Ben Postlethwaite 76676063

Table of Contents

Assignment 2

Setup

```
clear all; close all;
loadtools;
```

Select Image

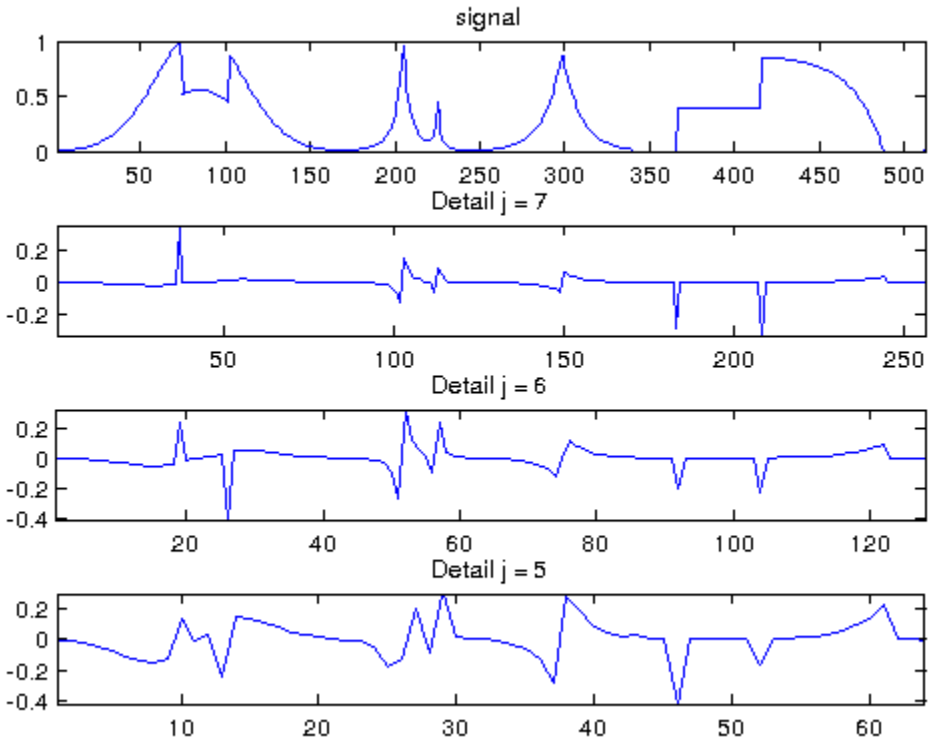
```
name = 'piece-regular';
n = 512;
f = rescale( load_signal(name, n) );
```

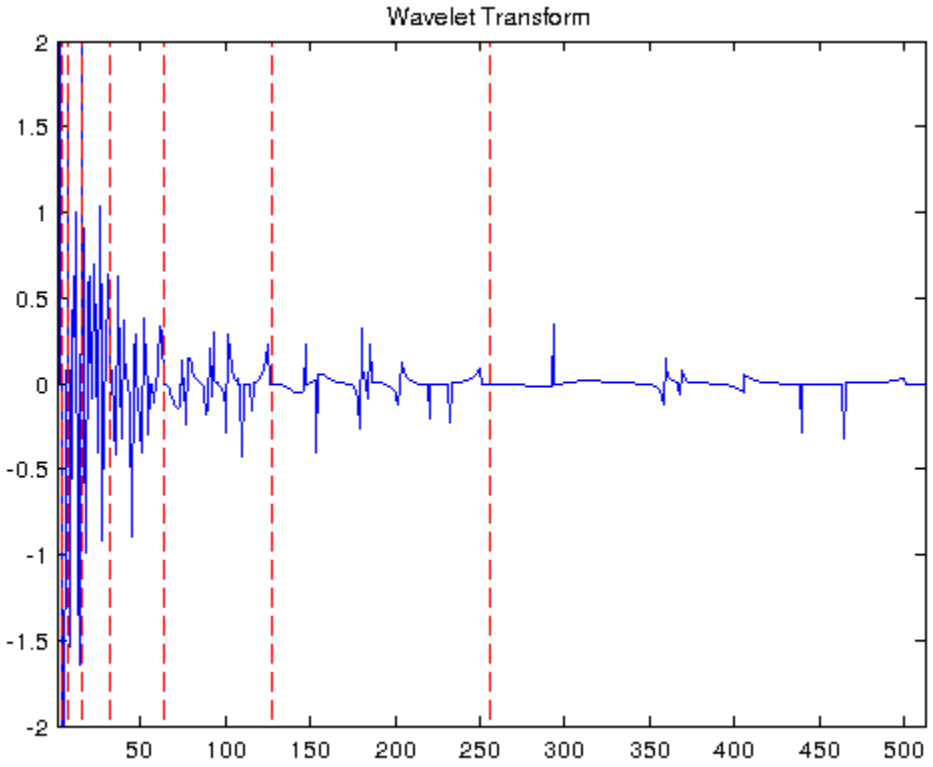
Exercise 1 - 1D Haar Wavelets

```
fw = f;
jj = 8;
Detail = [];
for ii = 1:jj+1
    j = log2(n)-ii;
    A = fw(1:2^(j+1));
    coarse = ( A(1:2:length(A)) + A(2:2:length(A)) )/sqrt(2);
    detail = ( A(1:2:length(A)) - A(2:2:length(A)) )/sqrt(2);
    Detail = [detail; Detail];
    fw = [coarse; Detail];
end

figure(1)
subplot(4,1,1)
    plot(f)
    axis tight
    title('signal')
```

```
for ii = 1:3
    subplot(4,1,ii+1)
        plot(fw( (2^(jj+1-ii))+1 : (2^(jj+2-ii)) ) )
        title(sprintf('Detail j = %i',jj - ii))
        axis tight
end

disp(strcat(['Energy of the signal       = ' num2str(norm(f).^2,3)]));
disp(strcat(['Energy of the coefficients = ' num2str(norm(fw).^2,3)]));

figure(2)
    plot_wavelet(fw);
    title('Wavelet Transform')
    axis([1 n -2 2]);
```

```
Energy of the signal       = 88.6
Energy of the coefficients = 88.6
```

