Table of Contents

Alexander Christenson Matlab Project 7

Continuous Wavelet Transform Main Project File

```
close all; clear all
```

Complex and Real Morlets

```
% Display the wavelet making function
type('makeMorlet.m')

% Setting up the variables for testing function
fs = 200; % Sampling frequency
t = [-2:1/fs:2]; % Time vector
s = 1; % Using a scaling factor of one until I figure out how to
 optimize
f0 = 5; % Frequency of exponential
ncycles = 8; % Num cycles for gaussian to decay

% Create the scaled mother wavelet
morletMother = makeMorlet(t,s,f0,ncycles);

% Plot the real and imaginary parts of the wavelet over time
figure
plot(t,real(morletMother),t,imag(morletMother))
title('Morlet Wavelet vs. Time')
ylabel('PSI')
xlabel('Time (sec)')
legend('Real','Imaginary')
axis([-1 1 -1 1])
% Depending on the colors, very trippy looking

% Get the continuous time fourier transform
[MorletMother,fMorlet] = ctft(morletMother,fs);
magMorlet = abs(MorletMother); % Get the magnitude of freq resp

% Plot the magnitude of frequency response
figure
plot(fMorlet,magMorlet)
title('Magnitude of Frequency Response of Morlet Mother Function')
ylabel('Magnitude')
```

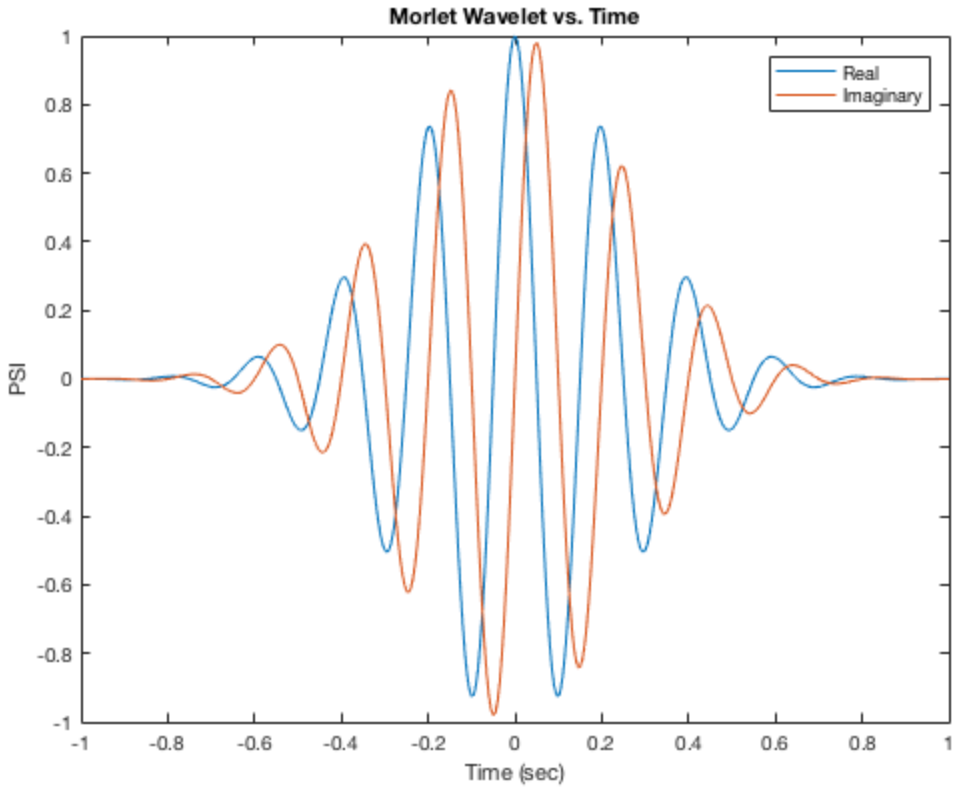
```
xlabel('Frequency (Hz)')
axis([-10 10 -1 1])

% Do the same for the frequency response of the real part of the
 signal
[RealMorletMother,fRealMorlet] = ctft(real(morletMother),fs);
magRealMorlet = abs(RealMorletMother);
figure
plot(fRealMorlet,magRealMorlet)
title('Magnitude of Frequency Response of Morlet Mother Function,
 Real')
ylabel('Magnitude')
xlabel('Frequency (Hz)')
axis([-10 10 -1 1])

%
 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% Discuss
%
 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% The original mother wavelet peaks at f0 = 5 Hz only on the positive
 side
% where the fft of the real only part of the signal has both the
 positive
% and negative 5 Hz peaks. This makes sense because the real only part
 is
% going to be even about the y axis. (for positive and negative
% frequencies). Basically the wavelet takes and shifts a gaussian in
% frequency, defined by sigma, by f0/s. This explains the peak at f =
 5 Hz
% and the symmetry for the real fourier transform
%
 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
%
 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

% Because the shift is defined by f0/s, the center frequency be set by
% s = f0/desired. So for f0 = 5, desired = 30 then s = 1/6. For
 desired
% frequency = 60, s = 1/12.

% Plotting these desired frequency wavelets.
s30 = 1/6;
s60 = 1/12;

morletMother30 = makeMorlet(t,s30,f0,ncycles);
morletMother60 = makeMorlet(t,s60,f0,ncycles);

% Plot the real parts of these guys
figure
plot(t,real(morletMother),t,real(morletMother30),t,real(morletMother60))
ylabel('PSI')
xlabel('Time (sec)')
legend('5 Hz','30 Hz','60 Hz');
```

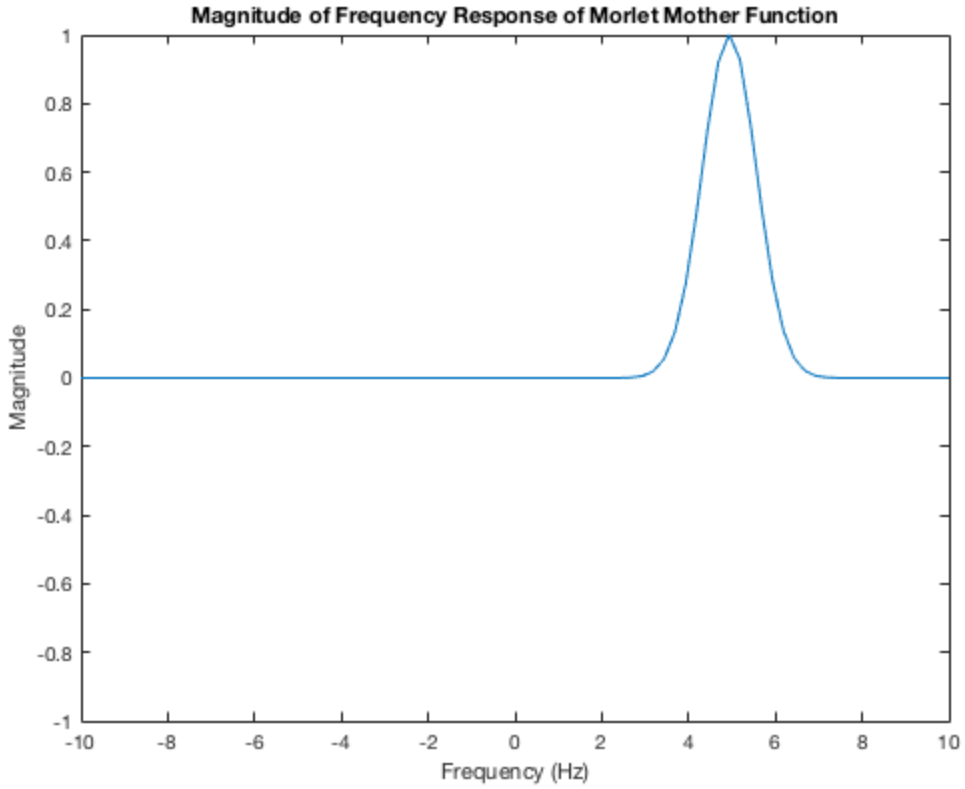
```
title('Real Parts of Wavelets Scaled to Center at Specific
 Frequencies')
axis([-1 1 -1 1])

% Plot their Fourier transforms
[Morlet30,fMorlet30] = ctft(morletMother30,fs);
[Morlet60,fMorlet60] = ctft(morletMother60,fs);

figure
plot(fMorlet,abs(MorletMother),fMorlet30,abs(Morlet30),...
    fMorlet60,abs(Morlet60))
ylabel('Magnitude')
xlabel('Frequency (Hz)')
legend('5 Hz','30 Hz','60 Hz');
title('Freq Resp of Wavelets Scaled to Center at Specific
 Frequencies')
axis([0 90 0 1])

% Apply the normalization factor to the wavelets
scaledMorlet5 = MorletMother/sqrt(abs(s));
scaledMorlet30 = Morlet30/sqrt(abs(s30));
scaledMorlet60 = Morlet60/sqrt(abs(s60));

figure
plot(fMorlet,abs(scaledMorlet5),fMorlet30,abs(scaledMorlet30),...
    fMorlet60,abs(scaledMorlet60))
xlabel('Frquency (Hz)')
ylabel('Magnitude')
title('Scaled Freq Response of Scaled Morlet Wavelets, Normalized')
legend('5 Hz','30 Hz','60 Hz')
axis([0 90 0 1])

%
 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% Discuss
%
 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% This scaling increases the magnitude to compensate for the reduction
 at
% higher frequencies, the amount that it is scaled is proportional to
 the
% scaling factor for the input, which spreads out the function
 according to
% the scaling factor.
%
 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
%
 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

% Make three more wavelets except with ncycles = 16
ncycles = 16;

s5 = 1;
```

