Wavelets with applications in signal and image processing

Adhemar Bultheel

October 26, 2006

Contents

Chapter 1

Introduction

The basic ideas which will be explained in this introduction are presented graphically in Figure 1.1. We have a signal consisting of 8 samples: $x = (1, 3, 5, 7, 2, -1, -3, -2)$ (top left). This can be written as 1 times a first 1×1 block + 3 times a second 1×1 block + 5 times a third 1×1 block $+ \cdots$ (top right). These eight 1×1 blocks are basis functions and the signal is a linear combination of these 8 basis functions. Next we replace every 2 samples by their average (middle left). To find the original signal, we also have to store the error, i.e., the difference between the original signal and the averaged signal (middle right). The averaged signal is denoted $x^{(-1)}$ and the difference signal is denoted $y^{(-1)}$. Thus $x = x^{(-1)} + y^{(-1)}$. The averaged signal $x^{(-1)}$ can be written as a linear combination of 2×1 blocks (bottom left) while the difference signal $y^{(-1)}$ can be witten as a linear combination of the functions plotted at the bottom right. Thus, the original signal can also be written as a linear combination of the 8 functions given at the bottom (4 on the left and 4 on the right). Thus we have made a change of basis from the 8 basis functions at the top right to the $4 + 4$ basis functions at the bottom.

Let us now work this out in a slightly more formal way. Consider a discrete signal that consists of 8 sample values. In the previous example:

$$x = (x_1, x_2, \ldots, x_8) = (1, 3, 5, 7, 2, -1, -3, -2).$$

The space V_0 of all such signals is in fact the space $\mathbb{R}^8$, namely the space of all 8-tupples. Note that this is a Euclidean space with inner product $\langle X, Y \rangle = X^t Y$. This space can be described by the natural basis $\{\varphi_{0k}\}_{k=1}^{8}$, where $\varphi_{0k} = e_k$ is the kth column of the 8×8 unit matrix: the columns of the matrix $[E^{(0)}] = I_8$. Thus, setting $H^{(0)} = I_8$, $x = x^{(0)} = [H^{(0)}]^t X^{(0)}$.

Such a basis vector can be represented graphically as a 1×1 block at position k. The values x_k are the coordinates of x with respect to these basis vectors. We denote this coordinate vector as $X = X^{(0)} = [1, 3, 5, 7, 2, -1, -3, -2]^t$. Note that the basis is orthonormal since

$$[\varphi_{0p}]^t [\varphi_{0q}] = \delta_{p-q}, \quad p, q = 1, \ldots, 8.$$

Now consider the subspace V_{-1} which is spanned by the basis functions

$$\varphi_{-1,k} = \frac{1}{\sqrt{2}}(\varphi_{0,2k-1} + \varphi_{0,2k}), \quad k = 1, 2, 3, 4$$

Figure 1.1: Averaging and differencing

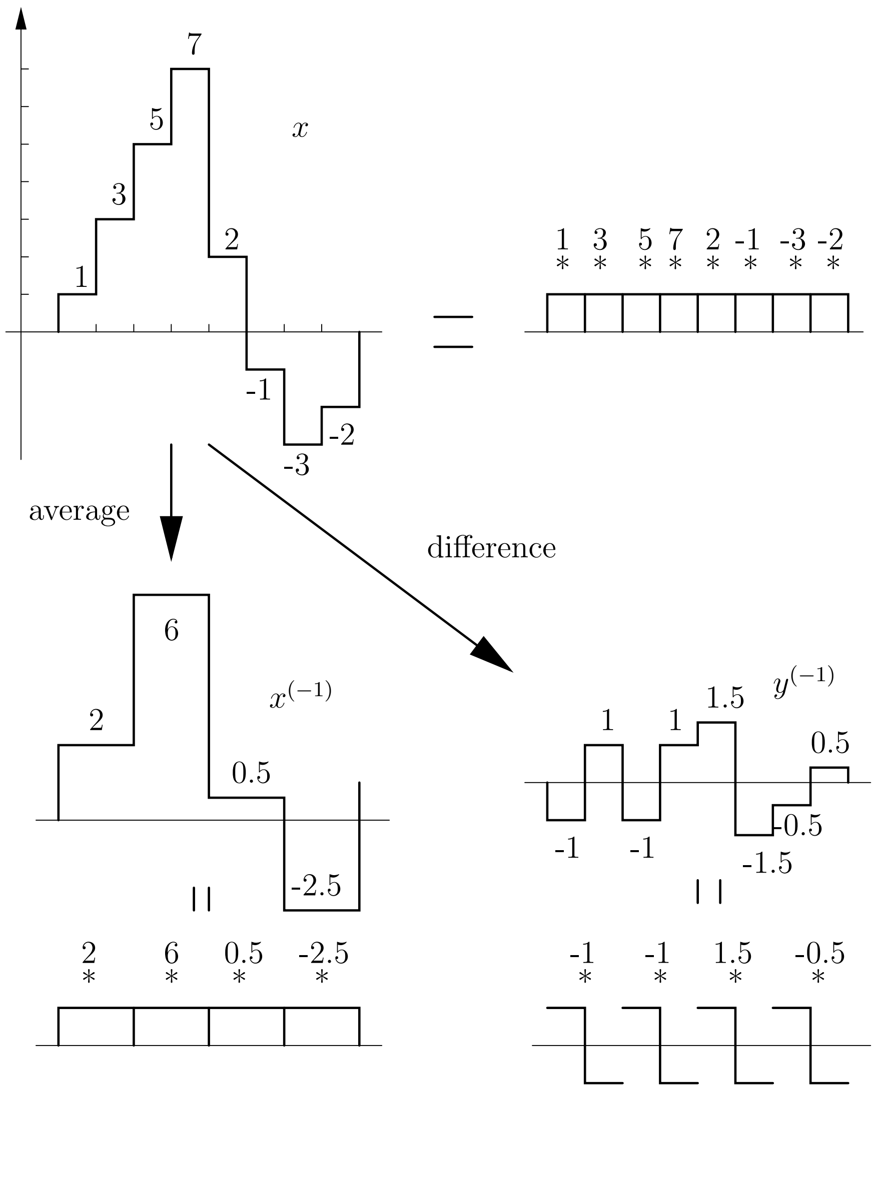
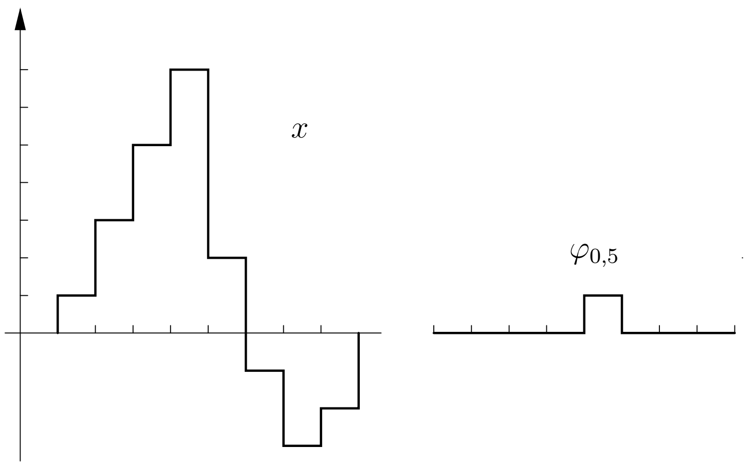