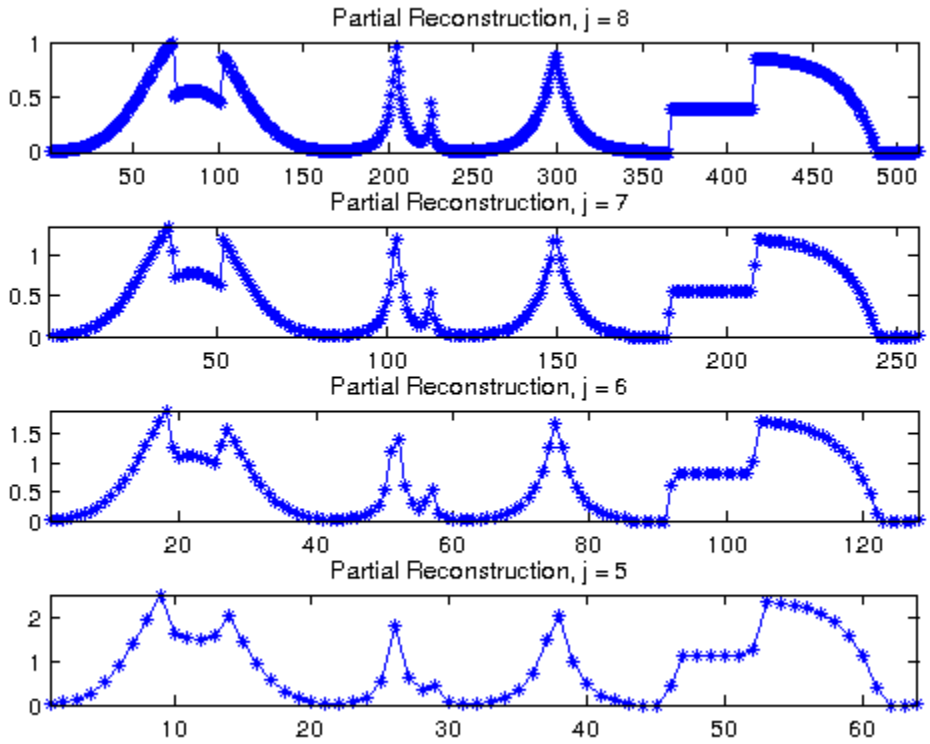




Exercise 2 - 1D Haar Wavelets

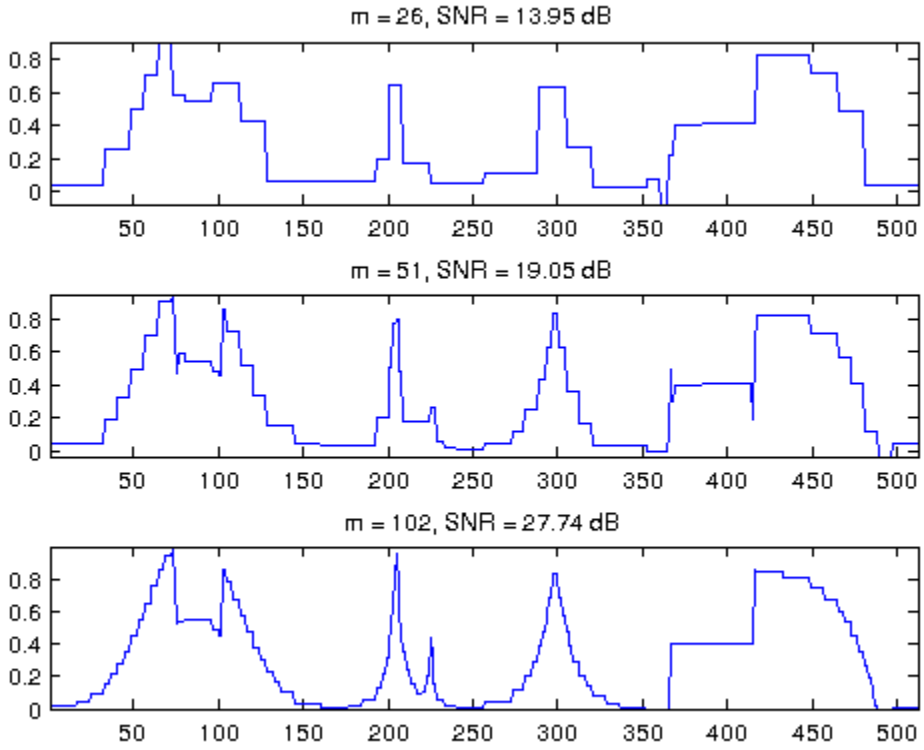
```
f1 = fw;
Coarse = [];

for j = 0:jj
    coarse = f1(1:2^j);
    Detail = f1(2^j+1:2^(j+1));
    f1(1:2:2^(j+1)) = ( coarse + Detail )/sqrt(2);
    f1(2:2:2^(j+1)) = ( coarse - Detail )/sqrt(2);
    Coarse = [Coarse ;coarse];
end

disp(strcat((['Error |f-f1|/|f| = ' num2str(norm(f-f1)/norm(f))])));

figure(3)
subplot(4,1,1)
    plot(f);
    title('Partial Reconstruction, j = 8')
    hold on;
    plot(f1,'*')
    axis tight
for ii = 1:3
subplot(4,1,ii+1)
    plot(Coarse( (2^(jj+1-ii)) : (2^(jj+2-ii)) - 1),'-*')
    title(sprintf('Partial Reconstruction, j = %i ',jj-ii))
    axis tight
    hold on
end
```

```
Error |f-f1|/|f| = 1.0168e-15
```



