

Fourier Analysis and Wavelets

Homework 5

Francisco Jose Castillo Carrasco

October 22, 2018

Problem 2

Let ϕ and ψ be the Haar scaling and wavelet functions, respectively. Let V_j and W_j be the spaces generated by $\phi(2^k x - k), k \in \mathbb{Z}$, and $\psi(2^j x - k), k \in \mathbb{Z}$, respectively. Consider the function defined on $0 \leq x \leq 1$ given by

$$f(x) = \begin{cases} 2 & 0 \leq x < 1/4, \\ -3 & 1/4 \leq x < 1/2, \\ 1 & 1/2 \leq x < 3/4, \\ 3 & 3/4 \leq x < 1. \end{cases}$$

Express f first in terms of the basis for V_2 and then decompose f into its component parts in W_1, W_0 and V_0 . In other words, find the Haar wavelet decomposition for f . Sketch each of these components.

Solution: The function f is easily expressed as

$$f(x) = 2\phi(4x) - 3\phi(4x - 1) + \phi(4x - 2) + 3\phi(4x - 3) \in V_2$$

For the decomposition we use the following relations

$$\begin{aligned} \phi(2^j x) &= \frac{\phi(2^{j-1} x) + \psi(2^{j-1} x)}{2}, \\ \phi(2^j x - 1) &= \frac{\phi(2^{j-1} x) - \psi(2^{j-1} x)}{2}. \end{aligned}$$

For this case we use

$$\begin{aligned} \phi(4x) &= \frac{\phi(2x) + \psi(2x)}{2}, \\ \phi(4x - 1) &= \frac{\phi(2x) - \psi(2x)}{2}, \\ \phi(4x - 2) &= \phi(4(x - 1/2)) = \frac{\phi(2x - 1) + \psi(2x - 1)}{2}, \\ \phi(4x - 3) &= \phi(4(x - 1/2) - 1) = \frac{\phi(2x - 1) - \psi(2x - 1)}{2}. \end{aligned}$$

With these relations we decompose f into its components in W_1 and V_1 :

$$f(x) = \frac{5}{2}\psi(2x) - \psi(2x - 1) - \frac{1}{2}\phi(2x) + 2\phi(2x - 1) \in W_1 \oplus V_1.$$

Next, we decompose the V_1 component, $f_{V_1} = -\frac{1}{2}\phi(2x) + 2\phi(2x-1)$, into its components in W_0 and V_0 using

$$\begin{aligned}\phi(2x) &= \frac{\phi(x) + \psi(x)}{2}, \\ \phi(2x-1) &= \frac{\phi(x) - \psi(x)}{2}.\end{aligned}$$

Then,

$$f_{V_1} = -\frac{5}{4}\psi(x) + \frac{3}{4}\phi(x) \in W_0 \oplus V_0.$$

Hence,

$$f(x) = \frac{5}{2}\psi(2x) - \psi(2x-1) - \frac{5}{4}\psi(x) + \frac{3}{4}\phi(x) \in W_1 \oplus W_0 \oplus V_0.$$

Every component of the function and the function itself are graphed in the following figures.

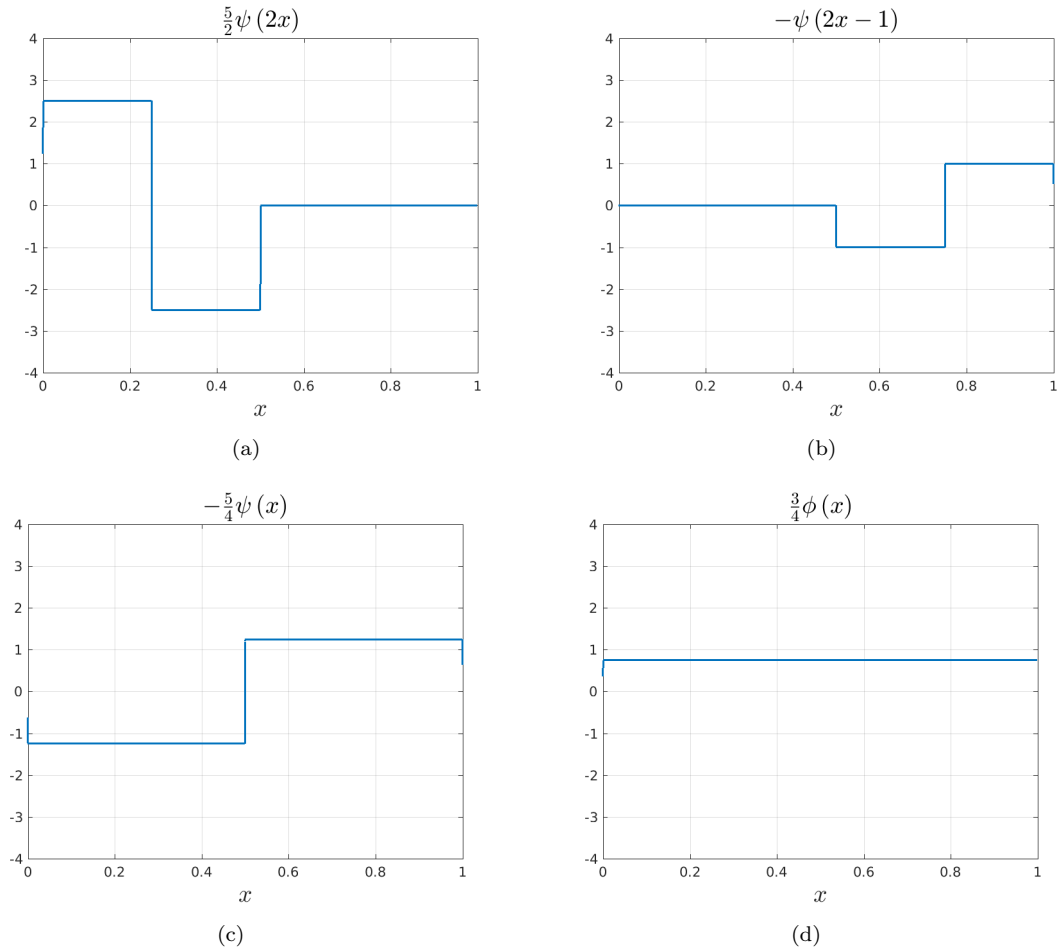


Figure 1: Haar components of the function g .

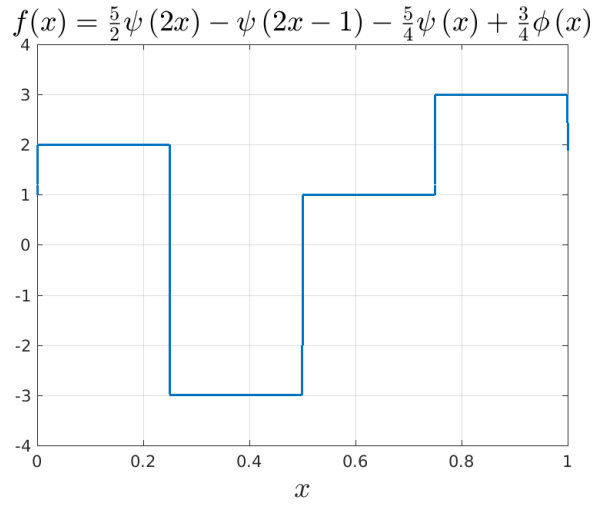


Figure 2: Function $g = \sum_{k=0}^7 a_k^3 \phi(8x - k)$.