Figure 1.2: Signal and basis function



Defining the matrix

$$H^{(-1)} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 & & & & & & \\ & & 1 & 1 & & & & \\ & & & & 1 & 1 & & \\ & & & & & & 1 & 1 \end{bmatrix}$$

then it is clear that the basis vector $\varphi_{-1,k}$ is the kth column of the matrix $[H^{(-1)}]^t$. Obviously, these basis functions are again orthonormal: $[H^{(-1)}][H^{(-1)}]^t = I_4$. Thus, we can easily find the projection $x^{(-1)}$ of x onto the subspace V_{-1}. It is given by $x^{(-1)} = [H^{(-1)}]^t X^{(-1)}$ with $X^{(-1)} = H^{(-1)} X^{(0)}$. Thus

$$X^{(-1)} = \frac{1}{\sqrt{2}}[x_1 + x_2, x_3 + x_4, x_5 + x_6, x_7 + x_8]^t.$$

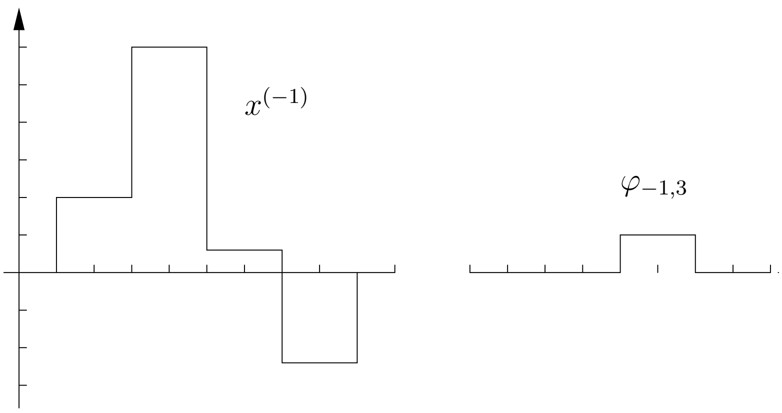
This projection gives a low resolution picture of the original signal because in the original picture we could distinguish between function values that were 1 unit apart in the horizontal direction, while in the new picture we have a resolution of 2 units in the horizontal direction. If the original signal is plotted on a screen that can not distinguish between two adjacent pixels in the horizontal direction, then Figure 1.3 is the best we can get as a picture of the signal. It is a low resolution approximation of the original.

By this projection we loose some information. This is captured in the orthogonal complement of V_{-1} in V_0. We denote this space as $W_{-1} = V_0 \ominus V_{-1}$. This space W_{-1} is also of dimension 4 and it is spanned by the basis vectors

$$\psi_{-1,k} = \frac{1}{\sqrt{2}}(\varphi_{0,2k-1} - \varphi_{0,2k}), \quad k = 1, 2, 3, 4.$$

We collect these basis vectors in a matrix. So they are given as the columns of the matrix

Figure 1.3: First projection and 1 basis function

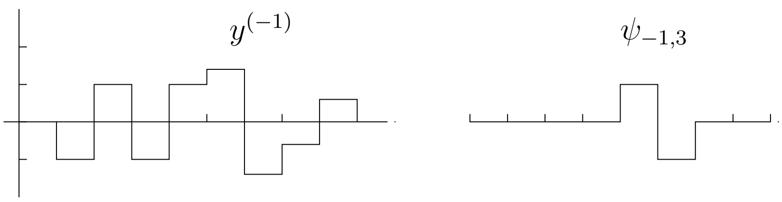


$[G^{(-1)}]^t$ where

$$G^{(-1)} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -1 & & & & & & \\ & & 1 & -1 & & & & \\ & & & & 1 & -1 & & \\ & & & & & & 1 & -1 \end{bmatrix}.$$

Note that also this basis is orthonormal since $[G^{(-1)}][G^{(-1)}]^t = I_4$. We have now decomposed our original signal x into an orthogonal direct sum $x = x^{(-1)} + y^{(-1)}$ where $x^{(-1)}$ is the projection of x onto V_{-1} as given above and $y^{(-1)}$ is the projection of x onto W_{-1}. This projection is given by $y^{(-1)} = [G^{(-1)}]^t Y^{(-1)}$ with $Y^{(-1)} = G^{(-1)} X$. This is depicted in Figure 1.4. The sum of the pictures 1.3 and 1.4 would give the Figure 1.2 back (up to a scaling

Figure 1.4: Projection on orthogonal complement and 1 basis function



factor).

Mathematically, we have represented the original signal with repect to a new basis, namely the columns of the matrix $E^{(-1)} = I_8[T^{(-1)}]^t$ where $(T^{(-1)})^t = [(H^{(-1)})^t \ (G^{(-1)})^t]$ describes the basis transformation. The new coordinates for this basis are given by

$$\begin{bmatrix} X^{(-1)} \\ Y^{(-1)} \end{bmatrix} = \begin{bmatrix} H^{(-1)} \\ G^{(-1)} \end{bmatrix} X = T^{(-1)} X.$$

Now suppose that the low resolution picture of Figure 1.3 is still too high for the screen we have available, then we can repeat the same kind of operation to this picture. Thus, we define the subspace V_{-2} of V_{-1} which is spanned by the orthonormal basis vectors

$$\varphi_{-2,k} = \frac{1}{\sqrt{2}}(\varphi_{-1,2k-1} + \varphi_{-1,2k}), \quad k = 1, 2.$$

and project $x^{(-1)}$ onto V_{-2}. This gives $x^{(-2)}$ defined as $x^{(-2)} = [\varphi_{-2,1}\ \varphi_{-2,2}]X^{(-2)}$ where $X^{(-2)}$ is obtained by $X^{(-2)} = H^{(-2)}X^{(-1)}$ and where

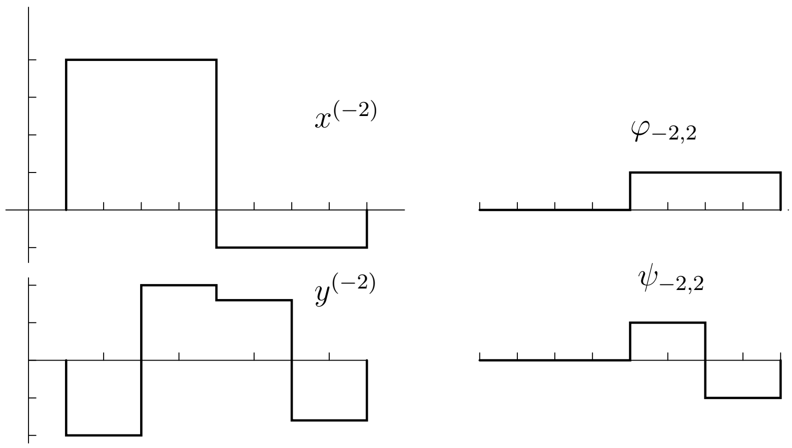
$$H^{(-2)} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 & & \\ & & 1 & 1 \end{bmatrix}.$$

If $W_{-2} = V_{-1} \ominus V_{-2}$, then this orthogonal complement is spanned by the vectors which are the columns of the matrix $[G^{(-2)}H^{(-1)}]^t$ with

$$G^{(-2)} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & -1 & & \\ & & 1 & -1 \end{bmatrix}.$$

and the projection of $x^{(-1)}$ onto W_{-2} is given by $y^{(-2)} = [G^{(-2)}H^{(-1)}]^t Y^{(-2)}$ with $Y^{(-2)} = G^{(-2)}X^{(-1)}$. This process can be repeated just one more time (since we run out of data). The

Figure 1.5: Second decomposition



space V_{-3} is one-dimensional and is generated by the vector $\varphi_{-3,1} = \frac{1}{\sqrt{2}}(\varphi_{-2,1}+\varphi_{-2,2})$, which has all entries equal to $2^{-3/2}$. The coordinate is $X^{(-3)} = H^{(-3)}X^{(-2)}$ with $H^{(-3)} = \frac{1}{\sqrt{2}}[1\ 1]$. This $X^{(-3)}$ is some kind of avarage of the signal because it equals $2^{-3/2}(x_1 + x_2 + \cdots + x_8)$. Similarly, $W_{-3} = V_{-2} \ominus V_{-3}$ is one-dimensional and is spanned by the vector $\psi_{-3,1} = \frac{1}{\sqrt{2}}(\varphi_{-2,1} - \varphi_{-2,2})$ which is equal to $2^{-3/2}$ in its first 4 entries and equal to $-2^{-3/2}$ in its last 4 entries. The coordinate of the projection $y^{(-3)}$ is $Y^{(-3)} = G^{(-3)}X^{(-2)}$ with $G^{(-3)} = \frac{1}{\sqrt{2}}[1\ -1]$.