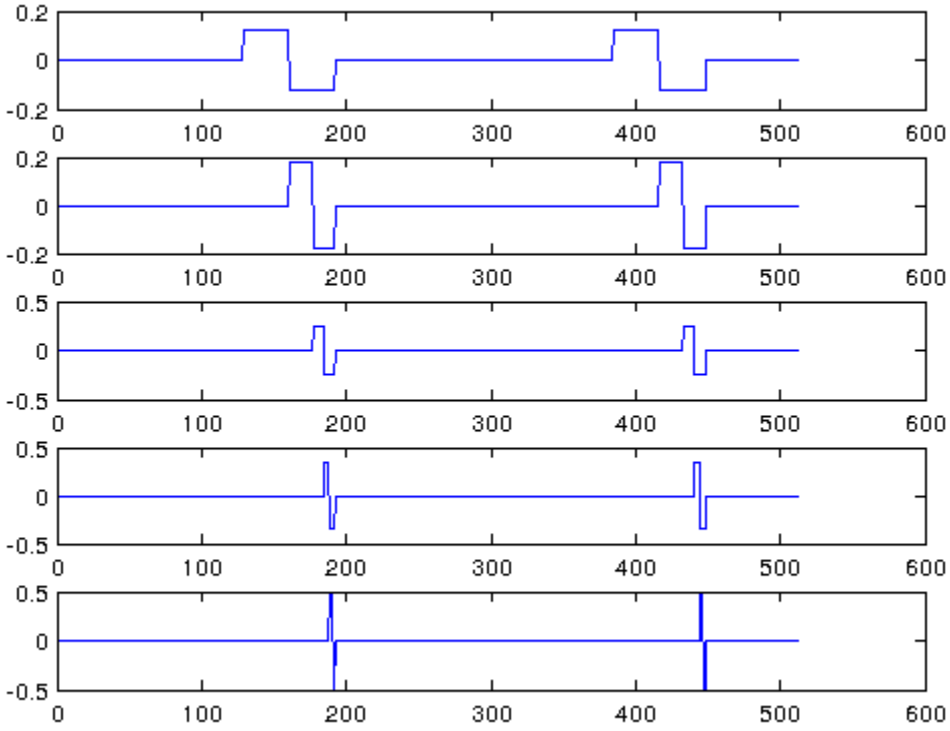
Exercise 3 - 1D Haar Wavelets

```
m = [26,51,102];

for ii = 1:3

    snark = sort(abs(fw),1,'descend');
    T = snark(m(ii));
    fwT = fw .* (abs(fw)>=T);

    for j = 0:jj
        coarse = fwT(1:2^j);
        Detail = fwT(2^j+1:2^(j+1));
        fwT(1:2:2^(j+1)) = ( coarse + Detail )/sqrt(2);
        fwT(2:2:2^(j+1)) = ( coarse - Detail )/sqrt(2);
    end

    figure(35)
    subplot(3,1,ii)
      plot(fwT)
      title(sprintf('m = %i, SNR = %2.2f dB',m(ii),snr(f,fwT) ))
      axis tight

end
```



Exercise 4 - 1D Haar Wavelets

```
j = 5:-1:1;
for ii = 1:length(j)

    p = 2^abs(5-j(ii));
    k = (n - 2^j(ii) * p)/2^j(ii);
    k2= (n - 2^j(ii) * (p+2^abs(7-j(ii))) )/2^j(ii);
    fww = zeros(n,1);
    fww(k) = 1;
    fww(k2) = 1;
    for jj = 0:8
        coarse = fww(1:2^jj);
        Detail = fww(2^jj+1:2^(jj+1));
        fww(1:2:2^(jj+1)) = ( coarse + Detail )/sqrt(2);
        fww(2:2:2^(jj+1)) = ( coarse - Detail )/sqrt(2);
    end
    figure(2132)
    subplot(5,1,ii)
    plot(fww)
end
```

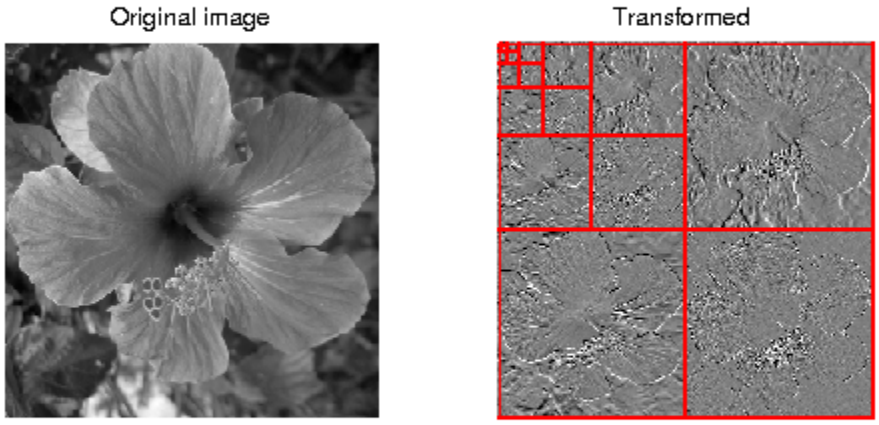


Exercise 1 - 2D Haar Wavelets

```
n = 256;
name = 'hibiscus';
M = load_image(name,n);
M = rescale( sum(M,3) );
MW = M;
j = log2(n)-1;

for j = j:-1:0
    A = MW(1:2^(j+1),1:2^(j+1));
    % Vertical Transform
    Coarse = ( A(1:2:size(A,1),:) + A(2:2:size(A,1),:) )/sqrt(2);
    Detail = ( A(1:2:size(A,1),:) - A(2:2:size(A,1),:) )/sqrt(2);
    A = cat3(1, Coarse, Detail );

    % Horizontal transformk
    Coarse = ( A(:,1:2:size(A,1)) + A(:,2:2:size(A,1)) )/sqrt(2);
    Detail = ( A(:,1:2:size(A,1)) - A(:,2:2:size(A,1)) )/sqrt(2);
    A = cat3(2, Coarse, Detail );
    % Subdivide
    MW(1:2^(j+1),1:2^(j+1)) = A;
end
% Check enery equality
disp(strcat(['Energy of the signal       = ' num2str(norm(M(:)).^2)]));
disp(strcat(['Energy of the coefficients = ' num2str(norm(MW(:)).^2)]));

figure(2456)
imageplot(M,'Original image',1,2,1);
subplot(1,2,2);
```

```
plot_wavelet(MW,1); title('Transformed')
```

```
Energy of the signal       = 12708.0303
Energy of the coefficients = 12708.0303
```