Problem 4

Let V_n be the spaces generated by $\phi(2^n x - k)$, $k \in \mathbb{Z}$, where ϕ is the Haar scaling function.

- On the interval $0 \leq x \leq 1$, what are the dimensions of the spaces V_n and W_n for $n \geq 0$?

Solution: The functions $\phi(2^n x - k)$, $k \in \mathbb{Z}$ represent step functions of width 2^{-n} . Given the interval $0 \leq x \leq 1$, it is immediate that we need 2^n functions to cover the whole interval. More formally, by *Theorem 4.6*, we know that $\{2^{n/2}\phi(2^n x - k), k \in \mathbb{Z}\}$ is a basis for V_n and $\{2^{n/2}\psi(2^n x - k), k \in \mathbb{Z}\}$ is a basis for W_n . In this case, because of the given interval, $k = 0, \dots, 2^n - 1$, which makes the dimension of both spaces V_n and W_n equal to 2^n .

- Using the result

$$\dim(A \oplus B) = \dim(A) + \dim(B),$$

count the dimension of the space on the right side of the equality

$$V_n = W_{n-1} \oplus W_{n-2} \oplus \dots \oplus W_0 \oplus V_0$$

Solution: We proceed using the previous relation

$$\begin{aligned} \dim(V_n) &= \dim(W_{n-1}) + \dim(W_{n-2}) + \dots + \dim(W_1) + \dim(W_0) + \dim(V_0), \\ &= 2^{n-1} + 2^{n-2} + \dots + 2^1 + 2^0 + 2^0, \\ &= 2^0 + \sum_{k=0}^{n-1} 2^k = 1 + 2^n - 1 = 2^n, \end{aligned}$$

we get the same result as in the first part.

Problem 6

Reconstruct $g \in V_3$, given these coefficients in its Haar wavelet decomposition:

$$a^2 = [1/2, 2, 5/2, -3/2],$$

$$b^2 = [-3/2, -1, 1/2, -1/2].$$

The first entry in each list corresponds to $k = 0$. Sketch g .

Solution: It is easy to reconstruct $g \in V_3$ using the relations from *Theorem 4.14*,

$$g(x) = \sum_{k=0}^{2^3-1} a_k^3 \phi(2^3 x - k) \in V_3,$$

where

$$a_k^3 = \begin{cases} a_l^2 + b_l^2 & \text{if } k = 2l \text{ is even,} \\ a_l^2 - b_l^2 & \text{if } k = 2l + 1 \text{ is odd.} \end{cases}$$

Therefore,

$$a^3 = [a_0^2 + b_0^2, a_0^2 - b_0^2, a_1^2 + b_1^2, a_1^2 - b_1^2, a_2^2 + b_2^2, a_2^2 - b_2^2, a_3^2 + b_3^2, a_3^2 - b_3^2]$$

$$a^3 = [-1, 2, 1, 3, 3, 2, -2].$$

Hence,

$$g(x) = \sum_{k=0}^7 a_k^3 \phi(8x - k),$$

which we can expand as

$$g(x) = -\phi(8x) + 2\phi(8x - 1) + \phi(8x - 2) + 3\phi(8x - 3) + 3\phi(8x - 4) + 2\phi(8x - 5) - 2\phi(8x - 6) - \phi(8x - 7),$$

and it is graphed in the next figure.

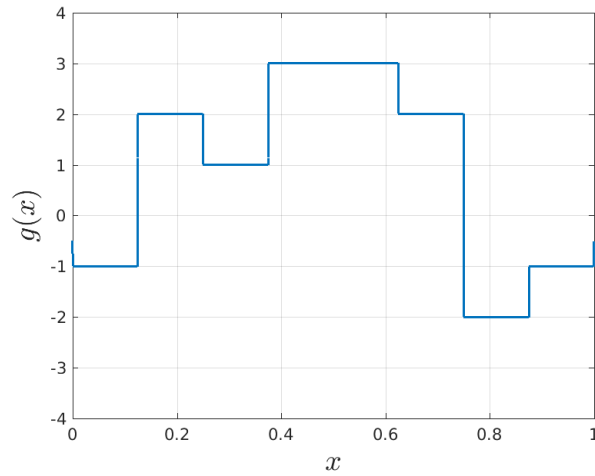


Figure 3: Function $g = \sum_{k=0}^7 a_k^3 \phi(8x - k)$.

Problem 9

Let

$$f(t) = e^{-t^2/10} (\sin(2t) + 2\cos(4t) + 0.4\sin(t)\sin(50t)) .$$

Discretize the function f over the interval $0 \leq t \leq 1$ as described in Step 1 of Section 4.4. Use $n = 8$ as the top level (so there are 2^8 nodes in the discretization). Implement the decomposition algorithm described in Step 2 of Section 4.4 using the Haar wavelets. Plot the resulting levels, $f_{j-1} \in V_{j-1}$ for $j = 8 \dots 1$ and compare with the original signal.

Solution: *The solution to this problem below, with the Matlab code.*

Problem 10

Filter the wavelet coefficients computed in exercise 9 by setting to zero any wavelet coefficient whose absolute value is less than $tol = 0.1$. Then reconstruct the signal as described in Section 4.4. Plot the reconstructed f_8 and compare with the original signal. Compute the relative l^2 difference between the original signal and the compressed signal. Experiment with various tolerances. Keep track of the percentage of wavelet coefficients that have been filtered out.

Solution: *The solution to this problem below, with the Matlab code.*

MATLAB CODE - FRANCISCO CASTILLO

Table of Contents

Preliminary Commands	1
Introduction	1
Define the function	1
Decomposition (problem 9)	2
Compressing the signal (Problem 10)	7

Preliminary Commands

```
clear all
close all
clc
linewidth=1.6;
labelfontsize=18;
legendfontsize=12;
```

Introduction

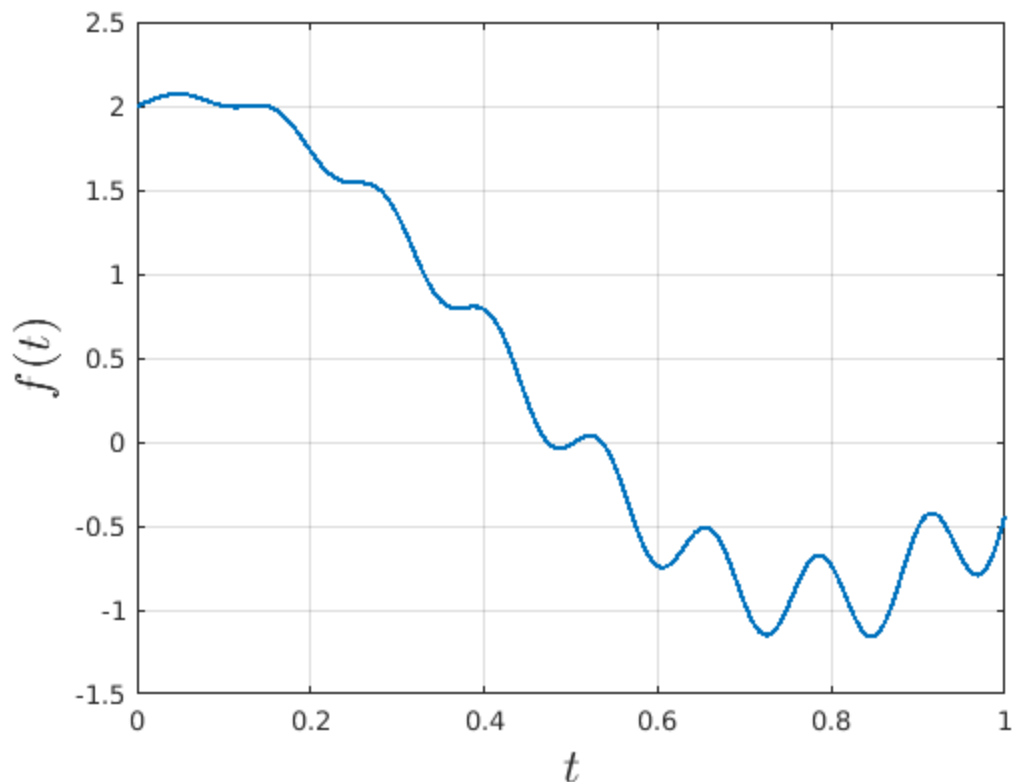
This code solves both problems 9 and 10 of the present homework assignment.

