Algorithms of Scientific Computing  
(Algorithmen des Wissenschaftlichen Rechnens)  

Haar Wavelets

The wavelet families we look at (e.g. Haar wavelets) are constructed around a multiresolution analysis, a nested sequence $V_n$ of function spaces some of which properties are

$$V_j \subset V_{j+1}, j \in \mathbb{Z}$$  \hspace{1cm} (1)

$$\bigcap_{j=-\infty}^{\infty} V_j = \{0\}$$  \hspace{1cm} (2)

$$f(t) \in V_l \iff f(2^{-l}t) \in V_0$$  \hspace{1cm} (3)

$$V_l = V_{l-1} \oplus W_{l-1}$$
$$= V_{l-2} \oplus W_{l-2} \oplus W_{l-1}$$
$$= V_0 \oplus W_0 \oplus W_1 \oplus \ldots \oplus W_{l-1},$$  \hspace{1cm} (4)

with orthogonal functions $f \in V_j$ and $g \in W_j$, i.e. $<f,g> = 0$.

The theory of multiresolution analysis further states the existence of a unique function $\phi$ which satisfies a so-called dilation equation of the form

$$\phi(t) = \sum_{k \in \mathbb{Z}} c_k \cdot \phi(2t - k).$$  \hspace{1cm} (5)

Define another function, known as the mother wavelet or the wavelet function of the form

$$\psi(t) := \sum_{k \in \mathbb{Z}} (-1)^k c_{1-k} \cdot \phi(2t - k).$$  \hspace{1cm} (6)

With the help of $\phi$ and $\psi$, we can define orthonormal nodal bases $\{\phi_{l,k}\}$ for $V_l$ with

$$\phi_{l,k}(t) = \phi(2^l t - k)$$
$$\text{span}\{\phi_{l,k}\} = V_l, \quad <\phi_{l,k},\phi_{l,m}> = \delta_{k,m} \quad k,m \in \mathbb{Z}.$$  \hspace{1cm} (7)

The function $\phi$ is called father wavelet or the scaling function, and together with a mother wavelet $\psi$, they define the wavelet family. It is not necessary to know a specific formula for $\phi$, the dilation equation (5) with its coefficients $c_k$ together with the theory of multiresolution analysis provide enough information to derive the mother wavelet $\psi$ as well as orthonormal wavelet bases $\{\psi_{l,m}\}$ for the $W_l$ with

$$\psi_{l,k}(t) = \psi(2^l t - k)$$
$$\text{span}\{\psi_{l,k}\} = W_l, \quad <\psi_{l,k},\psi_{l,m}> = \delta_{k,m} \quad k,m \in \mathbb{Z}.$$  \hspace{1cm} (8)
Excercise 1: Cranking the Machine

Typically the scaling function $\phi$ is not known explicitly, and sometimes a closed-form analytic formula does not even exist. However, for continuous $\phi$ we can approximate the function to arbitrarily high precision using the “Cascade Algorithm”, a fixed-point method for functions.

In this exercise we want to implement this algorithm by iterating over the expression

$$F(\gamma)(t) = \sum_k c_k \cdot \gamma(2t-k) \quad (9)$$

in order to find the fixed point $\gamma$ of $F$. That is, at iteration $n$

$$\gamma_{n+1}(t) = \sum_k c_k \cdot \gamma_n(2t-k) \quad (10)$$

Our starting point $\gamma_0$ will be the hat function $\psi_0(t) = \max\{1 - |x|, 0\}.$

(i) Over the interval $[-1,4]$ plot the approximations of the scaling function $\phi$ for the Haar wavelet family obtained in the first 7 iterations of the cascade algorithm. Do so by plugging the refinements coefficients $c_k, k = 0,1$ in (12) into (9) resp. (5).

$$c_0 = c_1 = 1 \quad (12)$$

(ii) Over the interval $[-1,4]$ plot the approximations of the scaling function $\phi$ for the Daubechies wavelet family obtained in the first 7 iterations of the cascade algorithm. Do so by plugging the refinements coefficients $c_k, k = 0,\ldots,3$ in (13) into (9) resp. (5).

$$c_0 = 1 + \frac{\sqrt{3}}{4} \quad c_1 = \frac{3 + \sqrt{3}}{4} \quad c_2 = \frac{3 - \sqrt{3}}{4} \quad c_3 = \frac{1 - \sqrt{3}}{4} \quad (13)$$

Excercise 2: The Haar Wavelet Basis

We derive the mother wavelet $\psi$ as well as orthonormal wavelet bases $\{ \psi_{l,m} \}$ with

$$\psi_{l,k}(t) = \psi(2^l t - k) \quad \text{span}\{ \psi_{l,k} \} = W_l, \quad \langle \psi_{l,k}, \psi_{l,m} \rangle = \delta_{k,m} \quad k, m \in \mathbb{Z}. \quad (14)$$

In this exercise we want to compute the 1-d wavelet transform for the Haar wavelet family and apply it to a signal vector $\tilde{s}$ of length $m = 2^n$. The transform can be implemented very efficiently as a “pyramidal algorithm” taking $\Theta(m)$ steps. For educational purpose we focus on the $\Theta(m^2)$ matrix-based algorithm.

(i) Write a function that constructs the transformation matrix $M$ consisting of the basis vectors $\psi_{l,k}, l \leq n, 0 \leq k \leq 2^n - 1$.

(ii) Use Python’s package numpy.linalg to invert the matrix.

(iii) Use the program to compute the transform $\tilde{d} = M^{-1}\tilde{s}$ as well as the reconstructed signal $\tilde{s} = M\tilde{d}$ of the vector $\tilde{s} = [1,2,3,-1,1,-4,-2,4]^T$.

(iv) Verify the program’s output tracing the steps by hand.