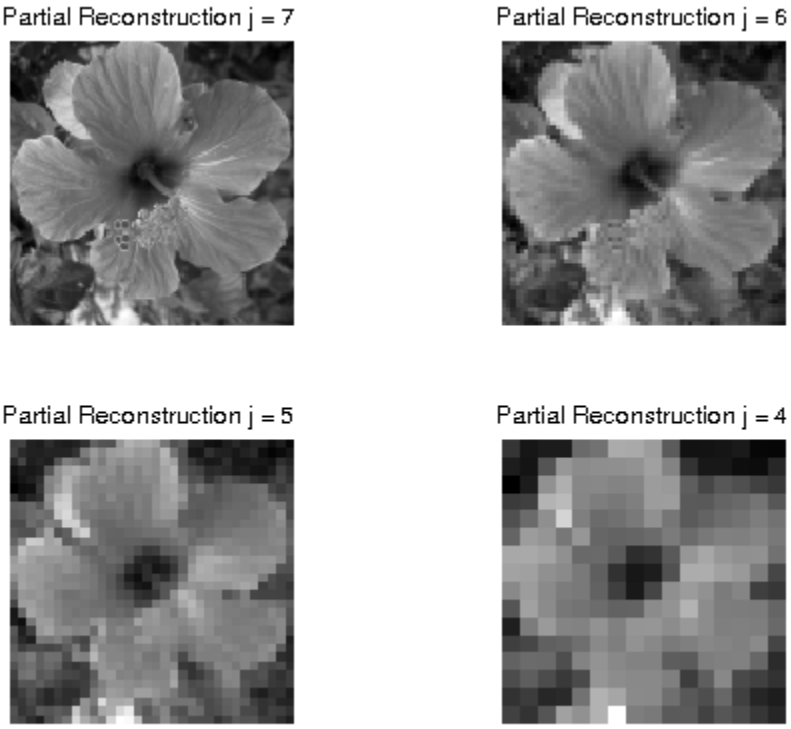


Exercise 2 - 2D Haar Wavelets

```
M1 = MW;
j = 0;

for j = 0:7
    A = M1(1:2^(j+1),1:2^(j+1));
    Coarse = A(1:2^j,:);
    Detail = A(2^j+1:2^(j+1),:);
    %Undo the transform by sum and difference.
    A(1:2:size(A,1),:) = ( Coarse + Detail )/sqrt(2);
    A(2:2:size(A,2),:) = ( Coarse - Detail )/sqrt(2);
    %Retrieve coarse and detail coefficients in the horizontal direction.
    Coarse = A(:,1:2^j);
    Detail = A(:,2^j+1:2^(j+1));
    %Undo the transform by sum and difference.
    A(:,1:2:size(A,1)) = ( Coarse + Detail )/sqrt(2);
    A(:,2:2:size(A,2)) = ( Coarse - Detail )/sqrt(2);
    %Assign the result.
    C{j+1} = A;
    M1(1:2^(j+1),1:2^(j+1)) = A;

end
```

```
q = [7,6,5,4];
figure(2517)
for ii = 1:4
    subplot(2,2,ii)
    imageplot(C{q(ii)})
    title(sprintf('Partial Reconstruction j = %i',q(ii)))
end
disp(strcat((['Error |M-M1|/|M| = ' num2str(norm(M(:)-M1(:))/norm(M(:)))])));
```