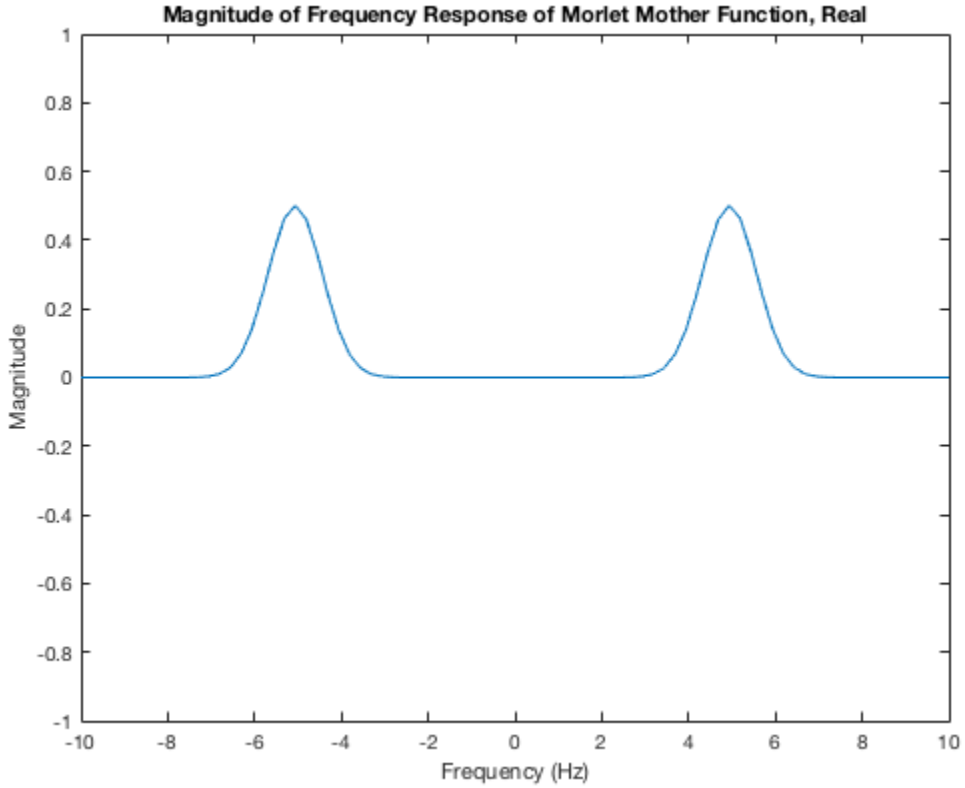
```
% Make the wavelets
morlet5 = makeMorlet(t,s5,f0,ncycles);
morlet30 = makeMorlet(t,s30,f0,ncycles);
morlet60 = makeMorlet(t,s60,f0,ncycles);

% Plot the real part in the time domain
figure
plot(t,real(morlet5),t,real(morlet30),t,real(morlet60))
xlabel('Time (sec)')
ylabel('PSI')
title('Real Values of Wavelet Functions, ncycles = 16')
legend('5 Hz','30 Hz','60 Hz')

% Get the fourier transform
[Morlet5,fMorlet5] = ctft(morlet5,fs);
[Morlet30,fMorlet30] = ctft(morlet30,fs);
[Morlet60,fMorlet60] = ctft(morlet60,fs);

% Normalize the frequency response
nMorlet5 = Morlet5/sqrt(abs(s5));
nMorlet30 = Morlet30/sqrt(abs(s30));
nMorlet60 = Morlet60/sqrt(abs(s60));

% Plot the frequency response
figure
plot(fMorlet5,abs(nMorlet5),fMorlet30,abs(nMorlet30),...
    fMorlet60,abs(nMorlet60))
title('Normalized Frequency Response Morlet Wavelets, ncycles = 16')
xlabel('Frequency (Hz)')
ylabel('Mag')
legend('5 Hz','30 Hz','60 Hz')
axis([0 90 0 2])

%
 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% Discuss
%
 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% In the time domain increasing the number of cycles for the gaussian
 in
% the wavelet to die widens the impulse response of the wavelet. This
 makes
% sense intuitively because you are literally increasing the number of
% cycles it takes for one of the components in the time domain
 definitin of
% the wavelet to die off. For the frequency domain the magnitude of
 the
% signal doubles. This makes sense because you are decreasing the
 standard
% deviation of the gaussian function which defines the frequency
 response
% of the wavelet so you are going to be increasing its magnitude.
%
 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

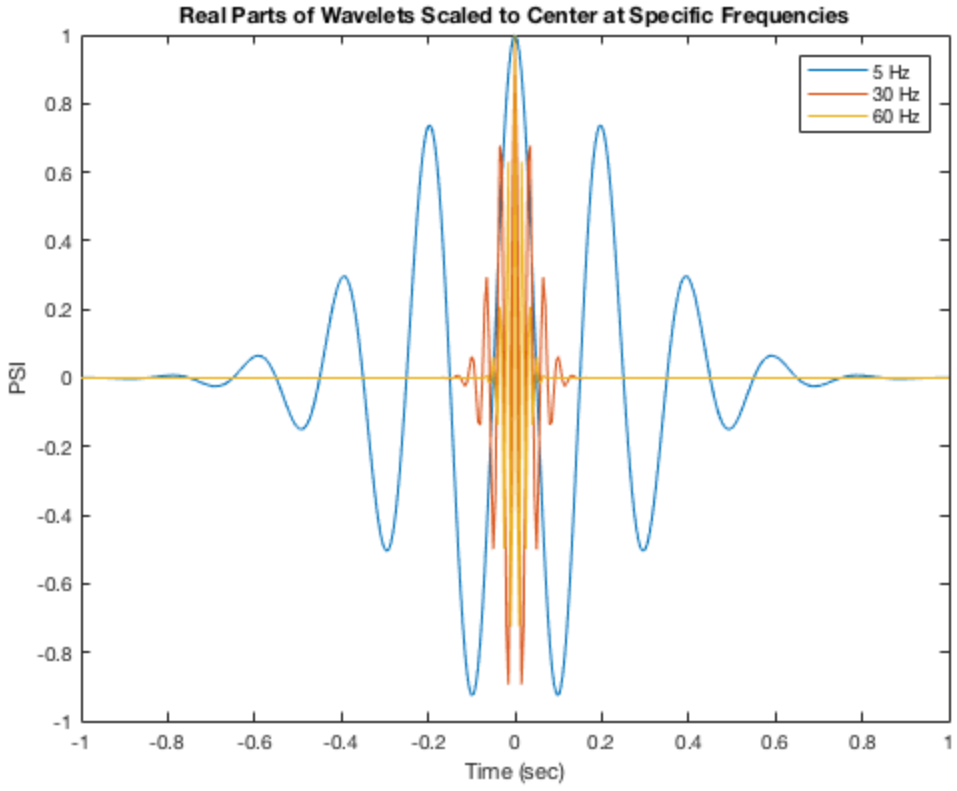
```
%
 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

% Reset ncycles to 8
ncycles = 8;

% Set up vector for s = f0/desired
fmin = 5;
fmax = 100;
nSamples = 101;
desired = linspace(fmin,fmax,nSamples);
sVec_new = f0./desired;
sVec = linspace(f0/fmin,f0/fmax,nSamples);

% Create a smaller time vector
tp = [-0.5:1/fs:0.5];

% Initialize psiM matrix
psiM = [];
psiM2 = [];

% Create a matrix of scaled wavelet functions
for idx = 1:nSamples
    psiM(:,idx) = makeMorlet(tp,sVec(idx),f0,ncycles);
    psiM2(:,idx) = makeMorlet(tp,sVec_new(idx),f0,ncycles);
end

% Checked the matrix in surf, looks really cool, and definitely checks
 out,
% as the frequency we're aiming for increases the wavelet gets
 skinnier in
% time.


function psi = makeMorlet(t,s,f0,ncycles)
% function psi = makeMorlet(t,s,f0,ncycles)
%
% This function takes in the set of inputs and produces a Morlet
 wavelet
% for the specified inputs using sigma = ncycles/(2*pi*f0)
%
% Inputs
%   t = input time vector
%   s = scaling factor
%   f0 = frequency of exponential
%   ncycles = number of cycles for gaussian to decay at f0
% Outputs
%   psi = scaled mother wavelet

% Calculate sigma
sigma = ncycles/(2*pi*f0);

