1. **Characterization of local regularity:**

In Section 5.1.2, we have seen how the continuous wavelet transform can characterize the local regularity of a function. Take the Haar wavelet for simplicity.

(a) Consider the function

\[ f(t) = \begin{cases} 
  t & 0 \leq t, \\
  0 & t < 0,
\end{cases} \]

and show, using arguments similar to the ones used in the text, that

\[ CWT_f(a, b) \approx a^{3/2}, \]

around \( b = 0 \) and for small \( a \).

(b) Show that if

\[ f(t) = \begin{cases} 
  t^n & 0 \leq t, \ n = 0, 1, 2 \ldots \\
  0 & t < 0,
\end{cases} \]

then

\[ CWT_f(a, b) \approx a^{(2n+1)/2}, \]

around \( b = 0 \) and for small \( a \).

2. **Nondownsampled filter bank:** Refer to Figure 3.1 without downsamplers.

(a) Choose \( \{H_0(z), H_1(z), G_0(z), G_1(z)\} \) as in an orthogonal two-channel filter bank. What is \( y[n] \) as a function of \( x[n] \)? Note: \( G_0(z) = H_0(z^{-1}) \) and \( G_1(z) = H_1(z^{-1}) \), and assume FIR filters.

(b) Given the “energy” of \( x[n] \), or \( \|x\|^2 \), what can you say about \( \|x_0\|^2 + \|x_1\|^2 \)? Give either an exact expression, or bounds.

(c) Assume \( H_0(z) \) and \( G_0(z) \) are given, how can you find \( H_1(z), G_1(z) \) such that \( y[n] = x[n] \)? Calculate the example where

\[ H_0(z) = G_0(z^{-1}) = 1 + 2z^{-1} + z^{-2}. \]

Is the solution \( (H_1(z), G_1(z)) \) unique? If not, what are the degrees of freedom? Note: In general, \( y[n] = x[n-k] \) would be sufficient, but we concentrate on the zero-delay case.

3. **Frame analysis of the Laplacian pyramid:**

Laplacian pyramid (LP) is an efficient multiscale data representation, proposed by Burt and Adelson in 1983 for image coding. Figure 1 depicts the scheme to compute the LP for an input signal, together with the usual reconstruction algorithm. Clearly, the LP is an overcomplete representation (there are more coefficients after the analysis than in the input), thus it can be treated as a frame operator. Consider a simple case of the LP with Haar filters: \( \hat{h} = g = (1/\sqrt{2}, 1/\sqrt{2}) \) and \( M = 2 \).

(a) Compute the tightest possible frame bounds for the LP in this case.
(b) Find the dual frame operator for the LP in this case. Compare it with the usual reconstruction scheme.

(c) The advantage of using the dual frame for reconstruction comes when the LP data are noisy (for instance due to quantization or thresholding). Assume the following simple noise model: \( \hat{y} = y + \eta \), where

i. Each noise component \( \eta_i \) has mean zero and variance \( \sigma^2 \).

ii. The noise components are uncorrelated.

These can expressed as:

\[
E[\eta_i] = 0 \quad \text{and} \quad E[\eta_i \eta_j] = \delta[i-j] \sigma^2.
\]

Compute the mean square error (MSE) of the reconstructed signal from the noisy LP data \( \hat{y} \) using the usual method and the dual frame operator.

\[
\begin{align*}
\text{(a)} & \quad \text{Figure 1: (a) Laplacian pyramid scheme. The outputs are a coarse approximation } y_0 \text{ and a difference } y_1 \text{ between the original signal and the prediction. The scheme can be iterated by decomposing the coarse version repeatedly. (b) The usual reconstruction scheme for the Laplacian pyramid.}
\end{align*}
\]

4. Matlab exercise - Nonlinear Approximation and Compression

In this exercise we will investigate a simple example on how different approaches for approximate representations of signals in a given space influence the reconstruction. In particular, we will consider the contrast between linear and non-linear approximation methods.