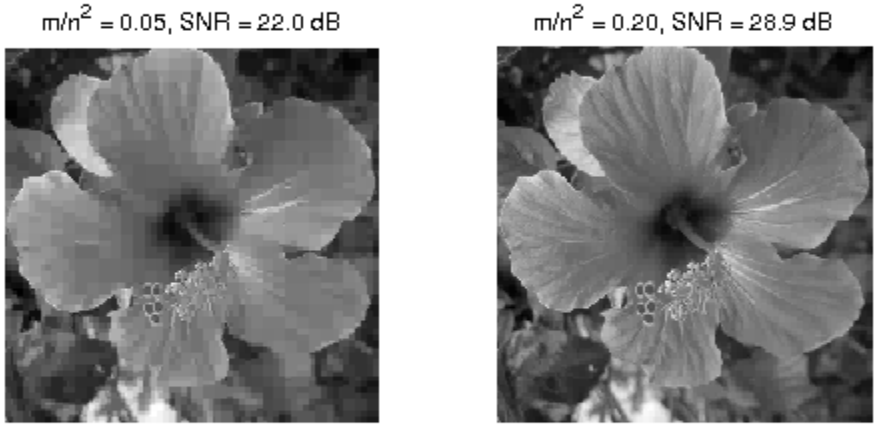
```
Error |M-M1|/|M| = 2.0389e-15
```



Exercise 3 - 2D Haar Wavelets

```
m = round(n^2*[0.05,0.2]);
for ii = 1:length(m)

    snark = sort(abs(MW(:)),1,'descend');
    T = snark(m(ii));
    MWT = MW .* (abs(MW)>T);

   for j = 0:7
    A = MWT(1:2^(j+1),1:2^(j+1));
    Coarse = A(1:2^j,:);
    Detail = A(2^j+1:2^(j+1),:);
    %Undo the transform by sum and difference.
    A(1:2:size(A,1),:) = ( Coarse + Detail )/sqrt(2);
    A(2:2:size(A,2),:) = ( Coarse - Detail )/sqrt(2);
    %Retrieve coarse and detail coefficients in the horizontal direction.
    Coarse = A(:,1:2^j);
```

```
    Detail = A(:,2^j+1:2^(j+1));
    %Undo the transform by sum and difference.
    A(:,1:2:size(A,1)) = ( Coarse + Detail )/sqrt(2);
    A(:,2:2:size(A,2)) = ( Coarse - Detail )/sqrt(2);
    %Assign the result.
    MWT(1:2^(j+1),1:2^(j+1)) = A;
   end

   figure(134)
   imageplot(MWT,sprintf('m/n^2 = %1.2f, SNR = %2.1f dB',...
    m(ii)/n^2,snr(M,MWT)),1,2,ii);

end
```

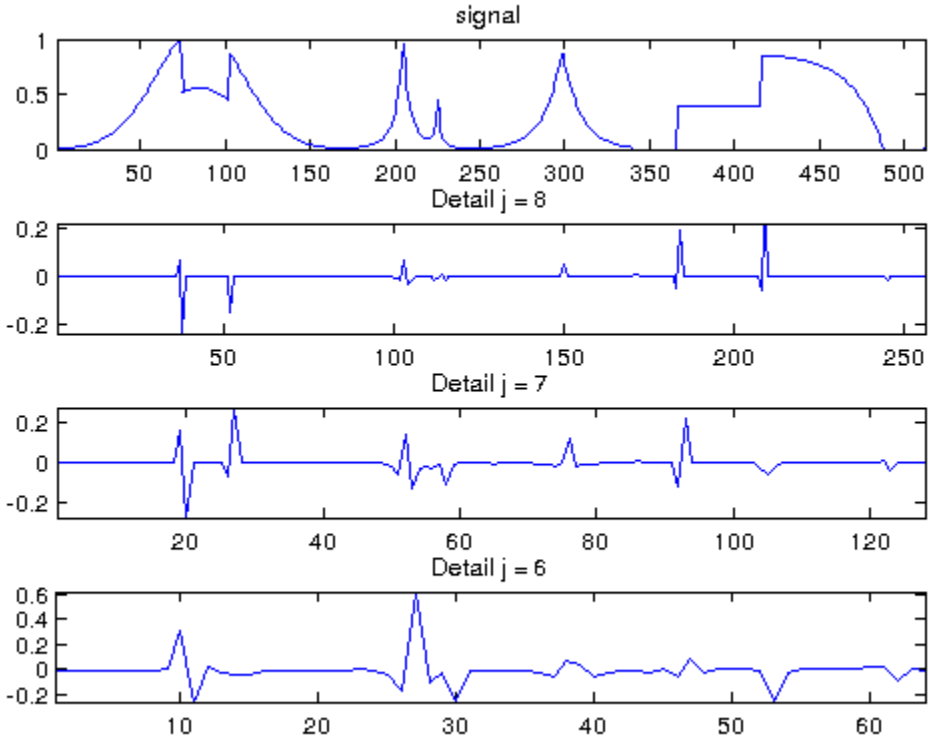


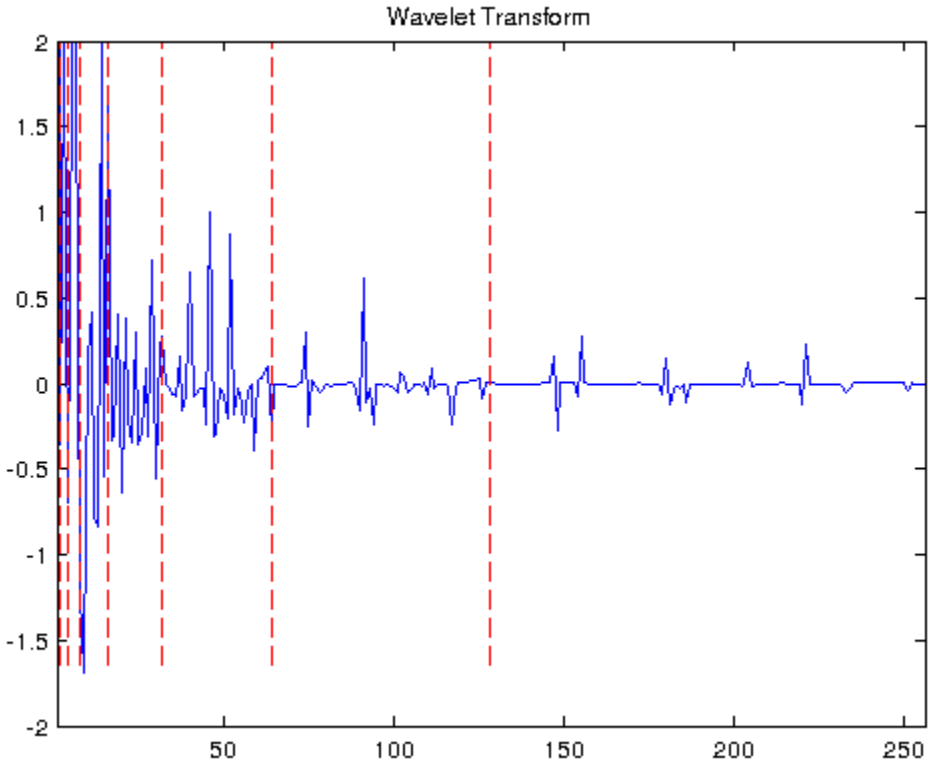
Exercise 1 - 1-D Daubechies Wavelets

```
p = 4;
[h,g] = compute_wavelet_filter('Daubechies',p);
% Create Signal
name = 'piece-regular';
N = 512;
f = rescale( load_signal(name, N) );
fw = f;
jj = log2(N);

for ii = 1:jj
% Set number of j's
    j = jj - ii;
% Get coefficients and store them in vector a1
    a1 = fw(1:2^(j+1));
```

```
% Low pass / High pass filter and subsample
    a = subsampling(cconv(a1,h));
    d = subsampling(cconv(a1,g));
% Concatenate
    fw(1:2^(j+1)) = cat(1, a, d );
end


figure(1132)
subplot(4,1,1)
    plot(f)
    axis tight
    title('signal')

for ii = 1:3
    subplot(4,1,ii+1)
        plot(fw( (2^(jj-ii))+1 : (2^(jj+1-ii)) ) )
        title(sprintf('Detail j = %i',jj - ii))
        axis tight
end

disp(strcat(['Energy of the signal       = ' num2str(norm(f).^2,3)]));
disp(strcat(['Energy of the coefficients = ' num2str(norm(fw).^2,3)]));

figure(234)
    plot_wavelet(fw);
    title('Wavelet Transform')
    axis([1 n -2 2]);
```

```
Energy of the signal       = 88.6
Energy of the coefficients = 88.6
```