% Calculate the wavelet function
psi = exp(1i*2*pi*f0*t/s).*exp(-1*t.^2/(2*s^2*sigma^2));
```

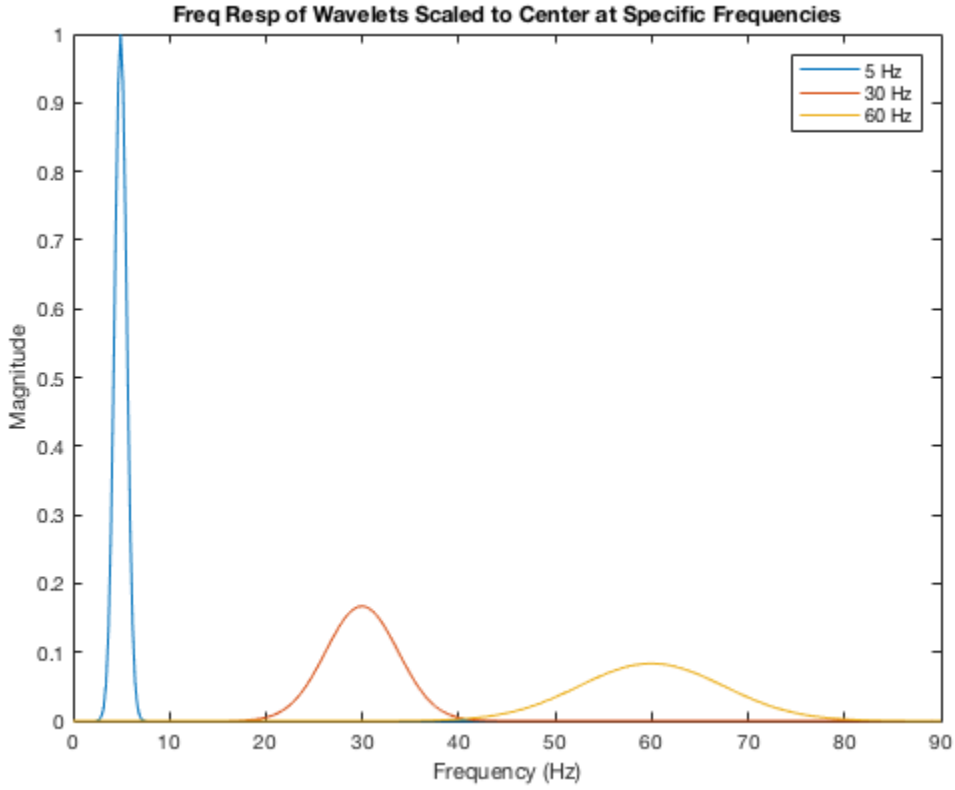
```
return
```

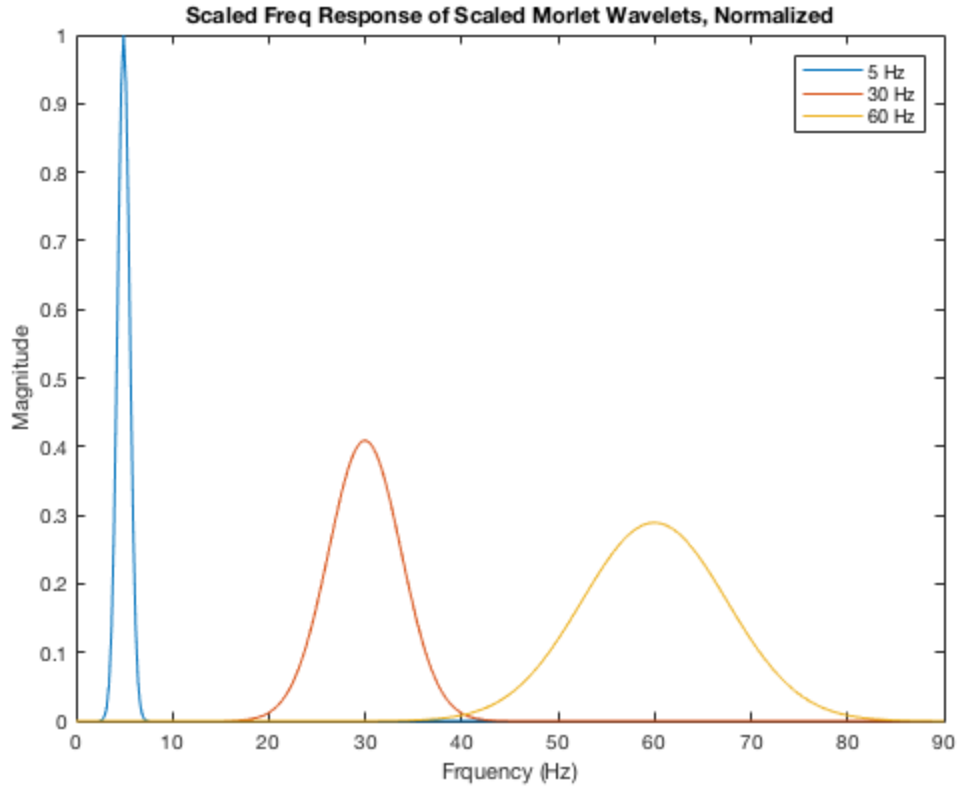

Morlet Wavelet vs. Time
Real
Imaginary
PSI
Time (sec)
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1

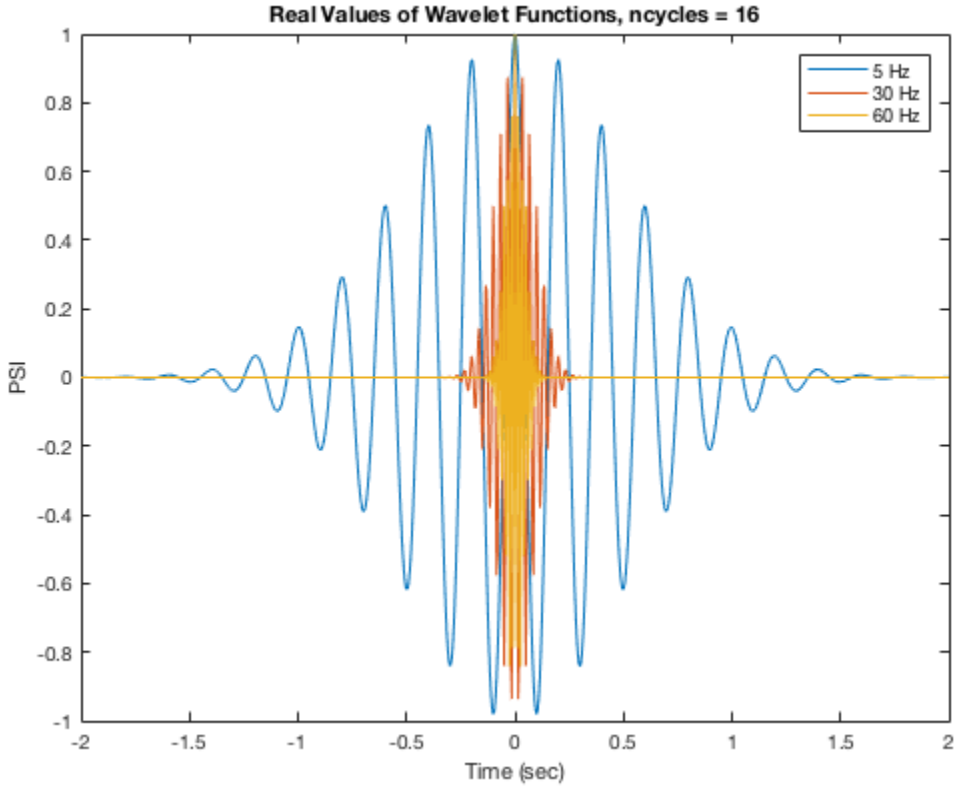

Magnitude of Frequency Response of Morlet Mother Function
Magnitude
Frequency (Hz)
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-10
-8
-6
-4
-2
0
2
4
6
8
10

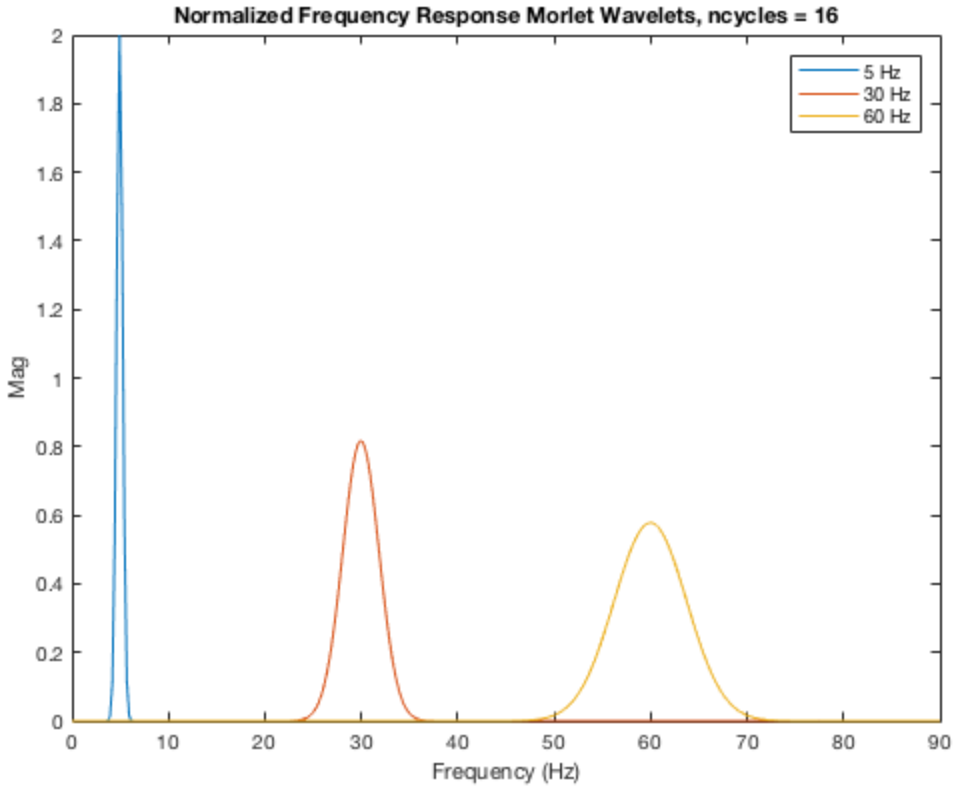

Magnitude of Frequency Response of Morlet Mother Function, Real
Magnitude
Frequency (Hz)
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-10
-8
-6
-4
-2
0
2
4
6
8
10


Real Parts of Wavelets Scaled to Center at Specific Frequencies
5 Hz
30 Hz
60 Hz
PSI
Time (sec)
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1


Freq Resp of Wavelets Scaled to Center at Specific Frequencies
5 Hz
30 Hz
60 Hz
Magnitude
Frequency (Hz)
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
10
20
30
40
50
60
70
80
90


Scaled Freq Response of Scaled Morlet Wavelets, Normalized
5 Hz
30 Hz
60 Hz
Magnitude
Frquency (Hz)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
10
20
30
40
50
60
70
80
90


Real Values of Wavelet Functions, ncycles = 16
5 Hz
30 Hz
60 Hz
PSI
Time (sec)
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2

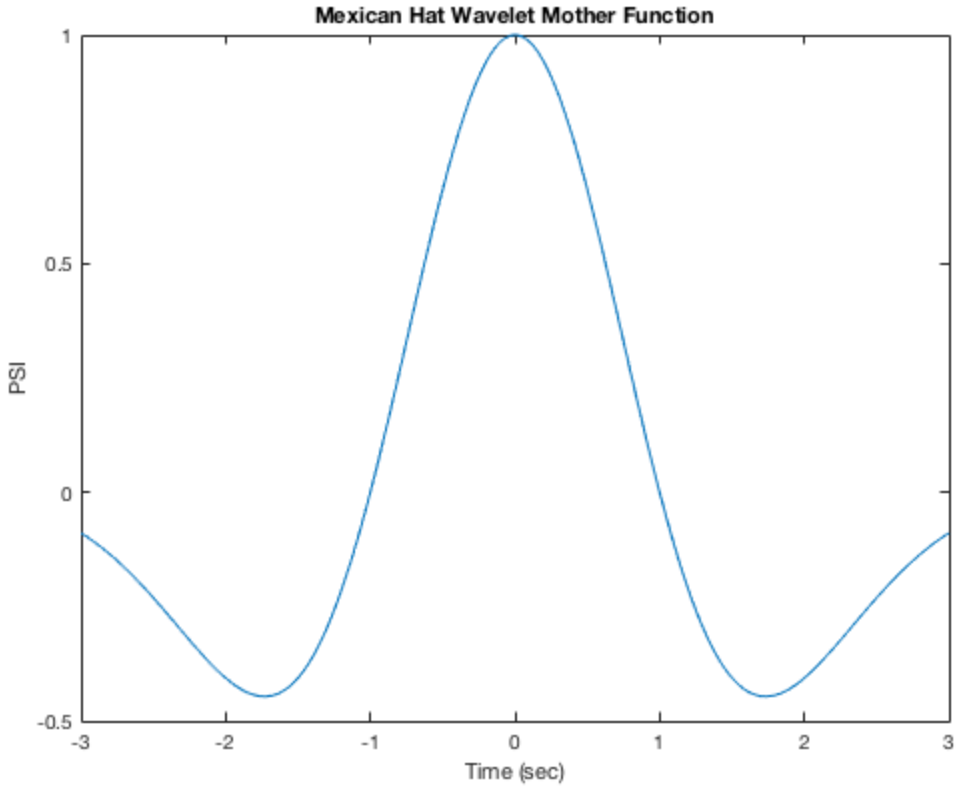


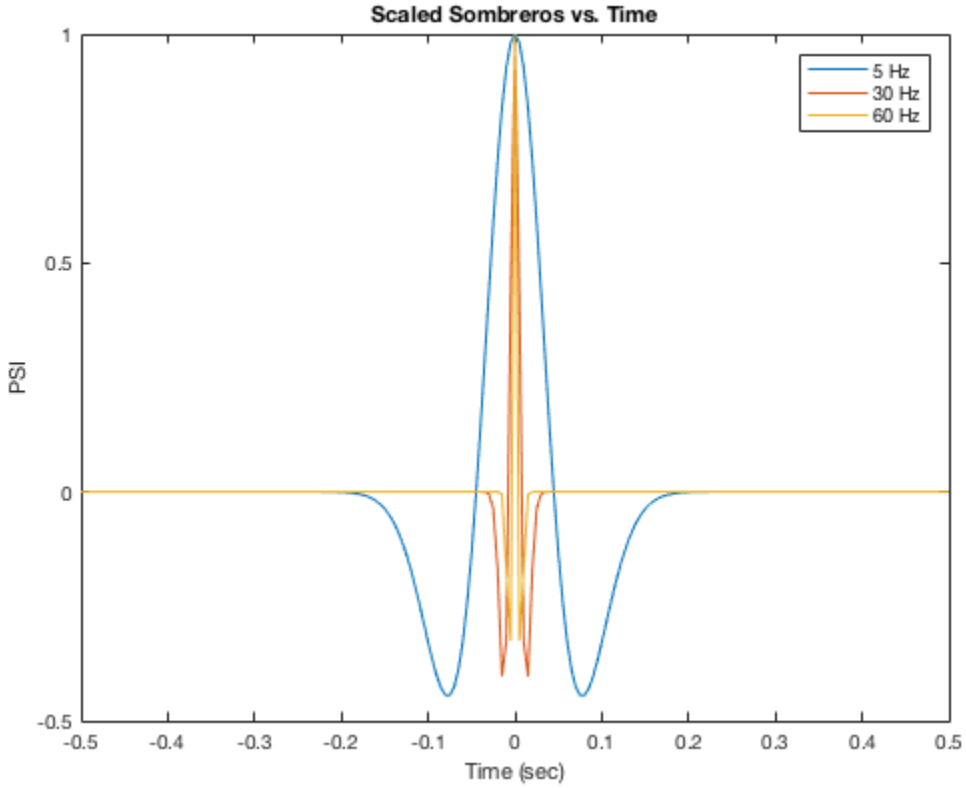
Sombrero/Mexican Hat Wavelet

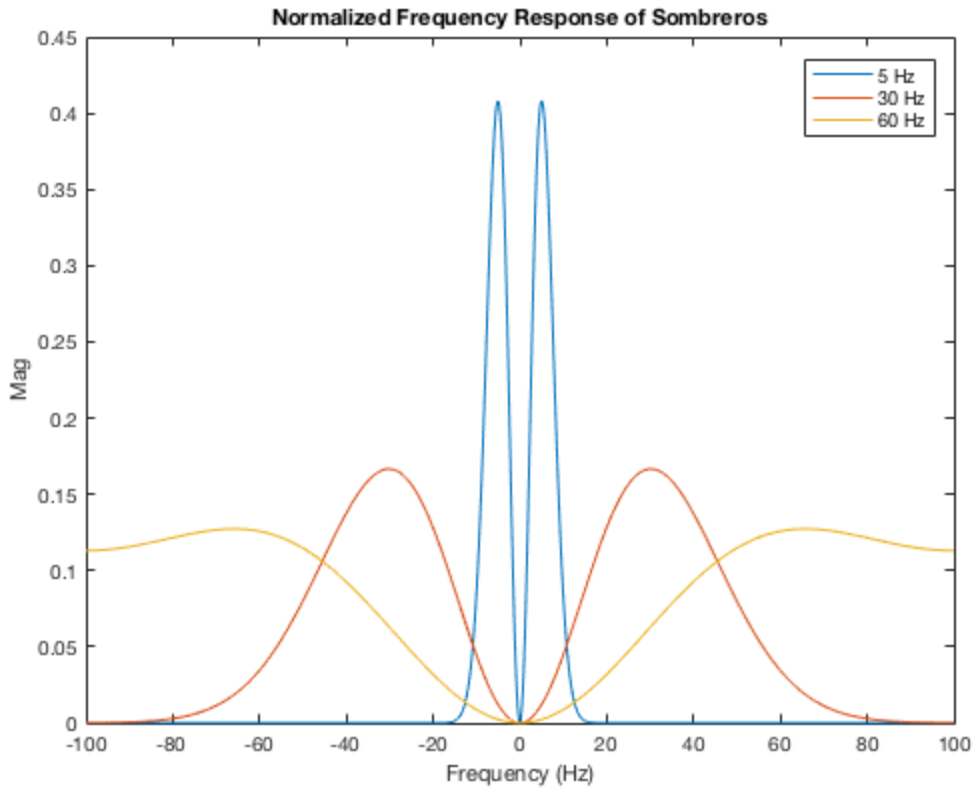
```
% For publishing
type('makeSombrero.m')