Finally we have decomposed the signal as

$$x = x^{(-3)} + y^{(-3)} + y^{(-2)} + y^{(-1)} = x^{(-2)} + y^{(-2)} + y^{(-1)} = x^{(-1)} + y^{(-1)}.$$

For increasing k, $x^{(k)}$ gives better and better approximations of x, i.e., they are approximations of x at a higher and higher resolution level. We have obtained this decomposition by successive orthogonal basis transformations. The successive bases we used for V_0 are given by

$$\begin{array}{ll}
E^{(0)} & = [\ \varphi_{0,1},\ \varphi_{0,2},\ \varphi_{0,3},\ \varphi_{0,4},\ \varphi_{0,5},\ \varphi_{0,6},\ \varphi_{0,7},\ \varphi_{0,8}\ ] \\
E^{(-1)} & = [\ \varphi_{-1,1},\ \varphi_{-1,2},\ \varphi_{-1,3},\ \varphi_{-1,4} |\ \psi_{-1,1},\ \psi_{-1,2},\ \psi_{-1,3},\ \psi_{-1,4}\ ] \\
E^{(-2)} & = [\ \varphi_{-2,1},\ \varphi_{-2,2} |\ \psi_{-2,1},\ \psi_{-2,2} |\ \psi_{-1,1},\ \psi_{-1,2},\ \psi_{-1,3},\ \psi_{-1,4}\ ] \\
E^{(-3)} & = [\ \varphi_{-3,1} |\ \psi_{-3,1} |\ \psi_{-2,1},\ \psi_{-2,2} |\ \psi_{-1,1},\ \psi_{-1,2},\ \psi_{-1,3},\ \psi_{-1,4}\ ]
\end{array}$$

which are given by the columns of the matrices $E^{(0)} = I_8$,

$$E^{(-1)} = \frac{1}{\sqrt{2}} \left[\begin{array}{cccc|cccc}
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & -1
\end{array}\right]$$

$$E^{(-2)} = \frac{1}{2} \left[\begin{array}{cc|cc|cccc}
1 & 0 & 1 & 0 & \sqrt{2} & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & -\sqrt{2} & 0 & 0 & 0 \\
1 & 0 & -1 & 0 & 0 & \sqrt{2} & 0 & 0 \\
1 & 0 & -1 & 0 & 0 & -\sqrt{2} & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 0 & \sqrt{2} & 0 \\
0 & 1 & 0 & 1 & 0 & 0 & -\sqrt{2} & 0 \\
0 & 1 & 0 & -1 & 0 & 0 & 0 & \sqrt{2} \\
0 & 1 & 0 & -1 & 0 & 0 & 0 & -\sqrt{2}
\end{array}\right]$$

$$E^{(-3)} = \frac{1}{2\sqrt{2}} \left[\begin{array}{c|c|cc|cccc}
1 & 1 & \sqrt{2} & 0 & 2 & 0 & 0 & 0 \\
1 & 1 & \sqrt{2} & 0 & -2 & 0 & 0 & 0 \\
1 & 1 & -\sqrt{2} & 0 & 0 & 2 & 0 & 0 \\
1 & 1 & -\sqrt{2} & 0 & 0 & -2 & 0 & 0 \\
1 & -1 & 0 & \sqrt{2} & 0 & 0 & 2 & 0 \\
1 & -1 & 0 & \sqrt{2} & 0 & 0 & -2 & 0 \\
1 & -1 & 0 & -\sqrt{2} & 0 & 0 & 0 & 2 \\
1 & -1 & 0 & -\sqrt{2} & 0 & 0 & 0 & -2
\end{array}\right]$$

The corresponding coordinates are given by

$$X^{(0)} = X, \quad \begin{bmatrix} X^{(-1)} \\ Y^{(-1)} \end{bmatrix}, \quad \begin{bmatrix} X^{(-2)} \\ Y^{(-2)} \\ Y^{(-1)} \end{bmatrix}, \quad \begin{bmatrix} X^{(-3)} \\ Y^{(-3)} \\ Y^{(-2)} \\ Y^{(-1)} \end{bmatrix}.$$

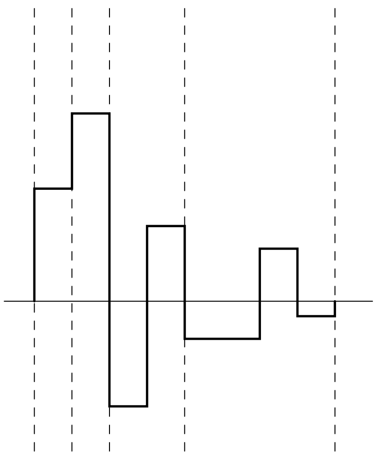
Thus multiplying these vectors from the left with the respective basis matrices $E^{(0)}$, $E^{(-1)}$, $E^{(-2)}$, $E^{(-3)}$, will all give the same signal x.

In our example these coordinates are respectively

$$\begin{bmatrix} 1 \\ 3 \\ 5 \\ 7 \\ 2 \\ -1 \\ -3 \\ -2 \end{bmatrix}; \quad \frac{1}{\sqrt{2}} \left[\begin{array}{c} 4 \\ 12 \\ 1 \\ -5 \\ \hline -2 \\ -2 \\ 3 \\ -1 \end{array}\right]; \quad \frac{1}{2} \left[\begin{array}{c} 16 \\ -4 \\ \hline -8 \\ 6 \\ \hline -2\sqrt{2} \\ -2\sqrt{2} \\ 3\sqrt{2} \\ -\sqrt{2} \end{array}\right]; \quad \frac{1}{2\sqrt{2}} \left[\begin{array}{c} 12 \\ \hline 20 \\ \hline -8\sqrt{2} \\ 6\sqrt{2} \\ \hline -4 \\ -4 \\ 6 \\ -2 \end{array}\right]$$

The last vector is called the wavelet transform of the signal and can be represented graphically by plotting it as a block function. If the signal consists of many thousands of samples, then

Figure 1.6: Wavelet transform



we have the impression of a continuous signal and we get a picture like for example in Figure 1.7 where we used 2048 samples. The signal consists of two sine waves. Note that most of the coefficients in the wavelet transform are zero or very small.

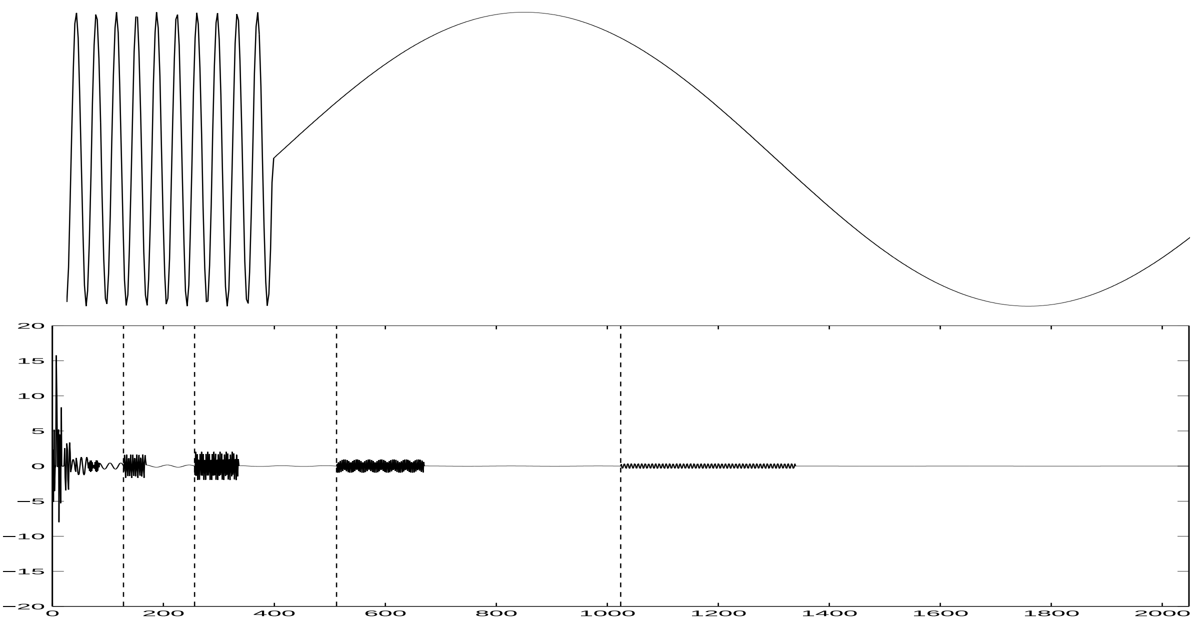
This simple example illustrates several aspects of wavelet analysis which will be studied in this text.