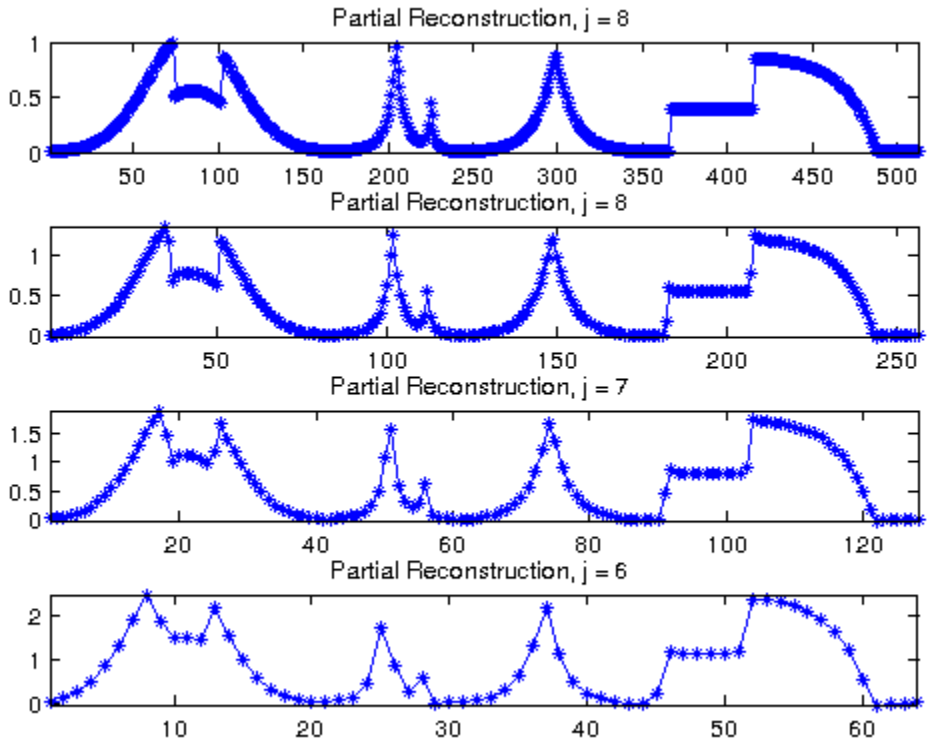




Exercise 2 - 1-D Daubechies Wavelets

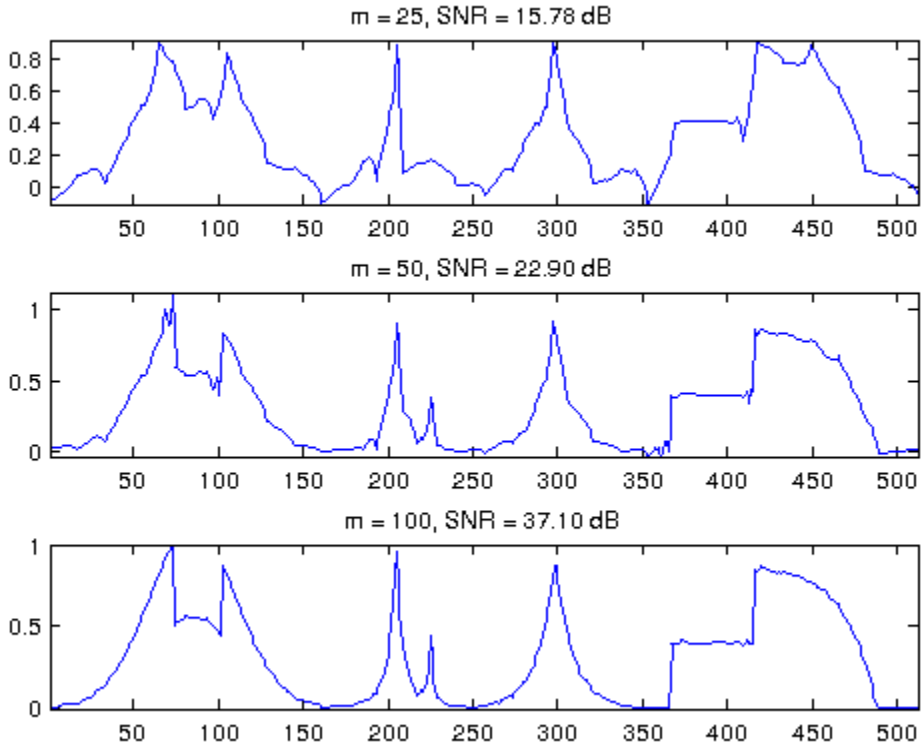
```
f1 = fw;
aa = [];

for j = 0:jj-1
    a = f1(1:2^j);
    d = f1(2^j+1:2^(j+1));
    a = cconv(upsampling(a,1),reverse(h),1) + cconv(upsampling(d,1),reverse(g),1);
    f1(1:2^(j+1)) = a;
    aa = [aa; a];
end

disp(strcat((['Error |f-f1|/|f| = ' num2str(norm(f-f1)/norm(f))])));

figure(311)
subplot(4,1,1)
    plot(f);
    title('Partial Reconstruction, j = 8')
    hold on;
    plot(f1,'*')
    axis tight
for ii = 1:3
subplot(4,1,ii+1)
    plot(aa( (2^(jj-ii)) : (2^(jj+1-ii)) - 1),'-*')
    title(sprintf('Partial Reconstruction, j = %i ',jj-ii))
    axis tight
    hold on
end
```

```
Error |f-f1|/|f| = 4.0189e-12
```