% Initialize the given constants for this section
t = [-3:1/fs:3];

% Create and plot the wavelet
somb = makeSombrero(t);
figure
plot(t,somb)
xlabel('Time (sec)')
ylabel('PSI')
title('Mexican Hat Wavelet Mother Function')
% Nice, this look like what we're expecting

% Scaling the mother wavelet
f1 = 0.2237;
s5 = f1/5;
s30 = f1/30;
s60 = f1/60;

% Creating the new wavelets
somb5 = makeSombrero(t,s5);
```

```
somb30 = makeSombrero(t,s30);
somb60 = makeSombrero(t,s60);

% Plot
figure
plot(t,real(somb5),t,real(somb30),t,real(somb60))
title('Scaled Sombreros vs. Time')
xlabel('Time (sec)')
ylabel('PSI')
legend('5 Hz','30 Hz','60 Hz')
axis([-0.5 0.5 -0.5 1])

% Get the fourier trans
[Somb5,fSomb5] = ctft(somb5,fs);
[Somb30,fSomb30] = ctft(somb30,fs);
[Somb60,fSomb60] = ctft(somb60,fs);

% Normalize
nSomb5 = Somb5/sqrt(abs(s5));
nSomb30 = Somb30/sqrt(abs(s30));
nSomb60 = Somb60/sqrt(abs(s60));

% Plot
figure
plot(fSomb5,abs(nSomb5),fSomb30,abs(nSomb30),fSomb60,abs(nSomb60))
title('Normalized Frequency Response of Sombreros')
xlabel('Frequency (Hz)')
ylabel('Mag')
legend('5 Hz','30 Hz','60 Hz')

%
 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% Discuss
%
 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% The Sombrero wavelet gets really really good time resolution, but as
 the
% targeted frequency becomes higher the wavelet loses a lot of its
% frequency resolution. You can see this how the graphs in the time
 domain
% become more scrunched up as the frequency target is increased and
 how at
% the same time the frequency domain gets spread out as the center
 freq is
% increased.
%
 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
%
 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

% Set up a vector and matrix for CWT
sVec2_new = f1./desired;
sVec2 = linspace(f1/fmin,f1/fmax,nSamples);
psiS = [];
```

```
psiS2 = [];