Define the function

```
t=linspace(0,1,256); % interval [0,1] partitioned into 2^8=256 pieces
y = exp(-t.^2/10).*(sin(2*t) + 2*cos(4*t) + .4*sin(t).*sin(50*t));
```

In the following figure we can see the function that we are going to work with in this two problems.

```
figure
plot(t,y,'linewidth',linewidth);
xlabel('$t$', 'interpreter','latex','fontsize',labelfontsize)
ylabel('$f(t)$', 'interpreter','latex','fontsize',labelfontsize)
grid on
axis([0 1 -1.5 2.5])
```

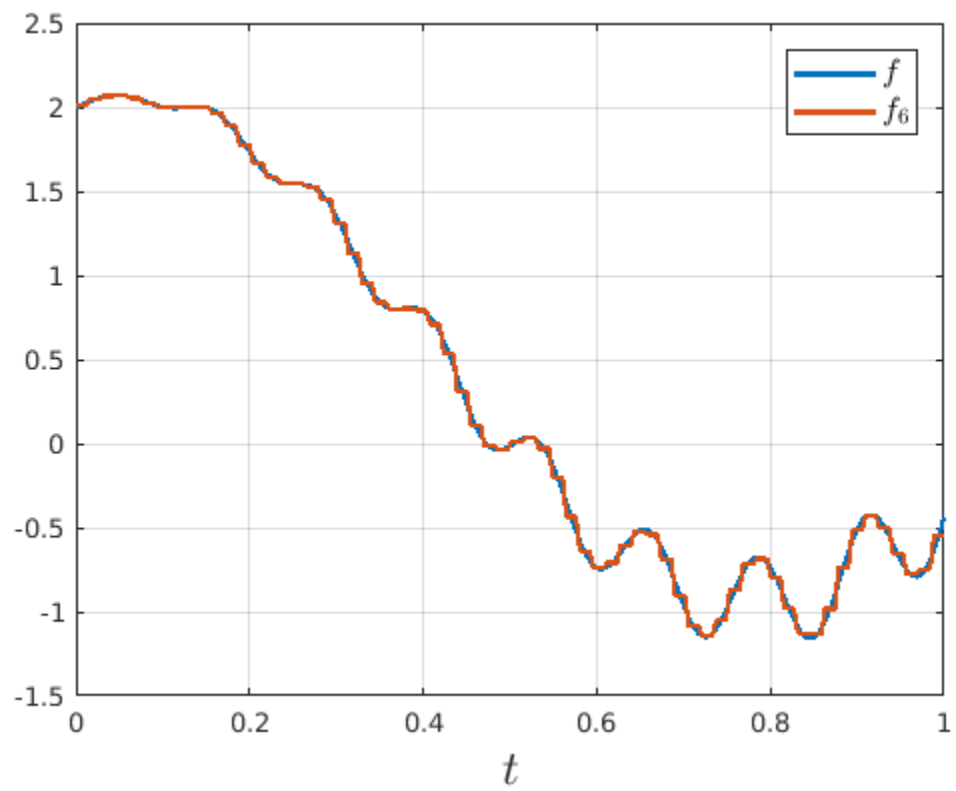
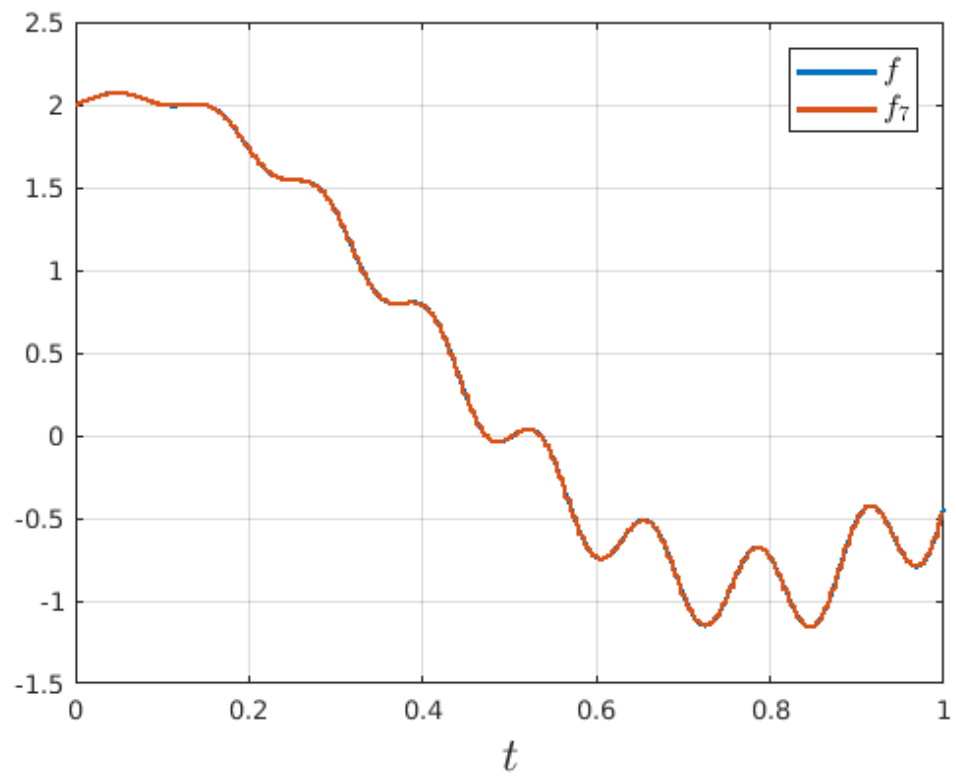