Exercise 3 - 1-D Daubechies Wavelets

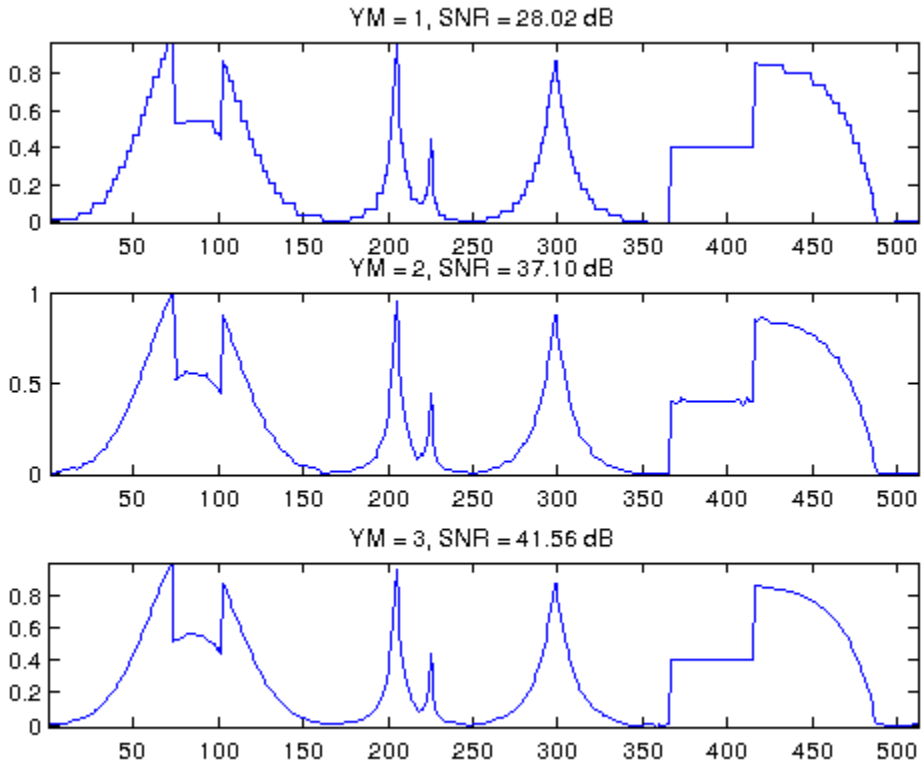
```
m = [25,50,100];

for ii = 1:3

    snark = sort(abs(fw),1,'descend');
    T = snark(m(ii));
    fwT = fw .* (abs(fw)>=T);

    for j = 0:jj-1
        a = fwT(1:2^j);
        d = fwT(2^j+1:2^(j+1));
        a = cconv(upsampling(a,1),reverse(h),1) + cconv(upsampling(d,1),reverse(g)
        fwT(1:2^(j+1)) = a;
    end

    figure(353)
    subplot(3,1,ii)
      plot(fwT)
      title(sprintf('m = %i, SNR = %2.2f dB',m(ii),snr(f,fwT) ))
      axis tight

end
```



Exercise 4 - 1-D Daubechies Wavelets

```
m = 100;
h1 = h; % Store k=2 wavelet filters
g1 = g;
for kk = 1:3
    fw = f;
    p = 2*kk;
    [h,g] = compute_wavelet_filter('Daubechies',p);

    for ii = 1:jj
        % Set number of j's
        j = jj - ii;
        % Get coefficients and store them in vector a1
        a1 = fw(1:2^(j+1));
        % Low pass / High pass filter and subsample
        a = subsampling(cconv(a1,h));
        d = subsampling(cconv(a1,g));
        % Concatenate
        fw(1:2^(j+1)) = cat(1, a, d );
    end

    snark = sort(abs(fw),1,'descend');
    T = snark(m);
    f1 = fw .* (abs(fw)>=T);

    for j = 0:jj-1
        a = f1(1:2^j);
        d = f1(2^j+1:2^(j+1));
        a = cconv(upsampling(a,1),reverse(h),1) + cconv(upsampling(d,1),...
```