% Get the matrix of scaled wavelets
for idx = 1:nSamples
    psiS(:,idx) = makeSombrero(tp,sVec2(idx));
    psiS2(:,idx) = makeSombrero(tp,sVec2_new(idx));
end


function psi = makeSombrero(t,s)
% function psi = makeSombrero(t,s)
%
% This function creates a mexican hat wavelet with an optional scaling
% factor
%
% Inputs:
%   t = time vector
%   s = scaling factor
% Outputs:
%   psi = wavelet
%

if nargin == 2
    t = t/s;
end

psi = (1-t.^2).*exp(-1*t.^2/2);

return
```


Mexican Hat Wavelet Mother Function
PSI
Time (sec)
1
0.5
0
-0.5
-3
-2
-1
0
1
2
3


Scaled Sombreros vs. Time
PSI
Time (sec)
5 Hz
30 Hz
60 Hz
1
0.5
0
-0.5
-0.5
-0.4
-0.3
-0.2
-0.1
0
0.1
0.2
0.3
0.4
0.5



CWT Using Morlet Transform

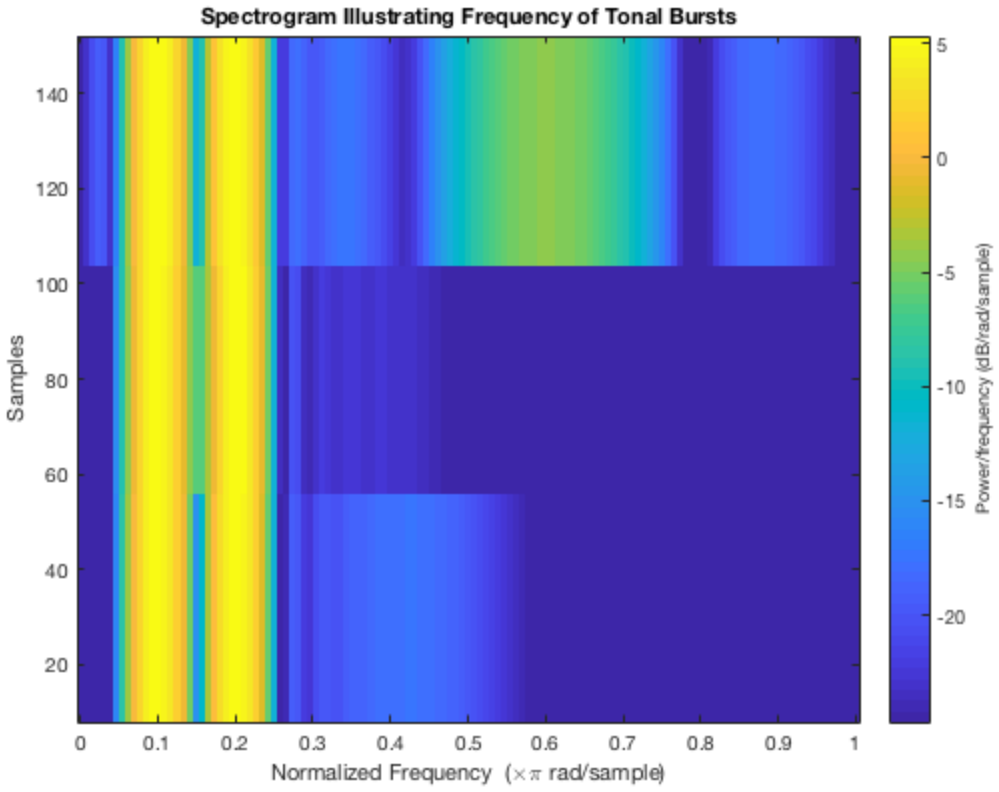
```
type('myCWT.m')

% Load the data
[data,Fs] = demoData;

% Create a spectrogram that lets you accurately see the two tone
 bursts
L = 64;
noverlap = L/4;
figure
spectrogram(data,hann(L),noverlap);
% Note that you can see the spikes in the frequency domain really
 well, but
% it is much hard to see the tone duration
lim = caxis;
lim(1) = lim(2)-30;
caxis(lim)
title('Spectrogram Illustrating Frequency of Tonal Bursts')

% Creating a spectrogram to see the time duration of the tonal bursts
L = 16;
noverlap = L/4;
figure
```

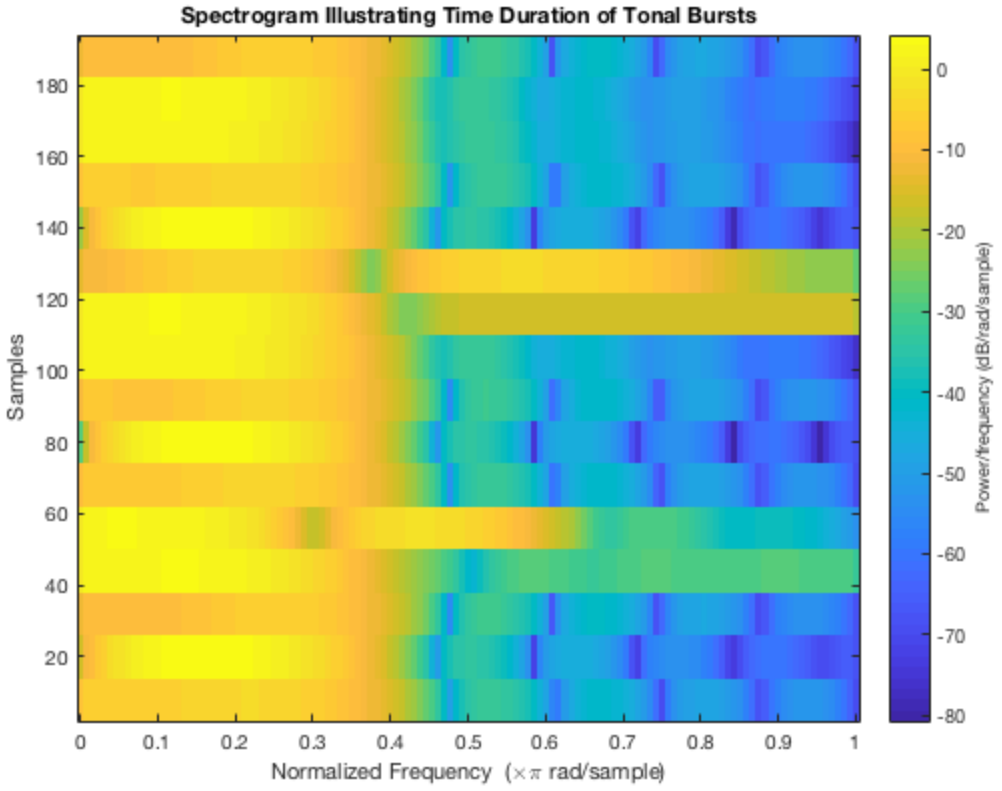
```
spectrogram(data,hann(L),noverlap);
title('Spectrogram Illustrating Time Duration of Tonal Bursts')

%
 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% Discuss
%
 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% There is a time and a frequency tradeoff for spectrogram resolution.
% Since for the first one we want a better frequency resolution we
 make a
% spectrogram using a large window (giving us a smaller frequency
 window)
% and this give us a good frequency resolution. To see the time
 duration of
% the pulses we want to make the window skinnier so that our time
% resolution is better, however, this comes at the cost of our
 frequency
% resolution. This can be very clearly seen from the graphs. To
 illustrate
% this point further I kept the percent overlap and window type the
 same
% (25%).
%
 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
%
 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

% Processing using the CWT
CWTm = myCWT(data,Fs,psiM,sVec);
nCWTm = abs(CWTm)/max(abs(CWTm));
nCWTm_dB = mag2db(nCWTm);

% Plot
figure
imagesc(tp,sVec,nCWTm_dB)
title('Scalagram of Data')
xlabel('Time (sec)')
ylabel('Scale Value')
colorbar

%
 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% Discuss
%
 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% There's lots of evidence of tones in the image, but not great
 evidence of
% the burst nature of the tones. Maybe I'm not implementing the system
% quite right, but it could also be because that the time resolution
 for
% the morlet wavelet isnt high enough to be able to see the bursts
 that are
% happening over the time that they are evaluated at.
```

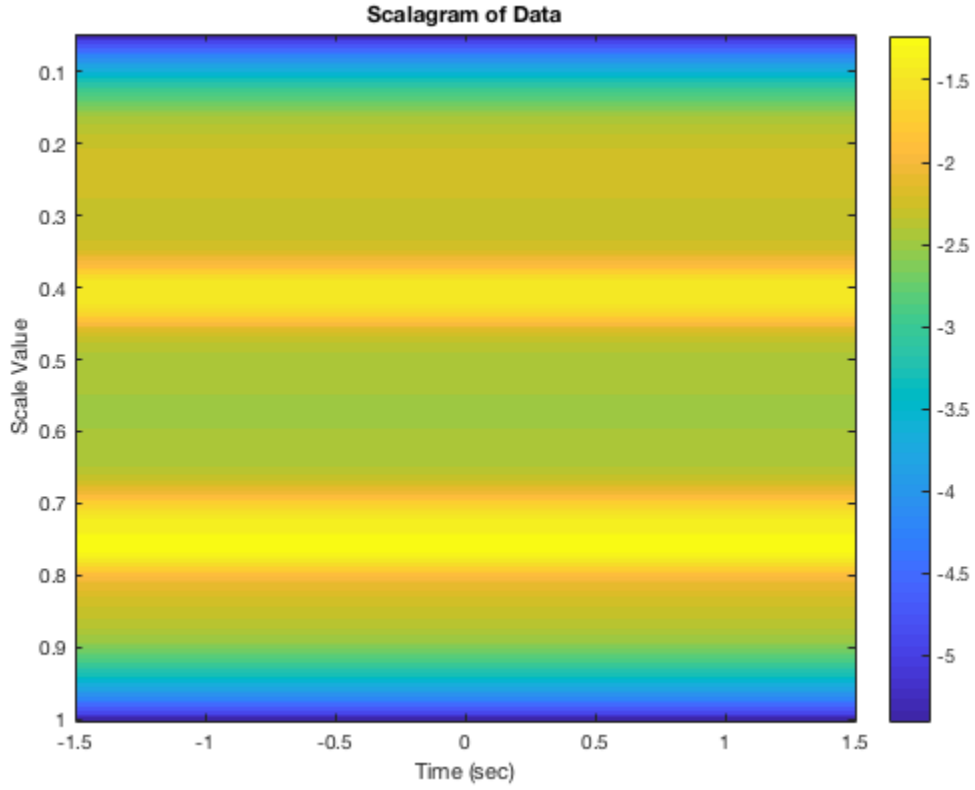
```
%
% The previous paragraph is talking about the nonlinear scale
 frequency
% relationship because I messed up earlier. When I fixed it this came
 out
% to be pretty difficult to interpret this way and the tonal
 components are
% not super clear (or nearly as clear as the nonlinear version)
%
 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
%
 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

% Plot the scale vs frequency
figure
plot(desired,sVec)
title('Scale vs. Frequency, Morlet')
xlabel('Frequency')
ylabel('Scale')
% Whoops, I accidentally did the scale calculation the second way the
 first
% time so I just went back and renamed the original calculation I did
 to
% sVec_new, this represents the evenly spaced frequency for the
 scaling
% values. I'll go back and fix my discussion but I don't think this
 should
% have caused too many errors.

% Plotting the nonlinear fixed version
CWTm = myCWT(data,Fs,psiM2,sVec_new);
nCWTm = abs(CWTm)/max(abs(CWTm));
nCWTm_dB = mag2db(nCWTm);

% Plot
figure
imagesc(tp,sVec_new,nCWTm_dB)
title('Scalagram of Data, nonlinear scale-frequency')
xlabel('Time (sec)')
ylabel('Scale Value')
colorbar

%
 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% Discuss
%
 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% As I said earlier, the tones are much clearer using the nonlinear
 scale
% frequency relationship because now the frequencies are evenly space
 and
% the tonal resolution is much better. This resolution looks much
 better
```

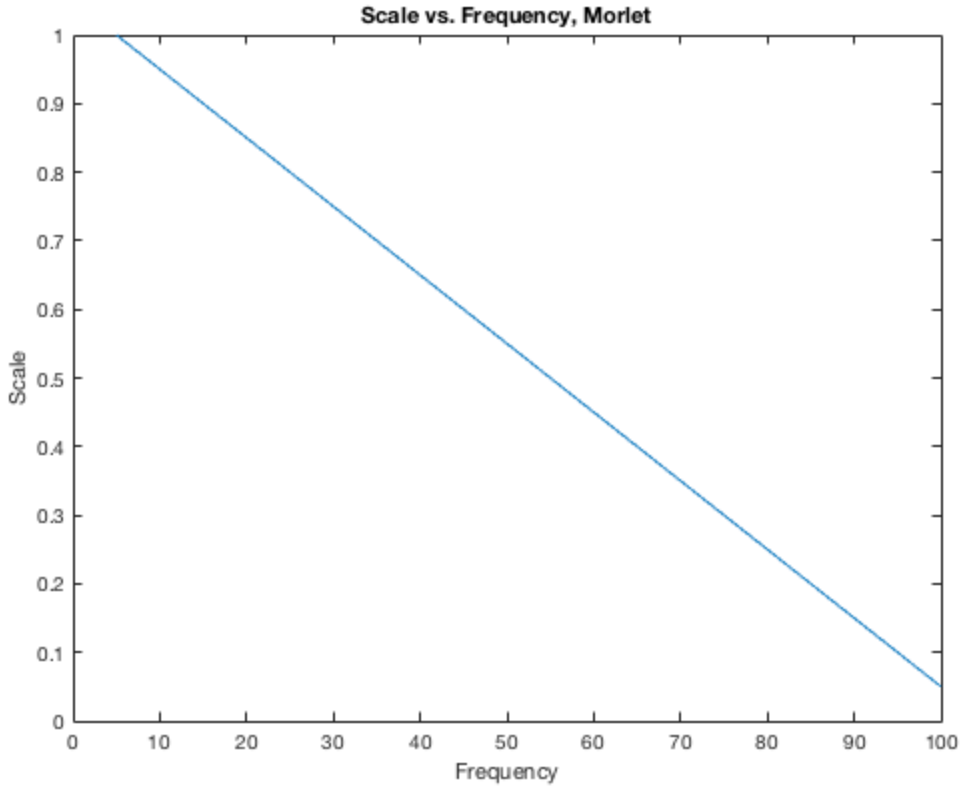
```
% than the spectrograms too, I think because of its more dynamic
 nature
% (and because wavelets are way cooler than windowing)
%
 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
%
 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

% Repeat last step for real
CWTm = real(CWTm);
nCWTm = abs(CWTm)/max(abs(CWTm));
nCWTm_dB = mag2db(nCWTm);

% Plot
figure
imagesc(tp,sVec_new,nCWTm_dB)
title('Scalagram of Data, nonlinear scale-frequency, Real')
xlabel('Time (sec)')
ylabel('Scale Value')
colorbar

%
 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% Discuss
%
 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% This real result has periodic (in frequency) blank spots that are
 missing
% from the graph. I think this is because of the symmetry in the
 frequency
% domain of the real version of the wavelet, so that when you perform
 all
% the convolutions with the data across the set there ends up being
% interference.
%
 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
%
 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~


function C = myCWT(data,fs,psi,scaleVec)
% function C = myCWT(data,fs,psi,scaleVec)
% Inputs:
%    'data' is vector of input data, length numSamps
%    fs is sampling rate of 'data' vector
%    psi is precomputed matrix of wavelet shapes, size
%        L x numScales, where L is the length of each wavelet
%    scaleVec is vector of scales used to compute psi, length
 numScales
% Output:
%    C:  numSamps x numScales CWT output

numScales = length(scaleVec);
% Ask Prof. Tracey about this in office hours
```

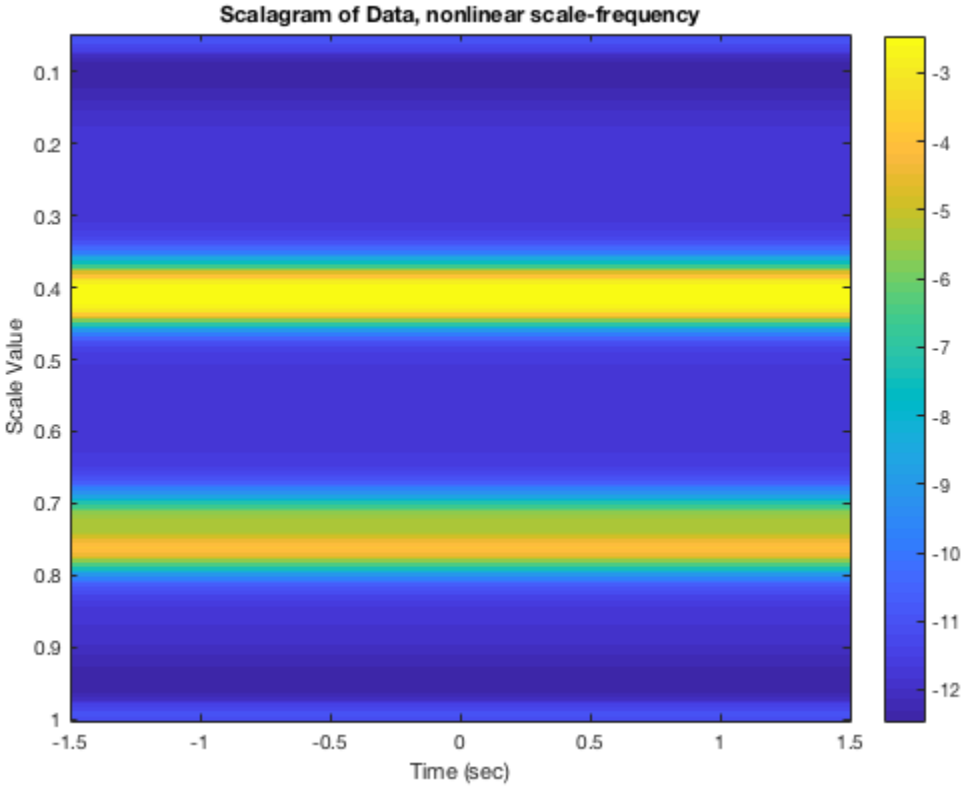
```
d = size(psi);
L = d(1);

% file loops over scales.  At each scale, convolves 'data' with a
% the apprpriate column of psi, and stores result in correspoding
% column of C

for ss = 1:numScales
    h = conj(psi(:,ss))/(fs*scaleVec(ss));
    C(:,ss) = conv(data,h,'same');
end
```

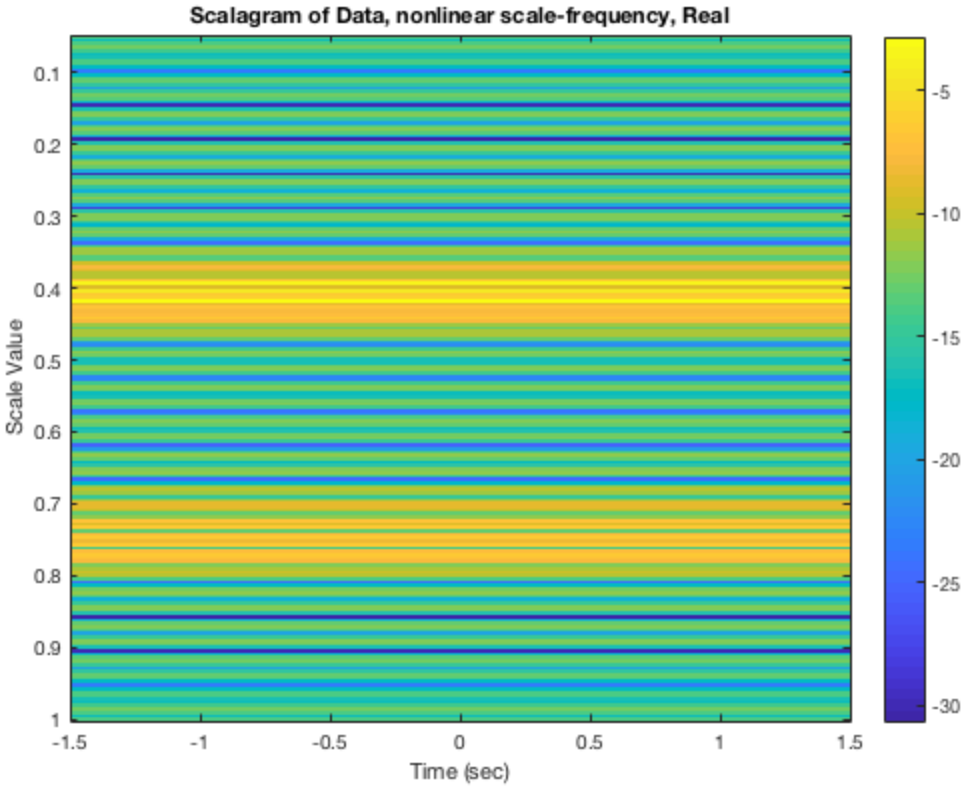



Spectrogram Illustrating Time Duration of Tonal Bursts
Samples
Normalized Frequency (×π rad/sample)
Power/frequency (dB/rad/sample)


Scalagram of Data
Scale Value
Time (sec)


Scale vs. Frequency, Morlet
Scale
Frequency
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
10
20
30
40
50
60
70
80
90
100

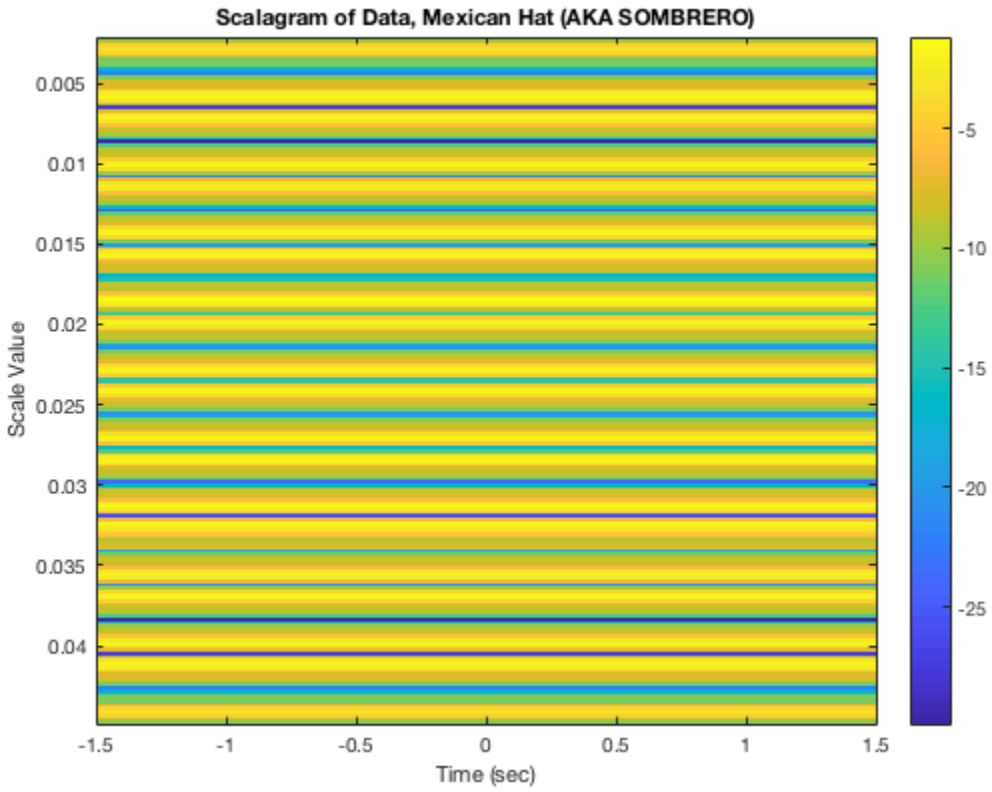

Scalagram of Data, nonlinear scale-frequency
Scale Value
Time (sec)
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
-1.5
-1
-0.5
0
0.5
1
1.5
-3
-4
-5
-6
-7
-8
-9
-10
-11
-12



CWT With a Sombrero