Assume a space \( V \) and an orthonormal basis \( \{g_n\} \) for \( V \). Thus, a function \( f \in V \) can be written as a linear combination

\[
f = \sum_{n \in N} \langle g_n, f \rangle g_n
\]

Suppose we want to approximate the function \( f \) by using only \( M \) vectors (and correspondingly coefficients) of the basis spanning the space \( V \). The linear approximation \( \hat{f} \) of \( f \) is given by the orthogonal projection of \( f \) onto a fixed subspace of \( V \). You can find a detailed explanation on what the optimal decomposition subspace is, when given a set of objects, in the lecture notes and the references. Obviously, since the subspace decomposition is fixed, the method is linear, that is the relation

\[
\hat{f} + \hat{g} = \hat{f + g}
\]

holds for any two functions in \( V \).
Instead, we can choose the most significative $M$ vectors (and the respective coefficients) for each object in part. In this case, the projection subspace is not fixed for all the objects anymore, the decomposition being dependent on the respective object. Then the condition for linearity (2) does not hold any longer.

In this Matlab exercise you will analyze how Fourier and wavelet bases behave for these two different scenarios, for a simple example.

Your task

Consider a 1-D signal of length $N = 1024$. You can download the Matlab procedure `generate.m` that generates the test signal that will be used for this exercise from the exercise section in the webpage of the course. This signal has to be compressed in the two different bases, Fourier and wavelet ($D_2$ Daubechies 4-tap) by using linear and non-linear approximation for each of the cases. Compression is done by zero-ing out the expansion coefficients corresponding to basis vectors left out of approximation. Reconstruction is done by the respective inverse transform using only the significant coefficients (the ones kept after approximation). The approximation will be done using $M = 64$ coefficients for each method.

- **Fourier basis with linear approximation.** For this method you have to keep the $M$ low frequencies coefficients and zero out the $N - M$ coefficients corresponding to high frequencies. **Hints:** use the `fft` function, and zero out the coefficients that are outside the $M$-coefficient interval that centers the spectrum you obtain with `fftshift`; reconstruct the signal using `ifft`.

- **Fourier basis with non-linear approximation.** The procedure is similar to the previous method, except that instead of low frequencies, you will approximate the signal using the largest absolute value $M$ coefficients of the Fourier decomposition.

- **Wavelet basis with linear approximation.** Expand the signal in a 4-tap Daubechies wavelet basis, on $J = 6$ levels. Keep $M$ coefficients corresponding to the low frequency (coarsest) part of the signal, zero-out the rest. Reconstruct the signal. **Hints:** use `wavedec` and `waverec` for the decomposition/reconstruction procedures. The help Matlab gives you on these functions will actually do most of the job for you.

- **Wavelet basis with non-linear approximation.** Use a similar procedure as with the previous method. Like in the Fourier case, keep now only most significant $M$ coefficients from the whole decomposition. Same hints apply.

Plot on one graph the original function and its compressed versions by using the two Fourier based approximation/compression/reconstruction schemes. Plot on the second graph the original function and its compressed versions by using the two wavelet based approximation/compression/reconstruction schemes. Compute the corresponding MSEs between the original and the reconstructed versions of the signal. Which scheme performs the best? Comment on the result. What is the difference between the two decompositions in the case of signals with discontinuities?

A taste of wavelet denoising

The idea behind denoising with wavelets is the fact that less significant (small valued) coefficients in a decomposition would correspond to noise. Thus, by zeroing them out, the noise is also removed. Consider again the original signal generated by the code in `generate.m`. Add Gaussian noise to it, of variance $\sigma = 0.1$ and zero mean. Consider a wavelet decomposition on $J = 4$ levels. Apply the same procedure as for the non-linear wavelet basis approximation case
in the previous section, only now instead of zeroing out a fixed number of coefficients, do this for coefficients smaller in absolute value than a given threshold $T$, and keep only coefficients larger than that value for the reconstruction. Use different values for $T$, say $T = 0, 0.1, \ldots, 1$. Do the reverse transform, and compare the result with the original signal you had before adding noise. Plot the original signal and the reconstructed denoised version for the value of $T$ that seems visually optimal to you. Comment on the result.

References: