=fftshift(X)/S; % Spectrum with proper scaling
%m=-floor((N)/2):1:floor((N-1)/2);
plot(linspace(-1/2,1/2,L),mag2db(abs(Y)),'marker','none','Linewidth',2);
hold on;
xlabel('k [frequency index]')
ylabel('|X[k]|')

title('Spectrum of windowed signal - Triangular')


%Chebwin
n = 0:(L-1);
X1 = (A1*sin(2*pi*F1*n));
X2 = (A2*sin(2*pi*F2*n));
x = X1+X2;

w = chebwin(L)'; % Choose Window
S1=sum(w)/L;     % Compute Window Scaling

X=fft(x.*w,L);   % Compute DFT
k=0:L-1;      % frequency index vector (for plots)

% Plot properly centered and scaled coefficients
S=L*S1; % Scaling = DFT * Windows Scaling
figure
Y=fftshift(X)/S; % Spectrum with proper scaling
%m=-floor((N)/2):1:floor((N-1)/2);
plot(linspace(-1/2,1/2,L),mag2db(abs(Y)),'marker','none','Linewidth',2);
hold on;
xlabel('k [frequency index]')
ylabel('|X[k]|')
title('Spectrum of windowed signal - Chebwin')

% To answer this question, it's a little skewed. Since the amplitudes
 are different,
% we are not clearly seeing the two peeks. In the past, the issue was
 that if we sample
% at a faster rate, we would see the split within the peeks. In this
 case, while the grand
% image looks like a peek, the granular plot does not split into peeks
 therefore I would say that
% there is too much noise and is not safe to conclude that we will
 indeed get two peeks, unless of course
```

```
% the limitation bar for noise can be accounted for as a peek simply
 in graphical terms.
```

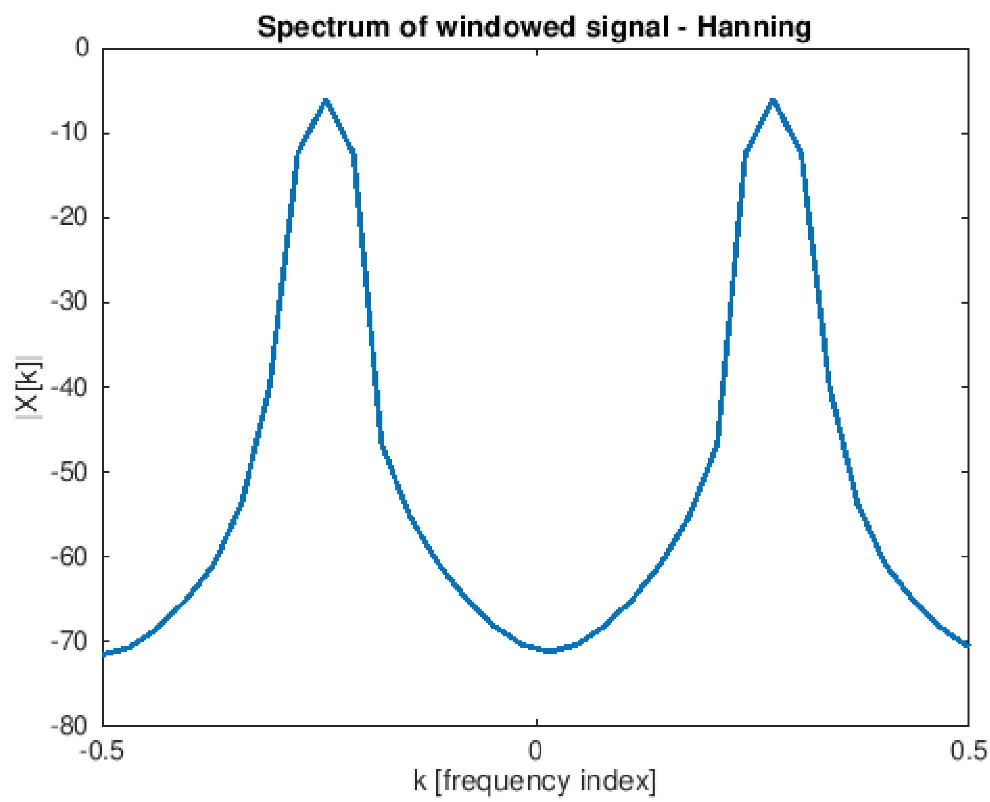



Spectrum of windowed signal - Hamming
0
-10
-20
-30
-40
-50
-60
-70
-80
|X[k]|
-0.5
0
0.5
k [frequency index]

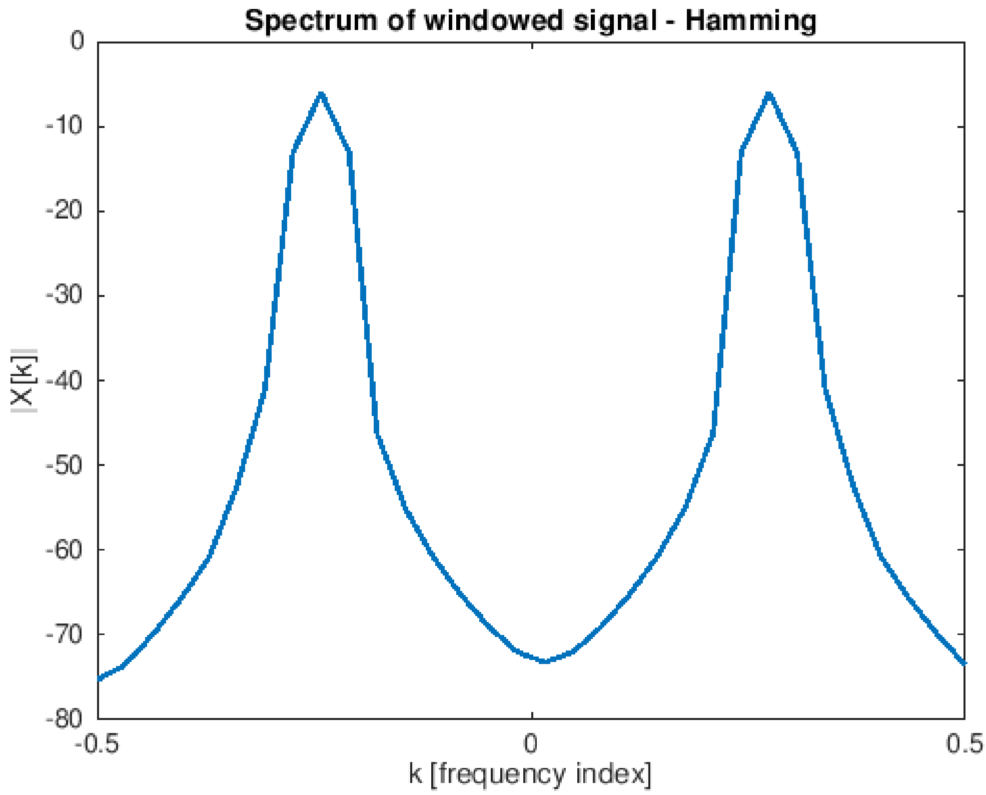

Spectrum of windowed signal - Triangular
0
-10
-20
-30
-40
-50
-60
-70
-80
-90
-100
|X[k]|
-0.5
0
0.5
k [frequency index]

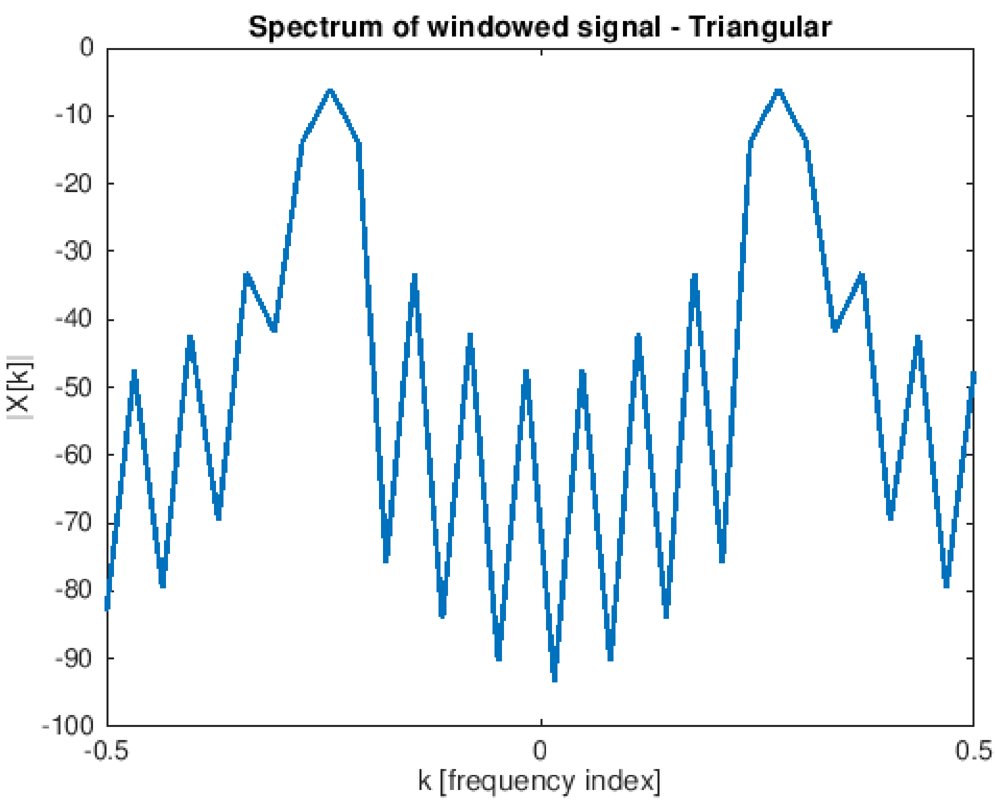


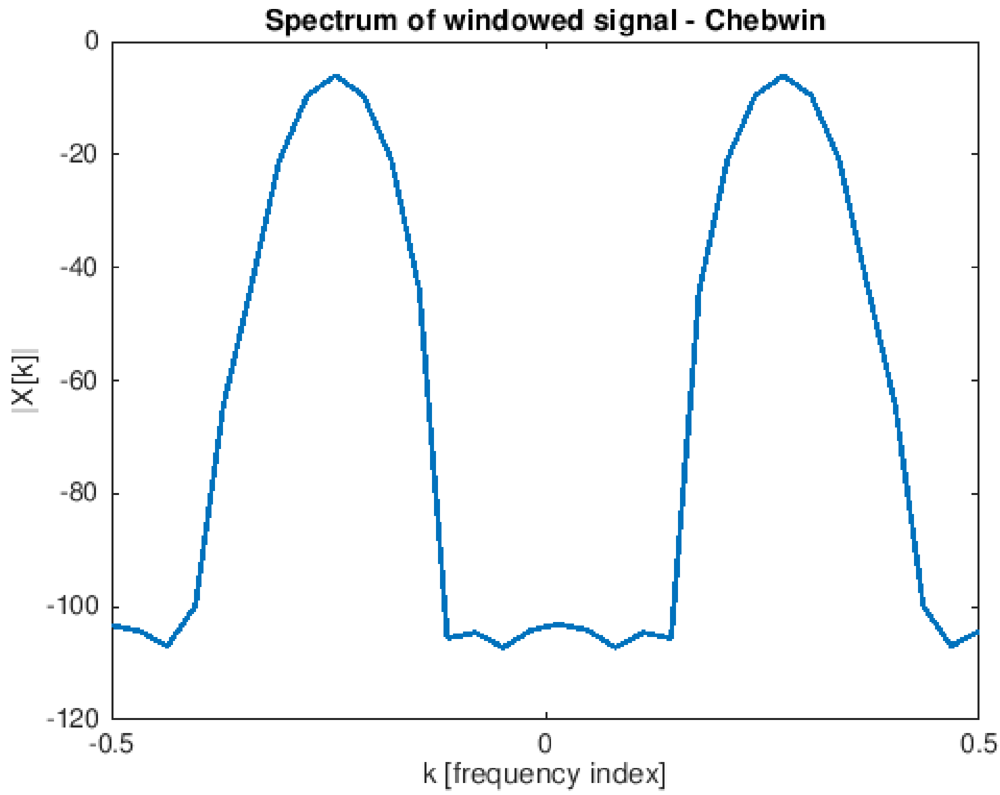
Spectrum of windowed signal - Chebwin
0
-20
-40
-60
-80
-100
-120
|X[k]|
-0.5
0
0.5
k [frequency index]



Spectrum of windowed signal - Hanning
0
-10
-20
-30
-40
-50
-60
-70
-80
|X[k]|
-0.5
0
0.5
k [frequency index]


Spectrum of windowed signal - Hamming
0
-10
-20
-30
-40
-50
-60
-70
-80
|X[k]|
-0.5
0
0.5
k [frequency index]


Spectrum of windowed signal - Triangular
0
-10
-20
-30
-40
-50
-60
-70
-80
-90
-100
|X[k]|
-0.5
0
0.5
k [frequency index]



Published with MATLAB® R2017b