```
CWTm = myCWT(data,Fs,psiS,sVec2);
nCWTm = abs(CWTm)/max(abs(CWTm));
nCWTm_dB = mag2db(nCWTm);

% Plot
figure
imagesc(tp,sVec2,nCWTm_dB)
title('Scalagram of Data, Mexican Hat (AKA SOMBRERO)')
xlabel('Time (sec)')
ylabel('Scale Value')
colorbar

%
 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% Discuss
%
 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% You don't get very good time or frequency resolution using this!!
 The
% scalagram is pretty much useless to the eye.
% The low frequency tones, at least to me, look the same as the higher
% frequency tones, but this is slightly counter to the logic from
 looking
```

```
% at the various frequency responses to the mexican hat wavelet
 filters at
% different scales. This could just be a result of the frequency
 smudging
% and what the frequency of the data looks like.
%
 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
%
 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

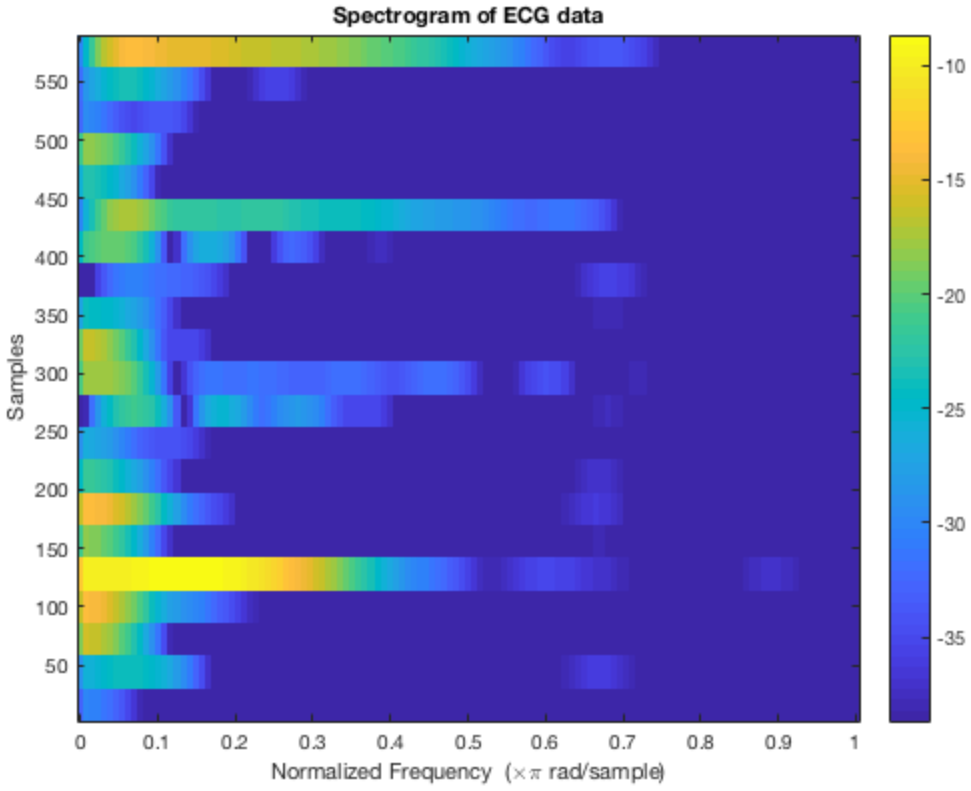


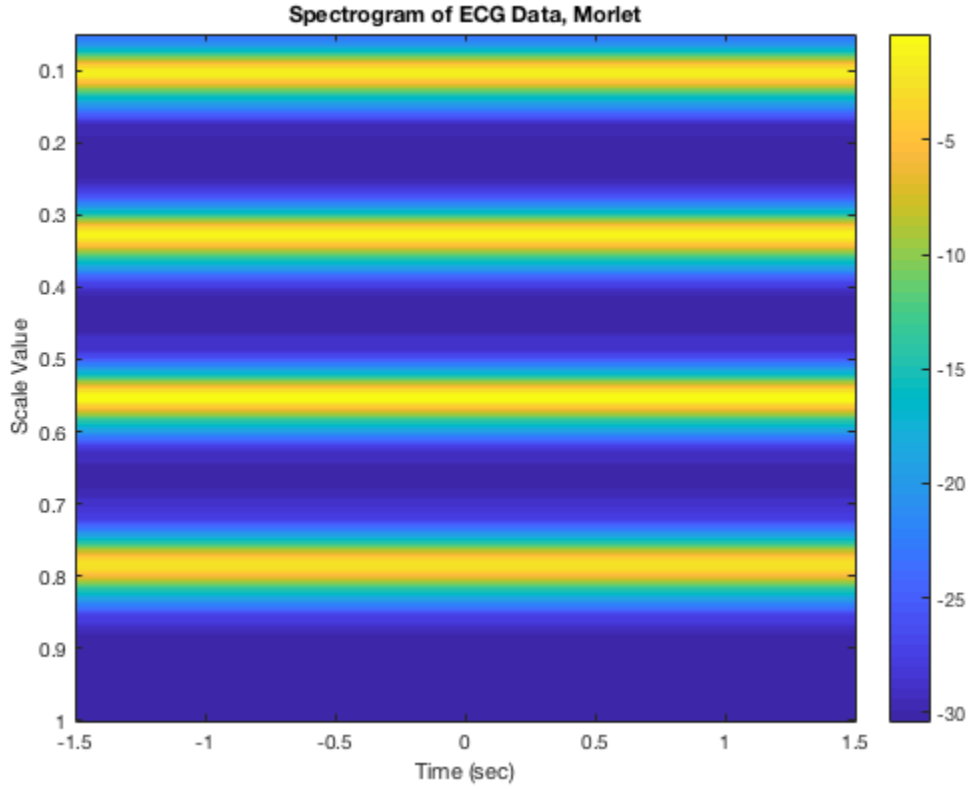
Computing and plotting the CWT - ECG data

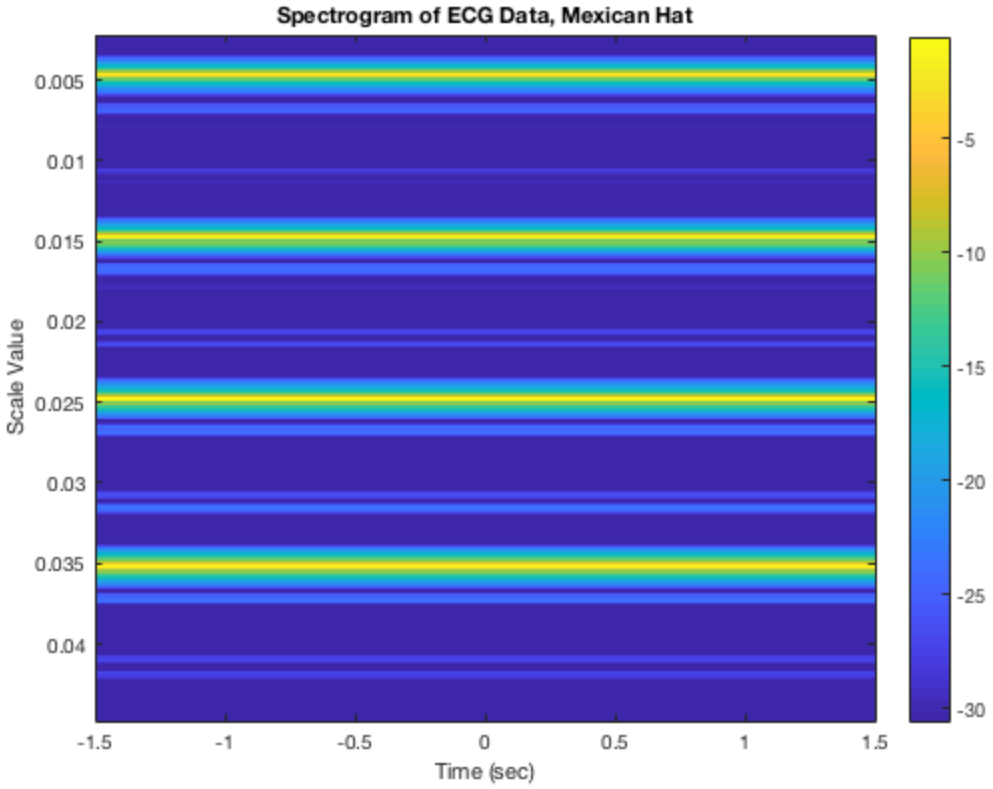
```
% Load the ECG data
load ecgData.mat % Loads data and fs
figure
spectrogram(data,hann(32),4)
title('Spectrogram of ECG data')
lim = caxis;
caxis([lim(2)-30 lim(2)]);
colorbar

% Using the Morlet Wavelet
CWTm = myCWT(data,fs,psiM2,sVec_new);
nCWTm = abs(CWTm)/max(abs(CWTm));
nCWTm_dB = mag2db(nCWTm);
```

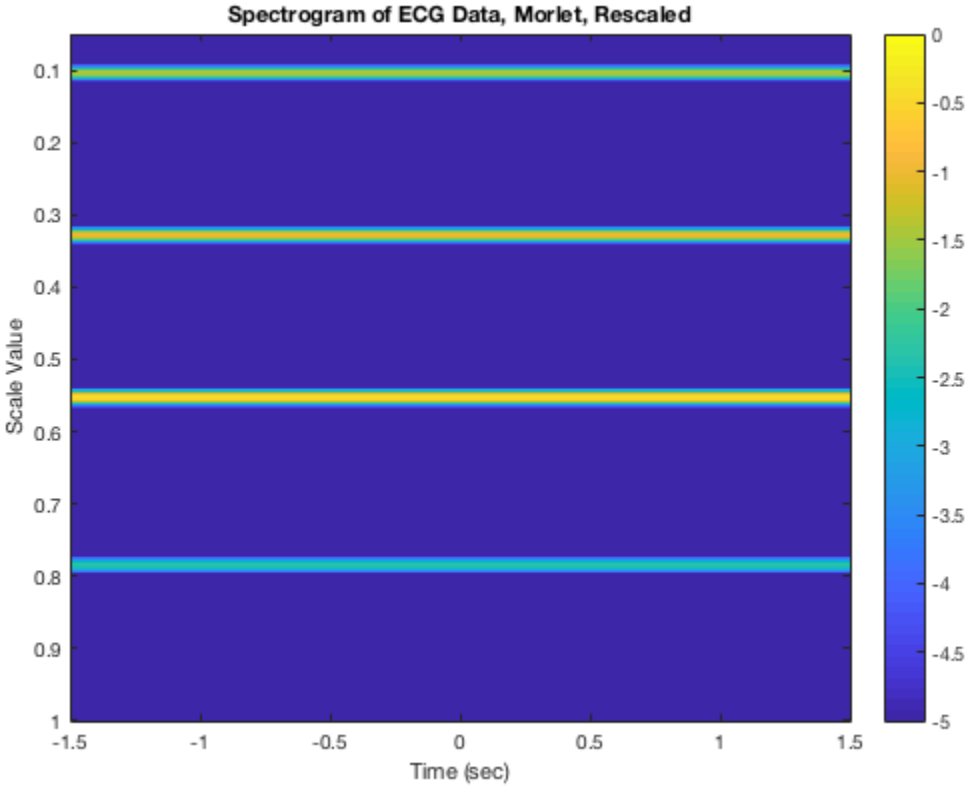
```
% Plot
figure
imagesc(tp,sVec_new,nCWTm_dB)
title('Spectrogram of ECG Data, Morlet')
xlabel('Time (sec)')
ylabel('Scale Value')
lim = caxis;
caxis([lim(2)-30 lim(2)]);
colorbar

% Using the Mexican Hat wavelet
CWTmh = myCWT(data,fs,psiS,sVec2);
nCWTmh = abs(CWTmh)/max(abs(CWTmh));
nCWTmh_dB = mag2db(nCWTmh);

% Plot
figure
imagesc(tp,sVec2,nCWTmh_dB)
title('Spectrogram of ECG Data, Mexican Hat')
xlabel('Time (sec)')
ylabel('Scale Value')
lim = caxis;
caxis([lim(2)-30 lim(2)]);
colorbar

%
 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% Discuss
%
 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% When you look at the spectrogram using the Morlet wavelet of the ECG
 data
% you get much better information about the frequency domain of the
 pulse,
% but you lose a lot of information about the time domain compared to
 the
% spectrogram. I think that the unfiltered spectrogram gives you the
 most
% information about the time because the wavelet filters will spread
 the
% signal out over time and cause the overlap to add up.
%
 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
%
 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

% Rescaling the spectrograms
figure
imagesc(tp,sVec_new,nCWTm_dB)
title('Spectrogram of ECG Data, Morlet, Rescaled')
xlabel('Time (sec)')
ylabel('Scale Value')
lim = caxis;
caxis([-5 0]);
```

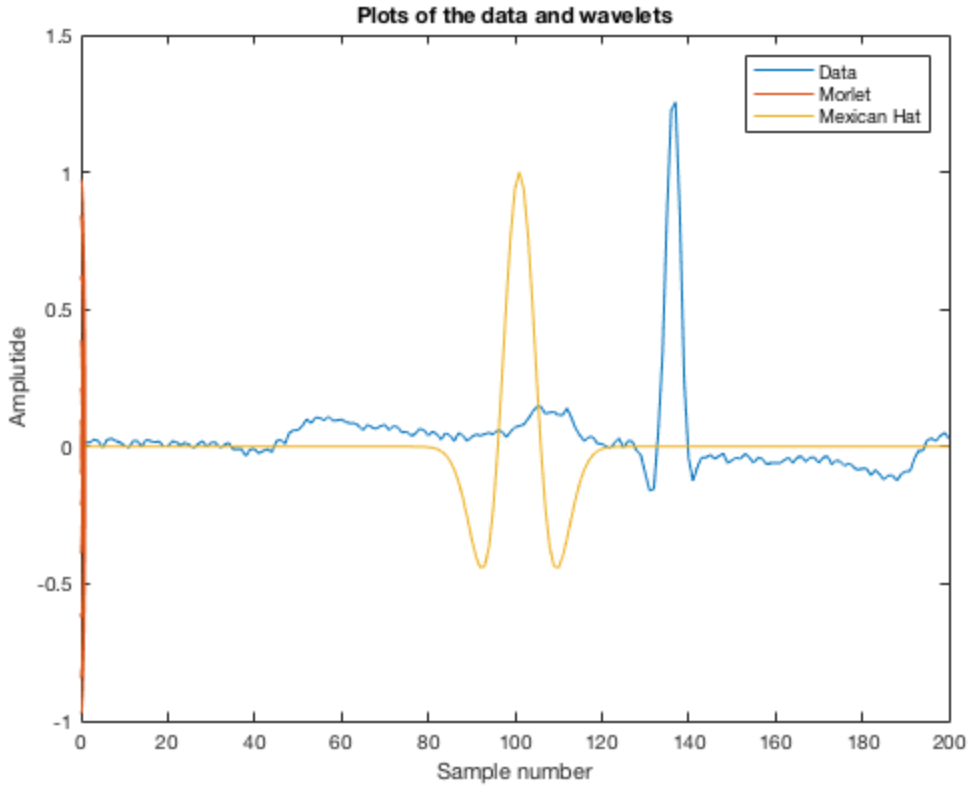
```
colorbar
sChosenM = 0.5519;
idxM = 5; % Find closest match in sVec

figure
imagesc(tp,sVec2,nCWTmh_dB)
title('Spectrogram of ECG Data, Mexican Hat, Rescaled')
xlabel('Time (sec)')
ylabel('Scale Value')
lim = caxis;
caxis([-5 0]);
colorbar
sChosenMH = 0.02476;
idxMH = 48;

% Plot the wavelets at the scale and ecg data
plot(data)
hold on
plot(psiM2(:,idxM))
plot(psiS(:,idxMH))
axis([0 200 -1 1.5])
title('Plots of the data and wavelets')
legend('Data','Morlet','Mexican Hat')
xlabel('Sample number')
ylabel('Amplutide')
% Sorry the plot doesn't look very nice

%
 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% Discuss
%
 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% Notice that when you match the wavelet shape with the expected data
 shape
% you get amazing resolution in frequency in the spectrogram. This is
% probably because when the data overlaps in the windowing you
 maximize the
% amplitude of the output signal, and when the shape maps close to the
% expected data there is not much extra data and not much missed data.
%
 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
%
 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```


Spectrogram of ECG data
Samples
550
500
450
400
350
300
250
200
150
100
50
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Normalized Frequency (×π rad/sample)
-10
-15
-20
-25
-30
-35


Spectrogram of ECG Data, Morlet
Scale Value
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
-1.5
-1
-0.5
0
0.5
1
1.5
Time (sec)
-5
-10
-15
-20
-25
-30


Spectrogram of ECG Data, Mexican Hat
Scale Value
0.005
0.01
0.015
0.02
0.025
0.03
0.035
0.04
-1.5
-1
-0.5
0
0.5
1
1.5
Time (sec)
-5
-10
-15
-20
-25
-30


Spectrogram of ECG Data, Morlet, Rescaled
Scale Value
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
-1.5
-1
-0.5
0
0.5
1
1.5
Time (sec)
0
-0.5
-1
-1.5
-2
-2.5
-3
-3.5
-4
-4.5
-5



Published with MATLAB® R2017b