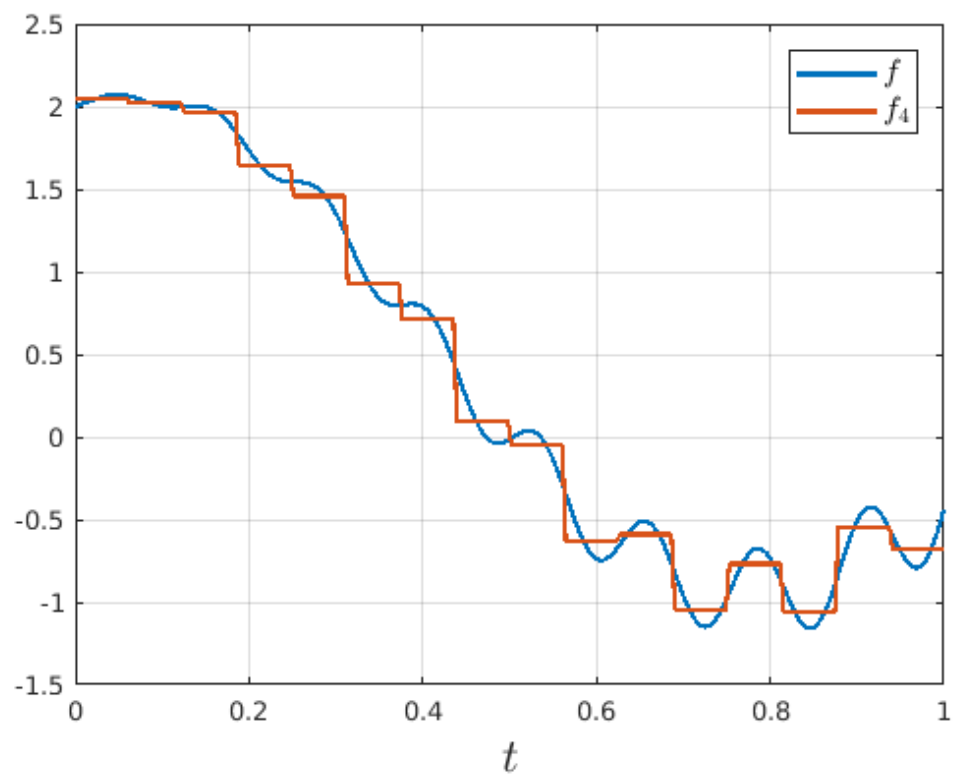
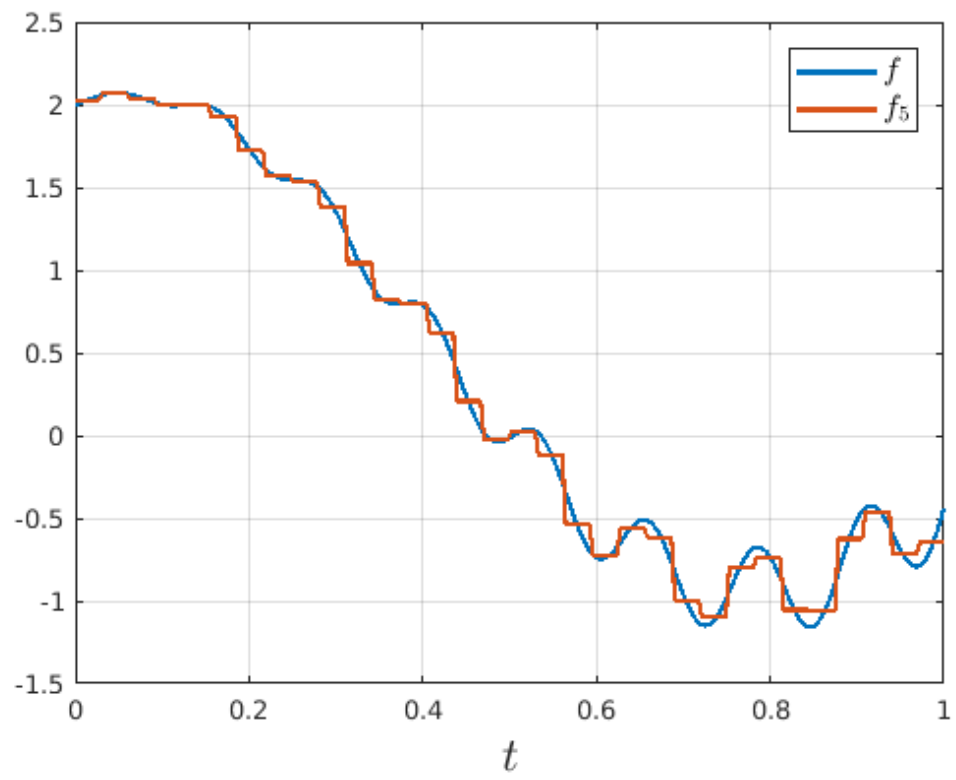
The goal in problem 9 will be decompose the signal using Haar wavelets and plot the resulting levels. In problem 10 the goal will be filter the high frequency noise and obtain a filtered signal, which we will compare to the initial.

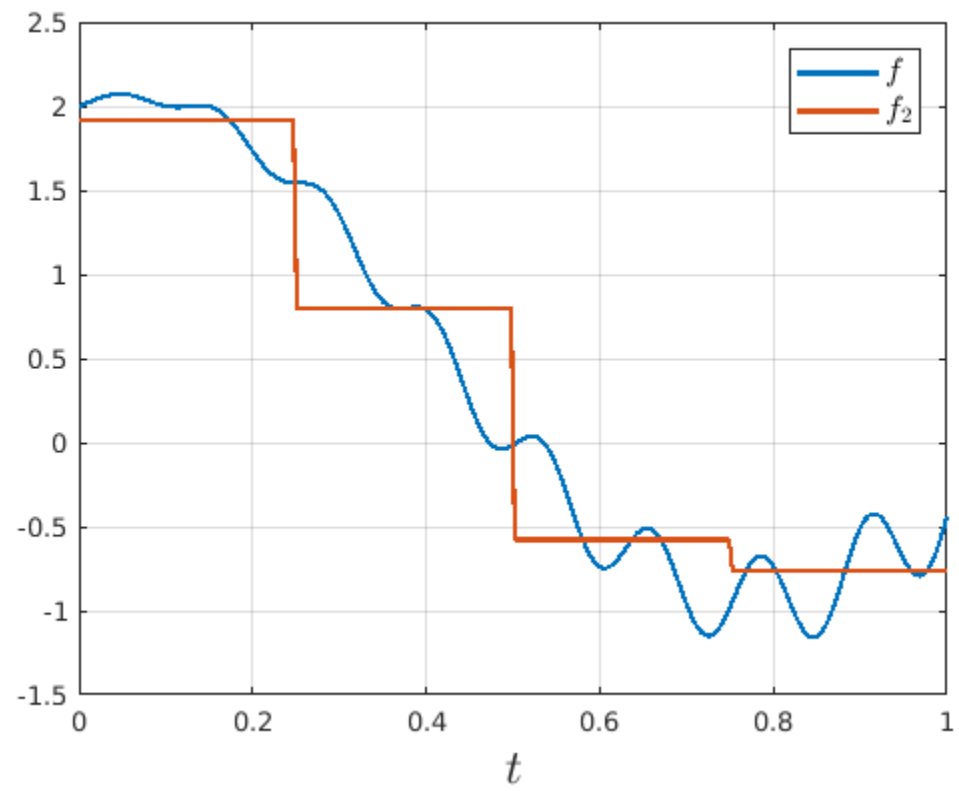
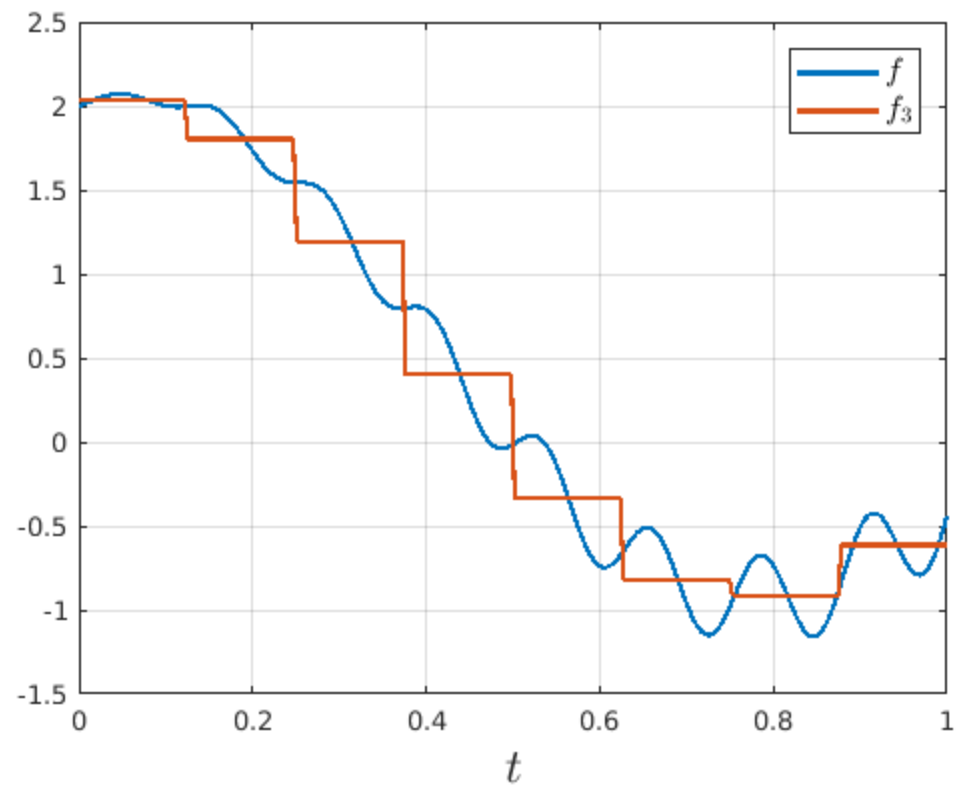
Decomposition (problem 9)