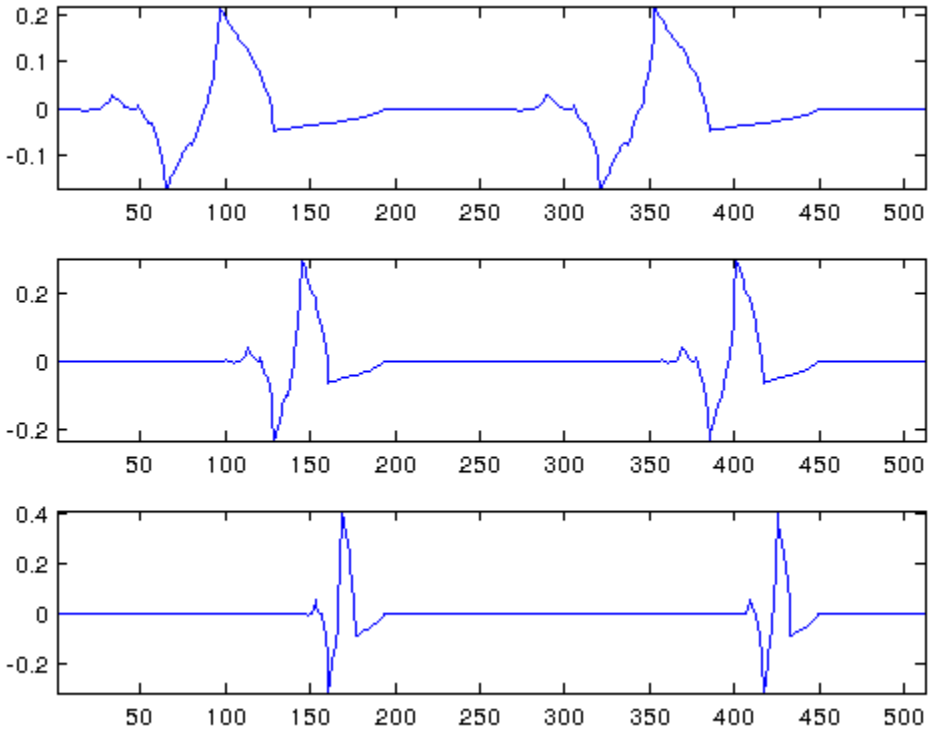
```
            reverse(g),1);
        f1(1:2^(j+1)) = a;
    end

    figure(311)
    subplot(3,1,kk)
    plot(f1)
    title(sprintf('YM = %i, SNR = %2.2f dB',kk,snr(f,f1) ))
    axis tight
end
```



Exercise 5 - 1_D Daubechies Wavelets

```
p = 4;
h = h1;
g = g1;

j = 5:-1:3;
for ii = 1:length(j)
    % Compute delta functions at several positions and scales
    p = 2^abs(5-j(ii));
    k = (N - 2^j(ii) * p)/2^j(ii);
    k2= (N - 2^j(ii) * (p+2^abs(7-j(ii))) )/2^j(ii);
    fww = zeros(N,1);
    fww(k) = 1;
    fww(k2) = 1;

    for jj = 0:8
        % Change delta function into wavelet
        a = fww(1:2^jj);
```

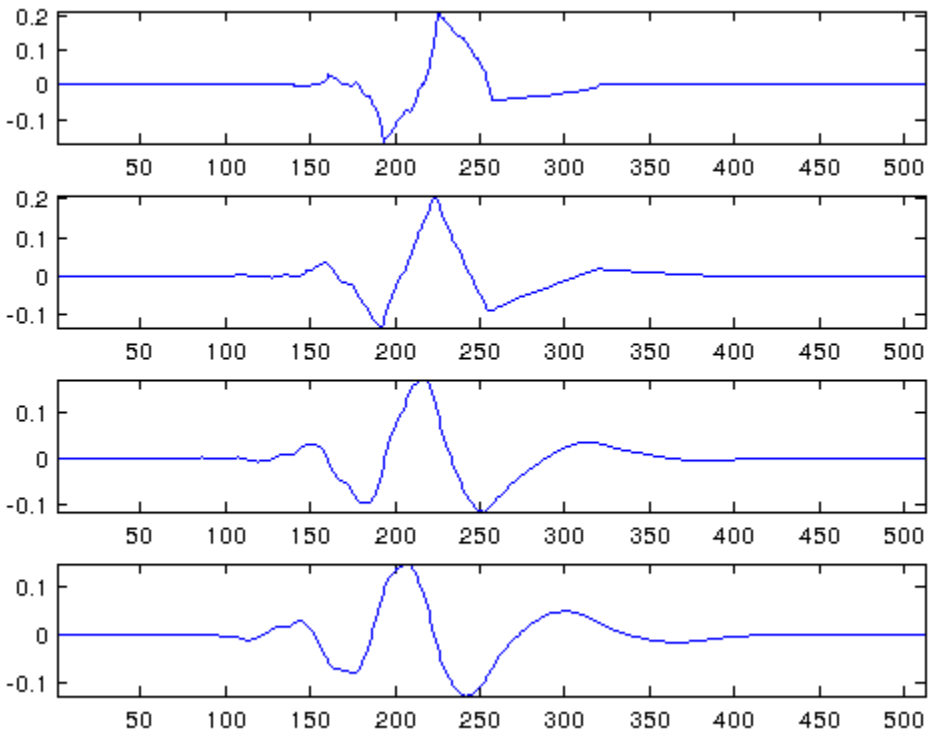
```
        d = fww(2^jj+1:2^(jj+1));
        a = cconv(upsampling(a,1),reverse(h),1) + cconv(upsampling(d,1),reverse(g)
        fww(1:2^(jj+1)) = a;
    end
    figure(2213)
    subplot(3,1,ii)
    plot(fww)
    axis tight
end
```



Exercise 6 - 1_D Daubechies Wavelets

```
j = 5;
K = [2:5];
for kk = 1:length(K)
    fw = f;
    p = 2*K(kk);
    [h,g] = compute_wavelet_filter('Daubechies',p);
    k = 13;
    fww = zeros(N,1);
    fww(k) = 1;


    for jj = 0:8
        % Change delta function into wavelet
        a = fww(1:2^jj);
        d = fww(2^jj+1:2^(jj+1));
        a = cconv(upsampling(a,1),reverse(h),1) + cconv(upsampling(d,1),reverse(g)
        fww(1:2^(jj+1)) = a;
    end
```

```
    figure(2213)
    subplot(4,1,kk)
    plot(fww)
    axis tight
end
```



Published with MATLAB® 7.11