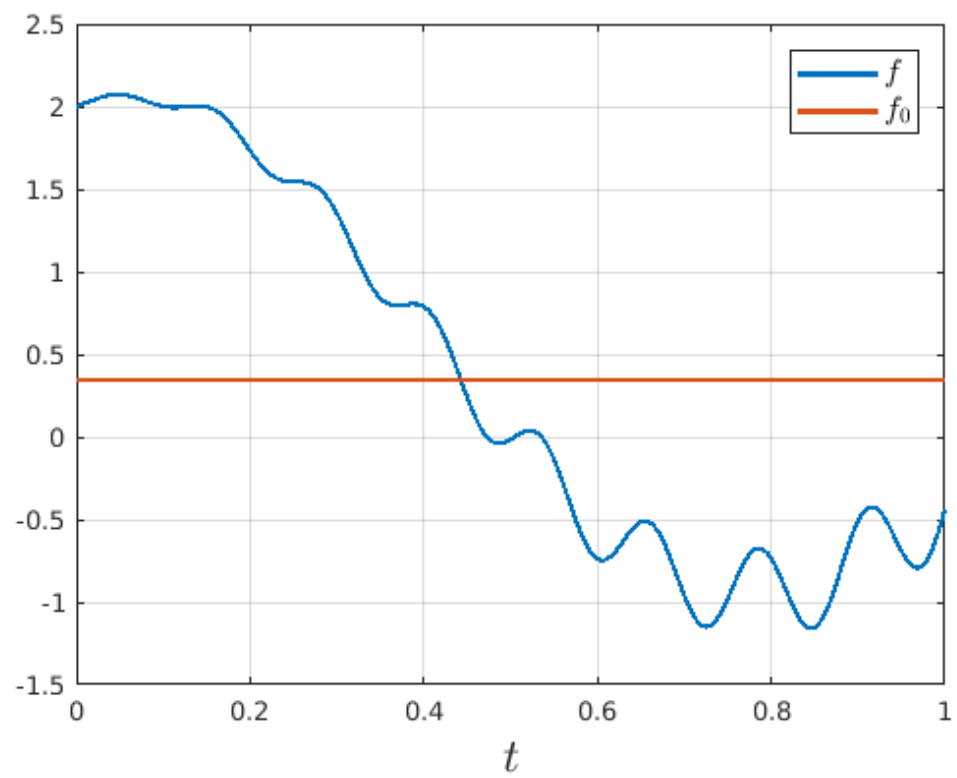
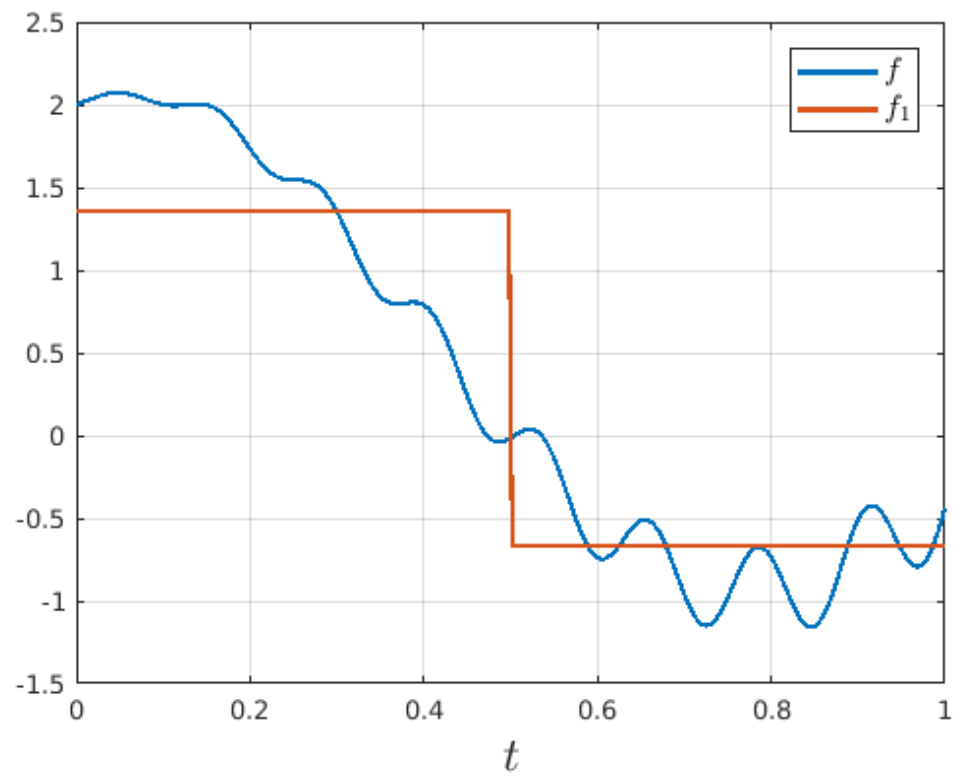
In this section we implement the decomposition algorithm described in Step 2 of Section 4.4.

```
for i=1:8
    [C,L] = wavedec(y,i,'db1'); % i-th level decomposition
    Ai = upcoef('a',C(1:L(1)),'db1',i,length(y));
    figure
    plot(t,y,t,Ai,'linewidth',linewidth); % Plot original function and
    the i-th level decomposition
    grid on
    xlabel('$t$', 'interpreter', 'latex', 'fontsize', labelfontsize)
    axis([0 1 -1.5 2.5])
    h=legend('$f$', ['$f_' num2str(8-i) '$']);
    set(h, 'interpreter', 'latex', 'fontsize', legendfontsize);
    saveas(gcf, ['$f_' num2str(8-i) '_p9'], 'png')
end
```







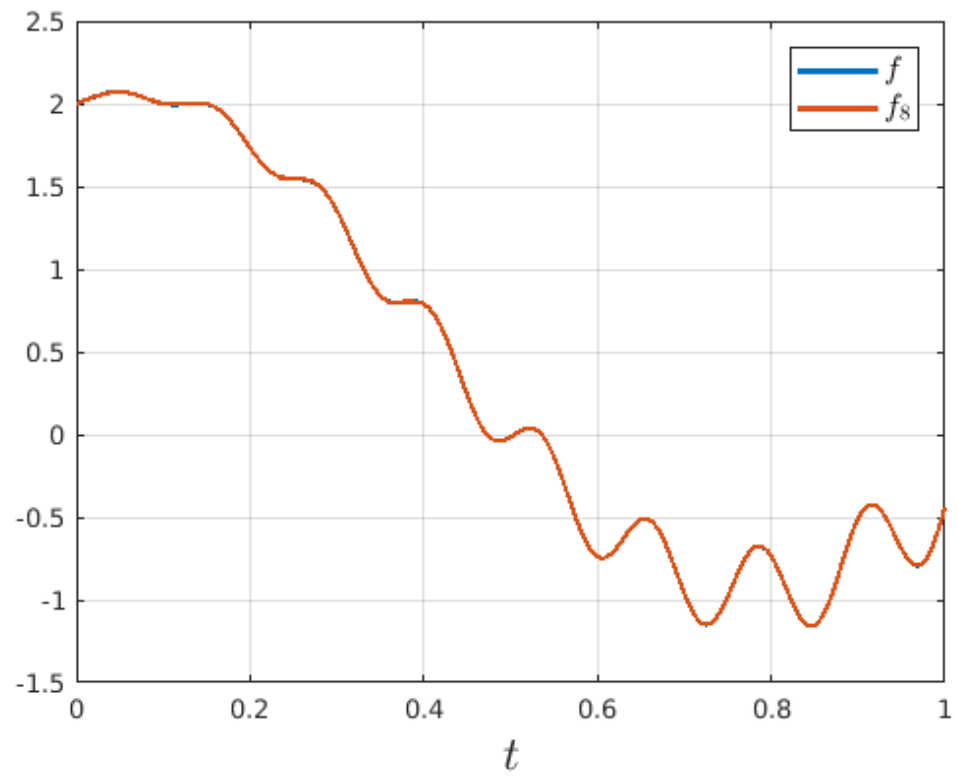
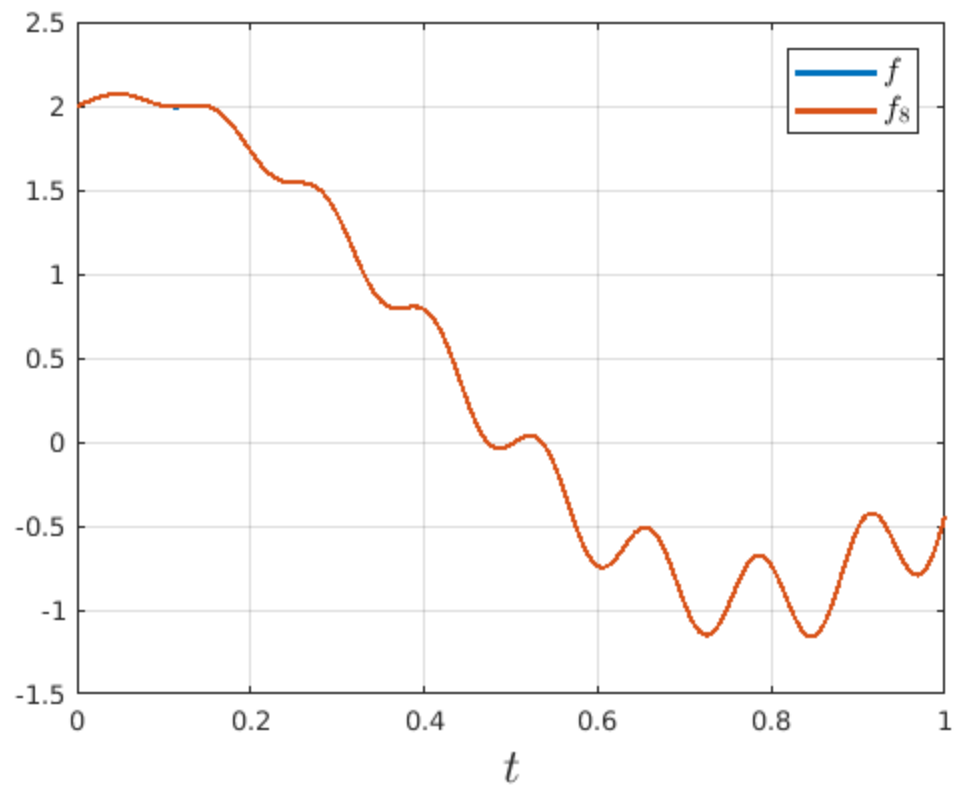


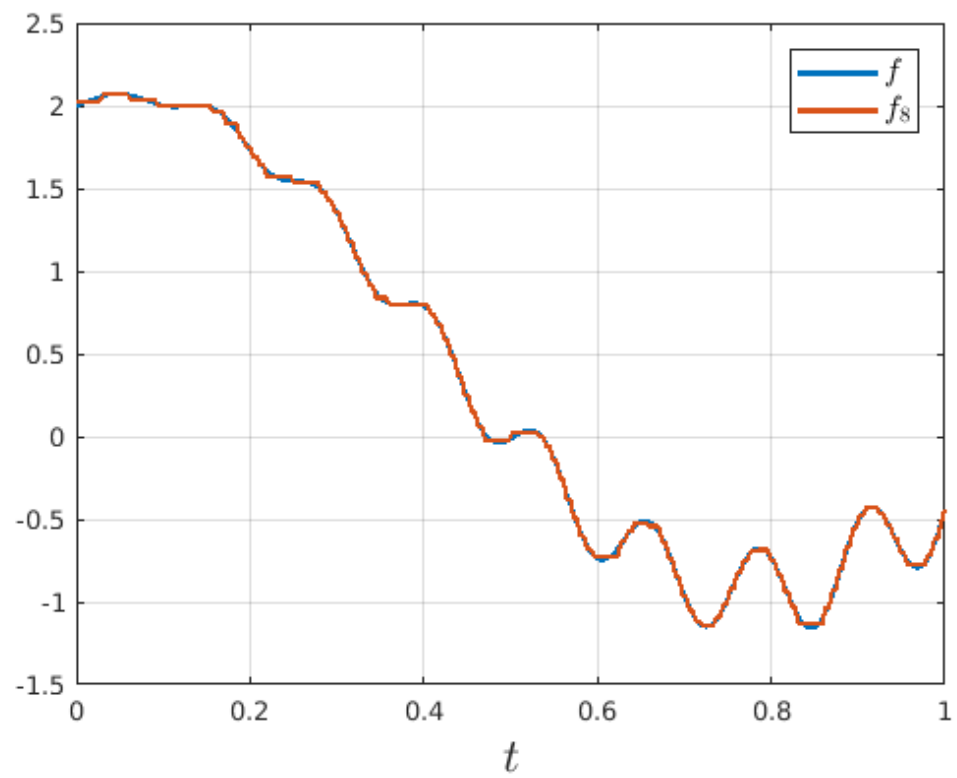
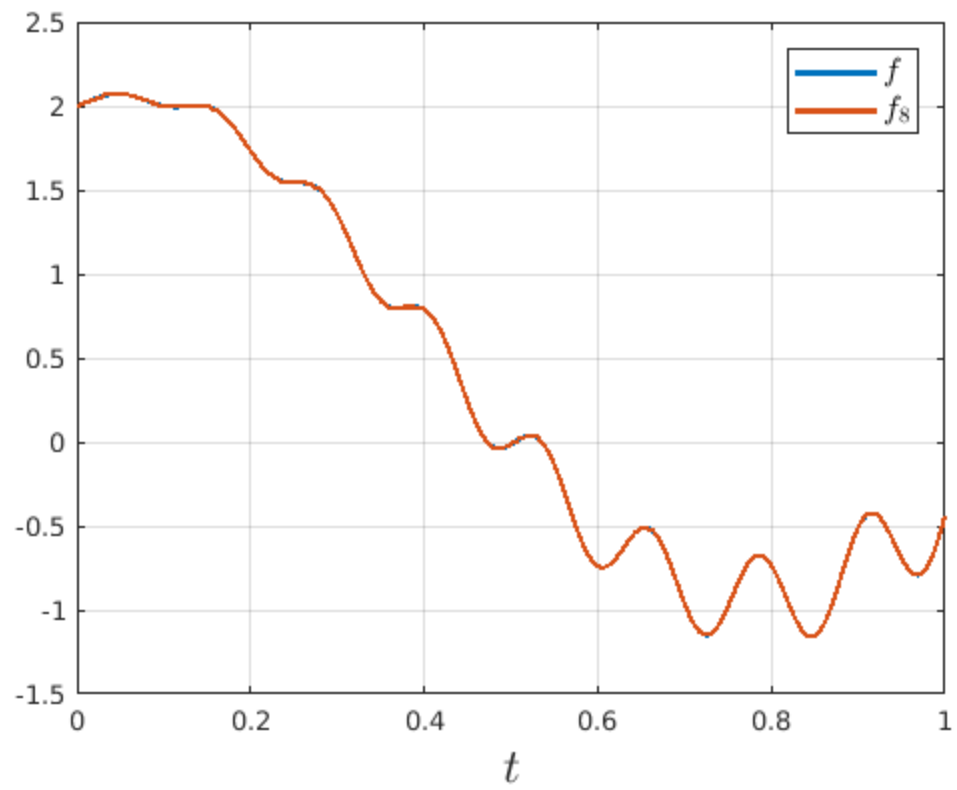
As we can see in the figures, the further we take the decomposition, the more inaccurate the approximation. This is expected because for lower j s the decomposed signal does not catch the higher frequencies.

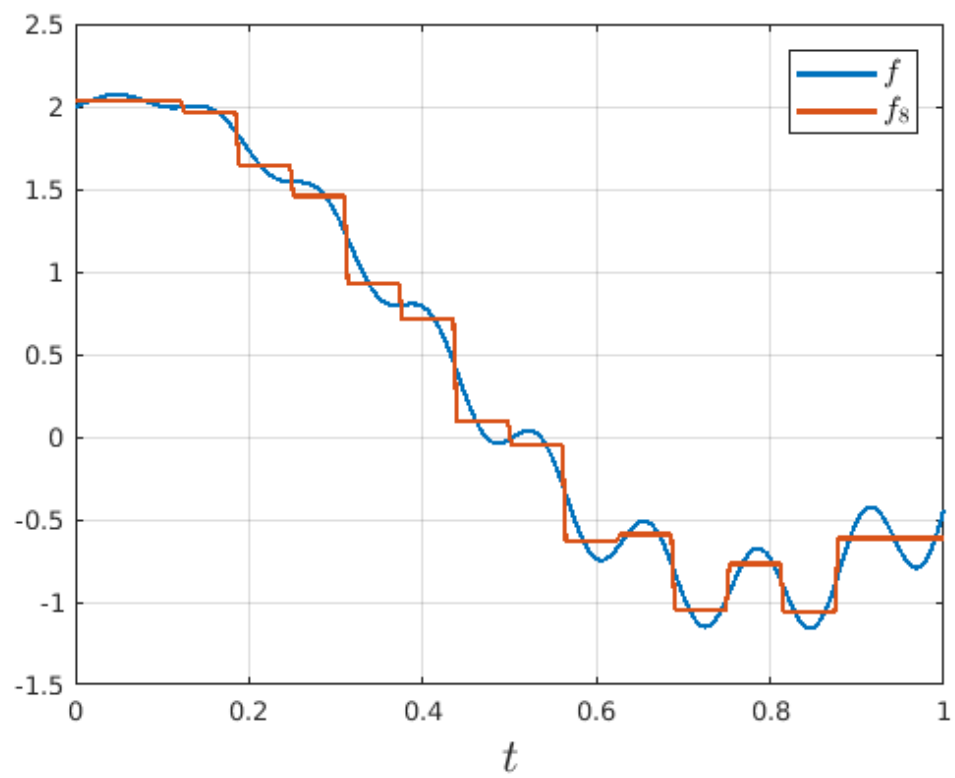
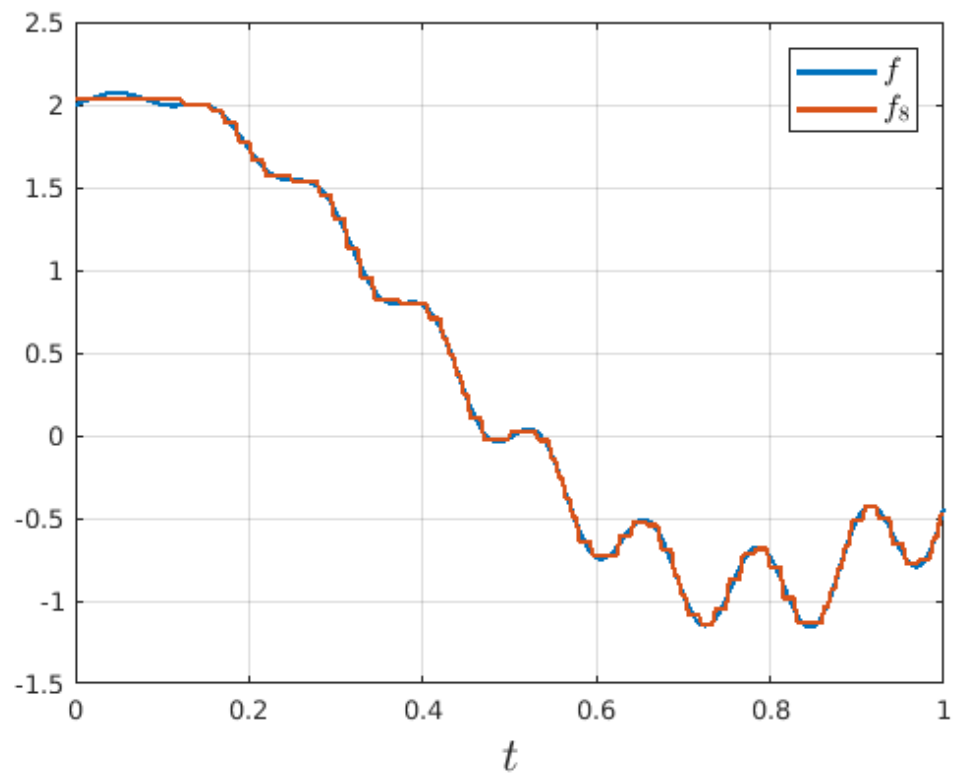
Compressing the signal (Problem 10)

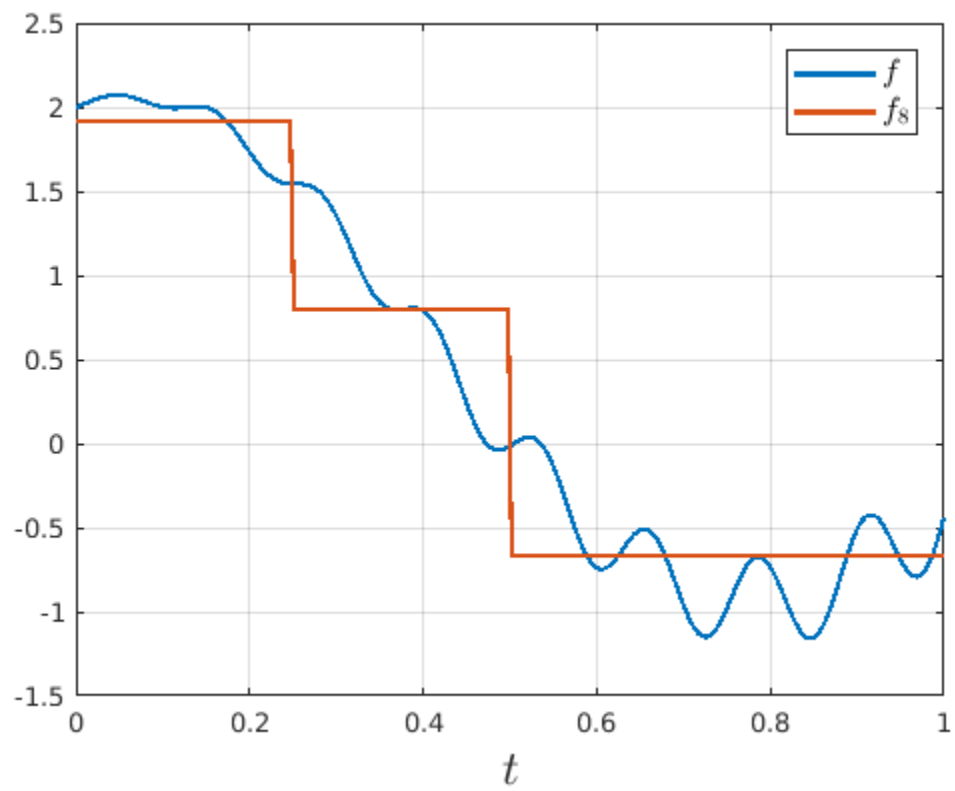
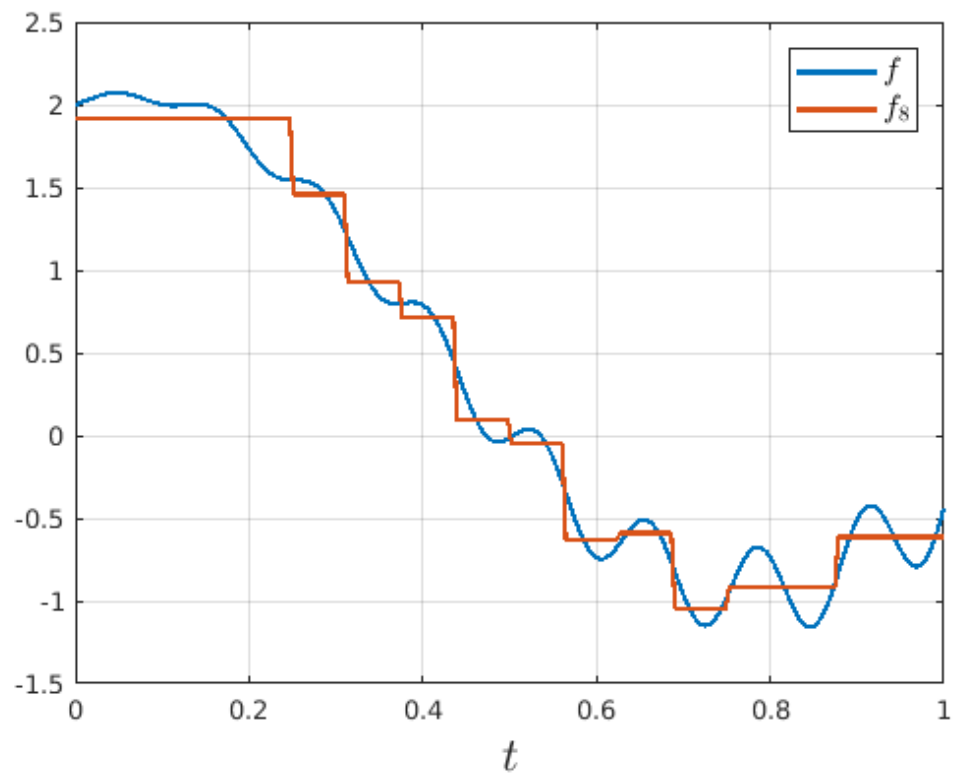
In this section we will filter out those coefficients smaller than a preset tolerance by setting them to zero. Once that is done, we will reconstruct the signal and compare it to the original one.

```
N=2^8;
tol=[1e-3 5e-3 1e-2 5e-2 1e-1 5e-1 1 5];
for i=1:length(tol)
    count(i)=0;
    [C,L]=wavedec(y,8,'db1');    %level 8 decomposition; analogous to
    full fft
    for j=1:N
        if (abs(C(j))<tol(i))
            C(j)=0;
            count(i)=count(i)+1;
        end
    end
    yc=waverec(C,L,'db1');    %reconstruct - analogous to ifft
    figure
    plot(t,y,t,yc,'linewidth',linewidth); % Plot original function and
    the i-th level decomposition
    grid on
    xlabel('$t$', 'interpreter','latex','fontsize',labelfontsize)
    axis([0 1 -1.5 2.5])
    h=legend('$f$', '$f_8$');
    set(h, 'interpreter','latex','fontsize',legendfontsize);
    saveas(gcf,['f_tol' num2str(i) '_p10'],'png')
    e(i)=norm(y-yc)/norm(y);
end
```









In the figures above it can be seen how we increase the tolerance and the reconstructed signal gets more and more inaccurate compared to the original signal. I have used the following tolerances:

`tol`

`tol =`

Columns 1 through 7

0.0010 0.0050 0.0100 0.0500 0.1000 0.5000 1.0000

Column 8

5.0000

Which have produced the removal of the following number of coefficients, respectively:

`count`

`count =`

9 40 65 173 200 242 245 253

This represent, respectively, the following percentages of the total number of data points

`percentage=count*100/N`

`percentage =`

Columns 1 through 7

3.5156 15.6250 25.3906 67.5781 78.1250 94.5312 95.7031

Column 8

98.8281

To finish, the relative error obtained is detail below:

`e`

`e =`

Columns 1 through 7

0.0001 0.0009 0.0021 0.0158 0.0258 0.0843 0.1161

Column 8

0.2908

Published with MATLAB® R2017a