By representing the signal with respect to another basis, we are able to immediately find low resolution approximations of the signal. For example if an image is considered as a two-dimensional signal, it is often interesting to have a low resolution approximation, and only afterwards add more and more detail to it. If this image is transmitted over the internet, then a low resolution can reveal that it is not the image we want and the transmission can be interupted. If it is the right image we can wait for all the details to be transmitted.

Technically such a representation of the signal is called a multiresolution representation.

The basis functions corresponding to low resolutions are in general relatively smooth while the basis functions which catch the detail information are usually less smooth and resemble a short wave (hence the name wavelet) with zero average. If we consider the signal

Figure 1.7: Function and wavelet transform



as depending on time (the index k), then the wavelet functions are "short" (they have a compact support) which means that they are localized in time. If some coordinate is large for a particular basis function, then we know where in the time-domain this large coefficient will have its main influence.

The wavelet basis has also a characteristic that for low resolution the basis functions oscillate less wildly than for high resolution basis functions. This expresses that the low resolution wavelets contain lower frequencies while the high resolution basis functions contain high frequencies. Thus a large coefficient for a high resolution basis function will have influence on the high frequencies which are contained in that basis function while a large coefficient for a low resolution basis function will have influence on the low frequencies which are contained in that basis function. The wavelet basis is also localized in the frequency-domain.

It was also clear that the orthogonality of the basis functions simplified the computation of the coordinates considerably. Therefore we shall try to construct in general a wavelet basis which is orthogonal if possible.

Two other aspects were used here: decimation and filtering. A linear filter applied to a discrete signal will recombine the signal samples to give another signal. For example when we replace x_k by $(Hx)_k = x'_k = \frac{1}{\sqrt{2}}(x_k + x_{k+1})$, then we apply a (moving average) filter to the signal. From our example, we have seen that this gives a lower resolution, i.e., it filters out the higher frequencies and keeps only the lower frequencies. This is called a low-pass filter. On the other hand $(Gx)_k = x''_k = \frac{1}{\sqrt{2}}(x_k - x_{k+1})$ deletes the low frequencies and keeps the higher frequencies. It is a high-pass filter for the given signal, or, since this is recursively applied, it is rather a band-pass filter, which selects a certain frequency band of the signal.

These filters would however transform the signal in a signal of as many samples as the original signal. Thus a signal x of length N is transformed into 2 signals x' and x'' which have both length N, which doubles the number of data. However, as we have seen in our example, it is sufficient to keep only one in every two samples of x' and x'' and this is sufficient to find the original signal back. This is called downsampling or decimation (by a factor 2).

This is a general principle: if we split the signal into m frequency bands by applying a bank of m filters H_k, $k = 1, \ldots, m$, each one selecting the mth part of the whole bandwidth of the signal, then it is sufficient to keep only one out of m samples in each of the results $H_k x$.

The wavelets that appeared in our example above are called Haar wavelets. The coordinates that were obtained in the final stage is again a signal of length 8 (just as the original signal) and it is called the (Haar) wavelet transform of the original signal. We have described the analysis phase, i.e., how the signal is decomposed in its wavelet basis. Reconstruction of the original signal from its wavelet transform is called the synthesis phase. The synthesis consists in undoing the operations from the analysis phase in reverse order. It is important to notice that by our analysis nothing is lost (assuming exact arithmetic) and thus that our signal will be perfectly reconstructed. This is an important property of a filter bank: the perfect reconstruction property.

In the chapters to follow we shall describe all these ideas in a better mathematical[1] framework and in more generality.

[1]A note about the mathematics: We will use mathematics, but proofs or defnitions may not always be completely justified. For example, we often interchange the order of summation of infinite sums or take an infinite power series and call it a function, without knowing that it actually converges etc. We took here an "engineering" point of view where we try to give the main ideas without always being very rigorous. Let us say once and for all that under "appropriate technical conditions", the statements made hold true. By the way, our infinite sums are in practice mostly sums with only a finite number of nonzero terms: filters will have a finite impulse response, signals will be finite length, etc. because these are the only things that can actually be computed.

Chapter 2

Signals

2.1 Fourier transforms

We will consider any real or complex function f of the real variable t (t refers to *time*) to be an analog signal.

If f is real periodic with period T, then $f(t)$ can be expanded in a Fourier series of the form

$$f(t) = a_0 + \sum_{n=1}^{\infty} a_n \cos n\omega_f t + \sum_{n=1}^{\infty} b_n \sin n\omega_f t,$$

where $\omega_f = 2\pi/T$. This ω_f is called the *fundamental frequency* (measured in radians per second). The multiples $n\omega_f$, $n = 0, 1, 2, \ldots$ of the fundamental frequency are called the *harmonic frequencies*. The coefficients are given by (sines and cosines are orthogonal)

$$\begin{aligned} a_0 &= \frac{1}{T}\int_{-T/2}^{T/2} f(t)\ dt \\ a_n &= \frac{2}{T}\int_{-T/2}^{T/2} f(t)\cos n\omega_f t\ dt, \quad n \in \mathbb{N} \\ b_n &= \frac{2}{T}\int_{-T/2}^{T/2} f(t)\sin n\omega_f t\ dt, \quad n \in \mathbb{N}. \end{aligned}$$

If a non-periodic signal f is given in an interval $[-T/2, T/2]$, then we consider it to be periodically extended (the given interval corresponds to one period). Thus, considering $f(t)$ in the interval $t \in [-T/2, T/2]$ only, it can still be represented by a Fourier series of the above form.

Note that, given only a finite piece of a signal (the interval $[-T/2, T/2]$), it becomes impossible to distinguish frequencies which are less apart than

$$\Delta f = \frac{\omega_f}{2\pi} = \frac{1}{T}$$

This Δf is called the (frequency) *resolution* of the data.

The sine/cosine series (called the *trigonometric form* of the Fourier series) can be rearranged into the so called *polar form*

$$f(t) = \sum_{n=0}^{\infty} A_n \cos(n\omega_f t + \varphi_n), \quad A_n = \sqrt{a_n^2 + b_n^2}, \quad \varphi_n = \tan^{-1}(b_n/a_n).$$

A_n is the *magnitude* or *amplitude* and φ_n is called the *phase*. Plotting A_n gives a number of lines, called a (discrete) *amplitude spectrum*, plotting φ_n gives a (discrete) *phase spectrum*. Both are *line spectra* since they consist of a discrete set of lines.

Since sine and cosine functions can be expressed in terms of complex exponentials, one often uses the more compact notation (which is also valid for complex functions)

$$f(t) = \sum_{n=-\infty}^{\infty} c_n e^{in\omega_f t}, \quad c_n = \frac{1}{T}\int_{-T/2}^{T/2} e^{-in\omega_f t} f(t)dt.$$

Note that $c_0 = a_0$, $c_k = (a_k - ib_k)/2$, $k = 1, 2, \ldots$ and $c_{-k} = \bar{c}_k$ if f is real. For a general complex function $f(t)$, c_k and c_{-k} are complex numbers which are not directly related.

The spectral content, i.e. which frequencies (= harmonics) are contained in the signal is given by its Fourier series.

Aperiodic signals (ranging over the whole real axis $\mathbb{R}$), can be considered as the limit of finite signals (over an interval of length T), but with $T \to \infty$, hence $\omega_f = 2\pi/T \to 0$ but such that $n\omega_f \to \omega$ becomes a continuous variable. So we have to change the summation over $n\omega_f$ into a continuous integral over ω and instead of a discrete spectrum of coefficients c_n, we have a continuous function $F(\omega)$. More precisely we get with appropriate scaling factors

$$f(t) = \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} F(\omega) e^{i\omega t} d\omega$$

and the Fourier transform

$$F(\omega) = \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} f(t) e^{-i\omega t} dt$$

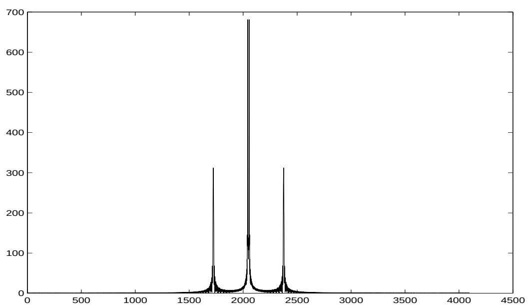
is called the (continuous) *spectrum* of f. Because $T \to \infty$, the resolution $1/T \to 0$, which means that, in principle, we know the signal with infinite precision.

For signals that depend on a discrete time variable, we can give analogous definitions for the Fourier transforms. We give the results in the table below: The upper half corresponds to functions with an infinite time support and the lower half to functions with a finite time support[1].

	digital	analog
signal	$(f_n) \in \ell^2(\mathbb{Z})$	$f(t) \in L^2(\mathbb{R})$
F.T.	$\hat{f}(e^{i\omega}) = \sum_{n\in\mathbb{Z}} f_n e^{-in\omega} \in L^2(\mathbb{T})$	$\hat{f}(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(t) e^{-i\omega t} dt \in L^2(\mathbb{R})$
I.F.T.	$f_n = \frac{1}{2\pi} \int_{-\pi}^{\pi} \hat{f}(e^{i\omega}) e^{in\omega} d\omega$	$f(t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \hat{f}(\omega) e^{i\omega t} d\omega$
signal	$(f_n)_{n=0}^{N-1} \in \mathbb{C}^N$	$f(t) \in L_T^2$
F.T.	$\hat{f}_k = \frac{1}{N} \sum_{n=0}^{N-1} f_n e^{-ink\frac{2\pi}{N}}, \quad k = 0, ..., N-1$	$\hat{f}_k = \frac{1}{T} \int_{-T/2}^{T/2} f(t) e^{-ikt\frac{2\pi}{T}} dt \in \ell^2(\mathbb{Z})$
I.F.T.	$f_n = \sum_{k=0}^{N-1} \hat{f}_k e^{ink\frac{2\pi}{N}}, \quad n = 0, \ldots, N-1$	$f(t) = \sum_{k\in\mathbb{Z}} \hat{f}_k e^{ikt\frac{2\pi}{T}}$

Some of the notation will be explained below. We shall not always treat all the four cases next to each other, but choose the setting that is the simplest to treat. Translation to the other cases is then left as an exercise. The left lower case corresponds to the discrete Fourier transform (DFT) and this is of course the most practical for computations. For example, the modulus of the DFT of the signal given in Figure 1.7 is plotted in Figure 2.1. This figure

Figure 2.1: DFT of the signal in Figure 1.7



shows that the signal can be written as a linear combination of essentially three complex exponentials.

[1]It is common practice to denote the Fourier transform by a hat: $\hat{f} = \mathcal{F}f$. Because we use the hat with another meaning, we do not always use this convention. Sometimes we use a capital to denote the Fourier transform. For example $H(e^{i\omega}) = \sum h_k e^{ik\omega}$. Note that $H(z)$ is a function of $z = e^{i\omega} \in \mathbb{T}$, the unit circle of the complex plane. It is sometimes more convenient to consider it as a function of $\omega \in [-\pi, \pi]$, so that in this case we also use the notation $\mathsf{H}(\omega)$ to mean $\mathsf{H}(\omega) = H(e^{i\omega})$.
There are several variations possible for the definitions in the table. Sometimes they differ in the normalizing factor $1/2\pi$ or $1/\sqrt{2\pi}$ etc., or sometimes the meaning of F.T. and I.F.T. is interchanged. The latter corresponds to writing $1/z$ instead of z where $z = e^{i\omega}$. The first is more common in the engineering literature while the latter is more commonly used by mathematicians.

2.2 The time domain

As we stated above, we consider a signal to be any (complex valued) function of time. In our discussion above, the signal depended on a continuous time variable t. The spectrum was a function depending on a discrete variable (in the periodic case) or a continuous variable (in the non-periodic case).

In practice however, to make digital computations possible, a signal is sampled and in that case, a signal is a time series, i.e., a function of a discrete time variable. Thus, the time can range over a continuum or over a discrete set. In the first case we say that the signal is an *analog* or a *continuous-time* signal, in the second case, it is called a *digital* or *discrete-time* signal.

2.2.1 Digital signals

We shall first consider digital signals because they are somewhat easier to treat. Thus a signal is a complex function of a discrete time variable, which we denote as f_n or occasionally as $f(n)$. In other words, it is an infinite complex sequence.

In practice, signals are often real, but this can be embedded in a complex setting by letting the real part of the complex signal be the real signal.

The *time domain* is the set of all possible signals.

$$\ell = \{f = (f_n) : f_n \in \mathbb{C}, n \in \mathbb{Z}\}$$

We define the *operator* $\mathcal{D}$ by

$$(\mathcal{D}f)_n = f_{n-1}.$$

It causes a *time delay* in the signal. For that reason, engineers call $\mathcal{D}$ a *delay operator*. Because it shifts the signal back one unit in time, it is also called a *(backward) shift* operator.

A *(unit) pulse* $\delta = (\delta_n)$ is a signal which is zero everywhere, except at the moment $n = 0$ where it is 1: $\delta_0 = 1$ while $\delta_n = 0$ for all $n \in \mathbb{Z}_0$. Note that the shifts of the unit pulse $\mathcal{D}^k\delta$ are zero everywhere, except at $n = k$ where it is one. Thus we can decompose a signal with respect to a set of basis functions which are just shifts of the unit pulse:

$$f = \sum_n f_n(\mathcal{D}^n\delta).$$

For mathematical reasons, we often restrict the function space of signals to classical Lebesgue spaces such as

$$\ell^p = \ell^p(\mathbb{Z}) = \left\{ f \in \ell : \|f\|_p = \left(\sum_n |f_n|^p\right)^{1/p} < \infty \right\}, \quad 0 < p < \infty$$

and

$$\ell^\infty = \ell^\infty(\mathbb{Z}) = \left\{ f \in \ell : \|f\|_\infty = \sup_n |f_n| < \infty \right\}.$$

For all $p \geq 1$, ℓ^p is a Banach space, which means that it is a complete space with respect to its norm $\|\cdot\|_p$.

In particular, the case $p = 2$ is interesting because ℓ^2 is a Hilbert space, which means that it is equiped with an inner product:

$$\langle f, g\rangle_{\ell^2(\mathbb{Z})} = \sum_N \overline{f_n} g_n.$$

For $f \in \ell^2$, the norm squared $\|f\|_2^2 = \langle f, f\rangle_{\ell^2(\mathbb{Z})}$ is called the *energy* of the signal.

If $u \in \ell^1$ and $h \in \ell^\infty$, then we can define the *convolution* of the two signals by

$$h * u = u * h = f, \qquad f_n = \sum_m h_m u_{n-m} = \sum_m u_m h_{n-m}.$$

For $f \in \ell^2$, we can define the complex convolution of a signal with shifted versions of itself, giving

$$r_n = \sum_m \overline{f_m} f_{n+m}.$$

This defines a new signal $r = (r_n)$ which is called the *autocorrelation function* of the signal. Note that $r_{-n} = \overline{r_n}$, as follows immediately from the definition. A large value of $|r_n|$ means that signal values which are n samples apart are highly correlated. For example, a signal which is measured every day and which has an approximate periodicity of one week will give a large value for r_7. Note that $r_0 = \|f\|^2$ is the energy of the signal f.

For digital signals, it is interesting to have a notation for the time-reversed signal. If $f = (f_k)$, then we shall define the substar conjugate $g = f_*$ as the signal with values $g_k = \overline{f_{-k}}$, $k \in \mathbb{Z}$. For example, the samples of the convolution signal $f = h * u$ can be denoted as

$$f_n = (h * u)_n = \langle [\mathcal{D}^n h]_*, u\rangle = \sum_m h_{n-m} u_m.$$

2.2.2 Analog signals

We can also consider signals which are depending on a continuous time variable. We suppose it is the infinite real axis. The treatment is completely similar to the discrete time case; we only have to replace discrete sums by continuous integrals. Usually, for digital computations, a continuous signal is sampled, for example with a *sampling period* T. We call $1/T$ the *sampling frequency.* The samples $f(nT)$ can be denoted as f_n and this gives again a discrete time signal.

Again, the *time domain* is the set of all possible signals, i.e. of all complex valued functions defined on the real axis:

$$L = \{f = f(t) : f(t) \in \mathbb{C}, t \in \mathbb{R}\}.$$

We now define the *delay operator* $\mathcal{D}$ by

$$(\mathcal{D}f)(t) = f(t-1).$$

As in the discrete time case, it causes a *time delay* in the signal. For discrete signals, the time delay was only used for integer time intervals, so that only integer powers of $\mathcal{D}$ had to be considered. Here, for continuous signals, we shall consider delays over an arbitrary real time interval h, which is indicated by real powers of $\mathcal{D}$, namely $(\mathcal{D}^h f)(t) = f(t-h)$.

An *impulse* is a Dirac delta function wich is a generalized function δ satisfying

$$\int_{\mathbb{R}} f(t)\delta(t-t_0)dt = f(t_0), \qquad f \in L.$$

Most often the function spaces of signals are restricted to classical Lebesgue spaces such as

$$L^p = L^p(\mathbb{R}) = \left\{ f \in L : \|f\|_p = \left(\frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} |f(t)|^p dt \right)^{1/p} < \infty \right\}, \quad p \geq 1$$

and

$$L^\infty = L^\infty(\mathbb{R}) = \left\{ f \in L : \|f\|_\infty = \sup_t |f(t)| < \infty \right\}.$$

For all $p \geq 1$, L^p is a Banach space, which means that it is a complete space with respect to its norm.

L^2 is a Hilbert space equiped with an inner product[2]:

$$\langle f, g \rangle_{L^2(\mathbb{R})} = \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} \overline{f(t)} g(t) dt.$$

When $f \in L^2$ we say that $\|f\|^2 = \langle f, f \rangle_{L^2(\mathbb{R})}$ is the *energy* of the signal.

The convolution of the signal u and the signal h is defined as[3]

$$(h * u)(t) = \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} h(\tau) u(t-\tau) d\tau.$$

The *autocorrelation function* of a signal f is defined by

$$r(t) = \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} \overline{f(\tau)} f(t+\tau) d\tau.$$

It holds that $r(-t) = \overline{r(t)}$, $t \in \mathbb{R}$.

For periodic signals, we take as standard interval $[-\pi, \pi]$, and the time domain is the space $L^2_{2\pi}$ of 2π-periodic signals. The treatment is as above, except that the inner product is given by

$$\langle f, g \rangle_{L^2_{2\pi}} = \frac{1}{2\pi} \int_{-\pi}^{\pi} \overline{f(t)} g(t) dt.$$

[2]Note that we use the normalization factor $1/\sqrt{2\pi}$.

[3]Again note the factor $1/\sqrt{2\pi}$ in this definition.

2.3 The frequency domain

2.3.1 Digital signals

Working in the time domain is not always efficient, so it is common practice to associate with a sequence some formal series, which is called its *z-transform* and which is defined by

$$\mathcal{Z} : f = (f_n) \mapsto F(z) = \sum_n f_n z^{-n}.$$

A shift (delay) in the time domain is translated into a multiplication with z^{-1} in the z-domain: If $g = \mathcal{D}f$, then $G(z) = z^{-1}F(z)$.

Note that in general, the z-transform series need not converge. The series being only a formal series, means that z^n has to be interpreted as a "place holder". However, if $f \in \ell^p$, the z-transform F does converge for $z = e^{i\omega} \in \mathbb{T}$ where $\mathbb{T}$ represents the complex unit circle. In fact it defines a function which is in the space

$$L^p = L^p(\mathbb{T}) = \left\{ F : \|F\|_p = \left(\frac{1}{2\pi} \int_{-\pi}^{\pi} |F(e^{i\omega})|^p d\omega \right)^{1/p} < \infty \right\}.$$

This function $F(z)$ for $z = e^{i\omega} \in \mathbb{T}$, is the *Fourier transform* of the signal f. The *inverse Fourier transform* describes the signal f in terms of F.

$$f_n = \frac{1}{2\pi} \int_{-\pi}^{\pi} e^{in\omega} F(e^{i\omega}) d\omega = \left\langle z^{-n}, F(z) \right\rangle_{L^2(\mathbb{T})}.$$

The f_n are the *Fourier coefficients* of the function F. Note that the z transform of a shifted impulse is $\mathcal{D}^m\delta$ is z^{-m}. Thus the decomposition of the signal with respect to the basis $\{\mathcal{D}^n\delta\}$ in the time domain corresponds to the decomposition in the z-domain of the series F with respect to the basis $\{z^{-n}\}$. Setting $z = e^{i\omega}$, the signal is transformed into the ω-domain, which is called the *frequency domain* and ω is called a *frequency.* Because $F(e^{i\omega})$ is 2π-peridodic in ω, we consider the frequency domain to be the interval $[-\pi, \pi]$. Note the reciprocity: a periodic signal has a discrete spectrum, while a discrete signal has a periodic spectrum (Fourier transform). Since the Fourier transform $F(e^{i\omega}) = \mathcal{F}(f_n) \in L^2(\mathbb{T})$ of $f = (f_n) \in \ell^2(\mathbb{Z})$, can also be considered as a 2π-periodic function of ω, we shall sometimes denote $F(e^{i\omega})$ as $\mathsf{F}(\omega) \in L^2_{2\pi}$. Of course $L^2(\mathbb{T})$ and $L^2_{2\pi}$ are isomorphic.

For signals in $\ell^2(\mathbb{Z})$, the Fourier transform is in $L^2(\mathbb{T})$ [or $L^2_{2\pi}$] and the Fourier transform is an isometric isomorphism between these spaces, i.e. $\langle f, g \rangle_{\ell^2(\mathbb{Z})} = \langle F, G \rangle_{L^2(\mathbb{T})} = \langle \mathsf{F}, \mathsf{G} \rangle_{L^2_{2\pi}}$ where $F(e^{i\omega}) = \mathsf{F}(\omega)$ and $G(e^{i\omega}) = \mathsf{G}(\omega)$ are the Fourier transforms of f and g respectively. The signal f and its Fourier transform F are called a *Fourier transform pair.*

A convolution of two signals in the time domain translates into an ordinary multiplication of their z-transforms in the z-domain. If $h \in \ell^1$ and $u \in \ell^\infty$, then the z-transform of $h * u$ is given by $H(z)U(z)$. This is the main reason why it is easier to work in the z-domain than to work in the time domain.

The Fourier transform of the autocorrelation function $r = (r_n)$ of a signal f is given by $R(z) = \sum_n r_n z^{-n}$ for $z = e^{i\omega}$. This Fourier transform is called the *power spectrum* of the signal f. Obviously

$$R(z) = F(z)F_*(z), \quad F_*(z) = \overline{F(1/\overline{z})}.$$

The transform $F \mapsto F_*$ is called *para-hermitian conjugation*. It is easily checked that if $F = \mathcal{F}(f)$, then $F_* = \mathcal{F}(f_*)$. Thus the para-hermitian conjugate in the frequency domain corresponds to a complex conjugate time reversal in the time domain. The para-hermitian conjugate extends the definition of complex conjugate on the circle $\mathbb{T}$ to the whole complex plane $\mathbb{C}$. Indeed, it is also easy to check that for $z \in \mathbb{T}$ we have $F_*(z) = \overline{F(z)}$. This observation implies that on $\mathbb{T}$, the power spectrum equals

$$R(e^{i\omega}) = |F(e^{i\omega})|^2$$

which is obviously a nonnegative function. Therefore, it can be used as a weight function to define a weighted L^2 space with inner product

$$\langle f, g \rangle_R = \frac{1}{2\pi} \int_{-\pi}^{\pi} \overline{f(e^{i\omega})} g(e^{i\omega}) R(e^{i\omega}) d\omega.$$

The measure $d\mu(\omega) = R(e^{i\omega})d\omega$ is called the *spectral measure* of the signal. Note that we can now write in this weighted space

$$\langle z^k, z^l \rangle_R = \frac{1}{2\pi} \int_{-\pi}^{\pi} e^{i(l-k)\omega} R(e^{i\omega}) d\omega = r_{l-k}.$$

Thus, the Gram matrix of this Hilbert space is given by

$$\left[\langle z^k, z^l \rangle_R\right]_{k,l\in\mathbb{Z}} = [r_{l-k}]_{k,l\in\mathbb{Z}}$$

which is a Hermitian positive definite Toeplitz matrix. A Toeplitz matrix M is a matrix whose entry $m_{k,l}$ at row k and column l depends only on the difference $k-l$. It has elements which do not change along the main diagonal and its parallels.

The infinite Toeplitz matrix $[r_{k-l}]$ defines a Toeplitz operator

$$f = (f_n) \mapsto \sum_l r_{k-l} f_l.$$

Note that also the convolution of two signals can be written as a multiplication with a Toeplitz matrix. Indeed, if $h = f * g$, then

$$h_n = \sum_k f_{n-k} g_k = \sum_k g_{n-k} f_k, \quad n \in \mathbb{Z}.$$

Thus, if we denote in bold face the infinite (column) vector containing the samples of a signal, then we can write a convolution as

$$h = f * g \quad \Leftrightarrow \quad \mathbf{h} = \mathbf{T}_f \mathbf{g}, \quad \mathbf{T}_f = [\cdots|\mathbf{Z}^{-1}\mathbf{f}|\mathbf{f}|\mathbf{Z}\mathbf{f}|\cdots]$$

where $\mathbf{Z}$ is the matrix which represents the (down-)shift operator, i.e. with ones on the first subdiagonal and zeros everywhere else.

$$\begin{bmatrix} \vdots \\ h_{-1} \\ \boxed{h_0} \\ h_1 \\ \vdots \end{bmatrix} = \begin{bmatrix} & \ddots & \ddots & \ddots & & & \\ \cdots & f_2 & f_1 & \boxed{f_0} & f_{-1} & \cdots & \\ & \cdots & f_2 & f_1 & f_0 & f_{-1} & \cdots \\ & & & \ddots & \ddots & \ddots & \end{bmatrix} \begin{bmatrix} \vdots \\ g_{-1} \\ \boxed{g_0} \\ g_1 \\ \vdots \end{bmatrix} \tag{2.1}$$

A signal is called *band-limited* if its Fourier transform is only different from zero in a part of the spectrum $[-\pi, \pi]$. Since the Fourier transform writes the signal as a linear combination of the basis functions $e^{ik\omega}$, and because these functions are highly oscillating for large ω and only slowly varying for small ω (the complex exponential represents cosines and sines), it is clear that if a signal has a Fourier transform which is only nonzero for "small" ω, then it will be a smooth function which is only slowly varying while a signal whose Fourier transform lives in a high frequency band will have high frequencies and it will thus contain much detail information.

2.3.2 Analog signals

The same reasoning can be followed for non-periodic analog signals, again replacing sums by integrals. For example, the Fourier transform for $f \in L^2(\mathbb{R})$ is defined by

$$F(\omega) = \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} e^{-i\omega t} f(t) dt = \left\langle e^{i\omega t}, f(t) \right\rangle_{L^2(\mathbb{R})},$$

while the inverse Fourier transform is given by the expansion

$$f(t) = \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} e^{i\omega t} F(\omega) d\omega = \left\langle e^{-i\omega t}, F(\omega) \right\rangle_{L^2(\mathbb{R})}.$$

For square integrable signals, the Plancherel formula says that

$$\|f\|^2 = \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} |f(t)|^2 dt = \|F\|^2 = \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} |F(\omega)|^2 d\omega$$

while the Parseval equality gives[4]

$$\langle f, g \rangle_{L^2(\mathbb{R})} = \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} \overline{f(t)} g(t) dt = \langle F, G \rangle_{L^2(\mathbb{R})} = \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} \overline{F(\omega)} G(\omega) d\omega.$$

The Fourier transform of a convolution $h * u$ is the product of the Fourier transforms[5]

$$\mathcal{F}(h * u) = \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} e^{-i\omega t} (h * u)(t) dt = H(\omega) U(\omega)$$

[4]The Plancherel and Parseval equalities still hold without the factor $1/\sqrt{2\pi}$ in the definition of the inner product of $L^2(\mathbb{R})$.

[5]Here the factor $1/\sqrt{2\pi}$ in the definition of the convolution is essential. Without this factor, the Fourier transform of the convolution is the product of the Fourier transforms times $\sqrt{2\pi}$.

where U and H are the Fourier transforms of u and h respectively.

The analog of the para-hermitian conjugare is now associated with a reflection in the real axis: $F_*(z) = \overline{F(\overline{z})}$. The interpretation as a time reversal is not immediate, but we do have that this extends the definition of complex conjugate on $\mathbb{R}$ to the whole complex plane $\mathbb{C}$. We have for example as before that the power spectrum R, which is the Fourier transform of the autocorrelation function $r(t)$ is given by

$$R(z) = F(z)F_*(z) \quad \text{which equals } |F(z)|^2 \geq 0 \text{ for } z \in \mathbb{R}.$$

Thus we can again define a weighted L^2 space with spectral weight $d\mu(\omega) = |F(\omega)|^2 d\omega$. The kernel function

$$k(t,\tau) = \left\langle e^{it\omega}, e^{i\tau\omega} \right\rangle_{L^2(\mathbb{R})} = \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} e^{i(\tau - t)\omega} R(\omega) d\omega = r(\tau - t)$$

depends only on the difference of its arguments and is called a Toeplitz kernel. It is the kernel of a Toeplitz integral operator defined by

$$f(t) \mapsto \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} r(t-\tau) f(\tau) d\tau$$

For periodic signals $f \in L^2_{2\pi}$, the Fourier transform is discrete and belongs to the space $\ell^2(\mathbb{Z})$

$$f_k = \frac{1}{2\pi} \int_{-\pi}^{\pi} e^{-ikt} f(t) dt = \left\langle e^{ikt}, f \right\rangle_{L^2_{2\pi}}.$$

The inverse transform gives the expansion (Fourier series)

$$f(t) = \sum_{k \in \mathbb{Z}} f_k e^{ikt}.$$

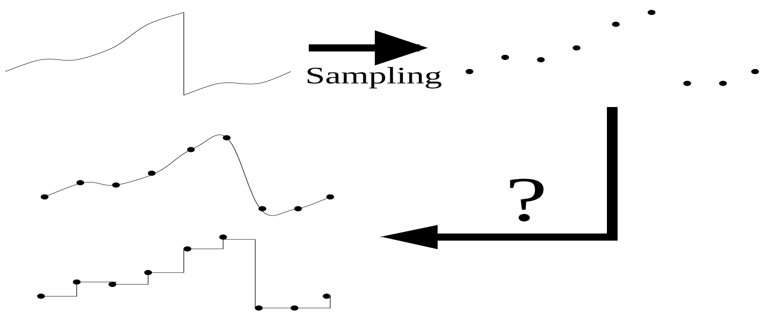
This is the mirror situation of a discrete signal with a periodic spectrum.

2.4 Sampling theorem

If a continuous signal $f(t)$ is sampled to give the discrete signal $f(nT)$, then in general, it is impossible to recover the original signal from these samples because we do not know how it behaves in between two samples. See for example Figure 2.2.

However, if we know that the signal is band-limited, then it is possible to recover the signal with a finite sampling rate. For example, if the signal is $\cos(2N\pi t)$ and if we sample this with a period $T = 1/N$, we get $f_n = \cos(2n\pi) = 1$ for all n. If we do not know more about the signal, we can not recover it because for example the constant function 1, will give exactly the same samples. Thus sampling at the highest frequency represented in the signal is not good enough (and a fortiori sampling at a slower rate). One has to sample at a rate that is at least twice the highest frequency. In our example, this will give the samples $f_{2n} = 1$ and $f_{2n+1} = -1$. Given the fact that the highest possible frequency is $2N$ cycles, such a sample sequence can only be generated by the original signal $\cos(2N\pi t)$. A fortiori, if

Figure 2.2: Sampling and reconstruction



the signal is sampled at a higher frequency, it will be reconstructable from its samples. The critical sampling frequency of twice the maximal frequency is called the *Nyquist frequency*:

$$\omega_s = 2\omega_m$$

where ω_s is the Nyquist sampling frequency and ω_m is the maximal frequency of the signal.

The sampling of a continuous signal can be caught in a formula as follows. Consider the function δ to be an analog signal (defined for all $t \in \mathbb{R}$) which represents the digital unit pulse:

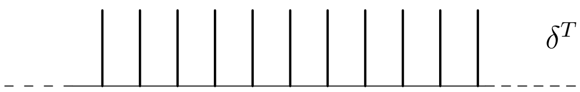
$$\delta(t) = \begin{cases} 1, & t = 0 \\ 0, & t \neq 0 \end{cases}$$

Obviously, the signal δ^T defined by

$$\delta^T(t) = \sum_{n \in \mathbb{Z}} \delta(t - nT)$$

will be an analog signal which is zero everywhere, except at the points $t = nT$, $n \in \mathbb{Z}$ where it is 1. This is called a (unit) *pulse train*.

Figure 2.3: Pulse train



Then we can consider for a given function f defined on $\mathbb{R}$ (an analog signal), the function

$$f_s(t) = f(t)\delta^T(t).$$

which is still an analog signal, zero everywhere, except at $t = nT$ where it takes the value of the nth sample $f_n = f(nT)$.

Since δ^T is a periodic function (period T), with Fourier coefficients $c_n = 1/T$, we can write the Fourier series expansion $\delta^T(t) = \frac{1}{T}\sum_n e^{in\omega_s t}$, where $\omega_s = 2\pi/T$ is the sampling frequency. Thus we have

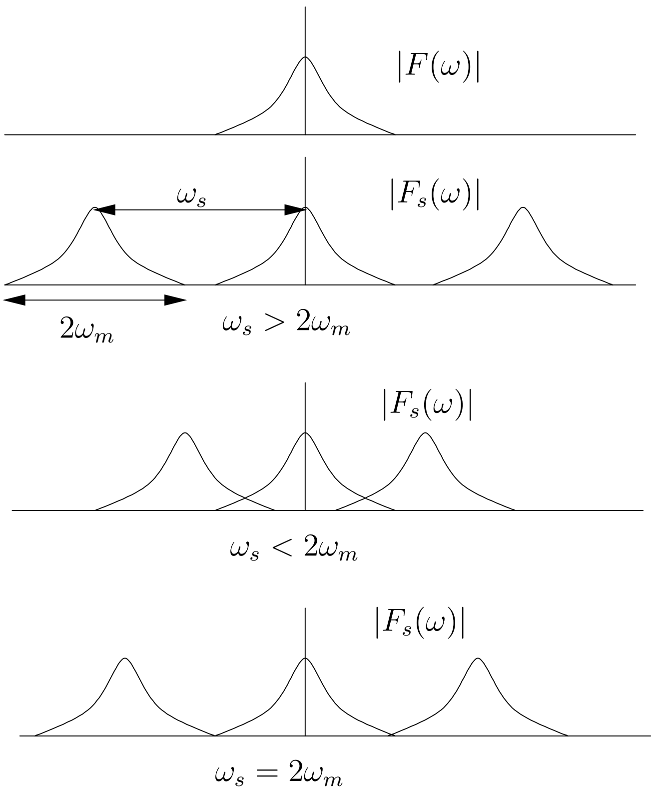
$$f_s(t) = \delta^T(t)f(t) = \frac{1}{T}\sum_n f(t)e^{in\omega_s t}$$

so that its Fourier transform is

$$F_s(\omega) = \frac{1}{T}\sum_n F(\omega - n\omega_s).$$

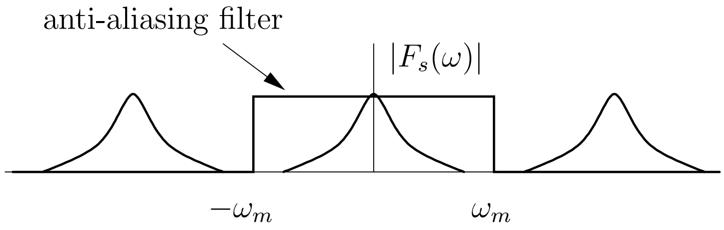
This shows that the spectrum F_s is a periodic repetition of the spectrum F, multiplied by $1/T$ (see Figure 2.4). The periodicity is defined by ω_s. Recall that the spectrum F lives in the interval $[-\omega_m, \omega_m]$ since its highest frequency is ω_m. Thus if ω_s is too small, the repeated spectra will overlap. This is called *aliasing*. It will not be possible to isolate the spectrum F of the continuous signal from the specrum F_s of the sampled signal. Thus f can not be recovered from the spectrum F_s, or equivalently from the sampled signal. If

Figure 2.4: Sampling theorem



however ω_s is large enough, i.e. if $\omega_s > 2\omega_m$, then the repetitions of the spectra F do not overlap and thus it is possible to isolate F from F_s and thus it is possible to recover the signal f from the spectrum F_s, i.e. from the samples f_n. Thus, if the sampling frequency is high enough, we have to isolate F first from F_s. This is obtained by filtering out only the frequencies $|\omega| < \omega_s/2$. This is done by a low pass filter, also called anti-aliasing filter (see next chapter). Mathematically, this means that (in the ideal case) we multiply F_s with a function $H(\omega)$ that is equal to T in $|\omega| < \omega_s/2$ and that is zero outside this interval. This

Figure 2.5: Anti-aliasing filter



selects the frequencies $|\omega| < \omega_m$ and kills all the frequencies that are larger. Thus F being isolated from F_s, we can reconstruct f. The IFT of this filter $H(\omega)$ is $(\sin \frac{\omega_s t}{2})/(\frac{\omega_s t}{2})$ (see next chapter or check it as an exercise). Since the multiplication in the frequency domain corresponds to a convolution in the time domain, it should now be plausible that we have the following sampling theorem. For the mathematics of its proof we refer to the literature.

Theorem 2.4.1 (Sampling theorem). *Suppose the continuous signal f is bandlimited, say low pass, which means that $F(\omega)$ is essentially nonzero in $|\omega| < \omega_m$ and it is zero everywhere else: the spectrum $F(\omega)$ lives in the band $|\omega| < \omega_m$.*

If the continuous signal is sampled at a rate

$$\frac{1}{T} = \frac{\omega_s}{2\pi}, \quad \text{with} \quad \omega_s > 2\omega_m,$$

then it is possible to recover the signal from its samples $f_n = f(nT)$, $n \in \mathbb{Z}$ by the formula

$$f(t) = \sum_{n\in\mathbb{Z}} f(nT) \frac{\sin \omega_m(t - nT)}{\omega_m(t - nT)}$$

Remark: Since an ideal low pass filter which changes from a nonzero value for $|\omega| < \omega_s/2$ immediately to the value 0 for $|\omega| > \omega_s/2$ can never be realized, one usually samples at a slightly higher rate than the Nyquist frequency to avoid aliasing because the filtering will not be ideal.

2.5 Subsampling and upsampling of a discrete signal

What has been said before about sampling of a continuous signal can be repeated with appropriate modifications when we want to (sub)sample a discrete signal. This means that for a given discrete signal $f = (f_n)$, we produce a subsampled signal $g = (g_n)$ where $g_n = f_{nM}$,

$n \in \mathbb{Z}$. Thus, we take every Mth sample of the original signal. The analysis parallels the previous one.

First define the discrete unit pulse: $\delta = (\delta_n)$, and the discrete pulse train $\delta^M = \sum_k \mathcal{D}^{kM}\delta$ which is a discrete signal where δ_n^M is equal to 1 for $n = kM$, $k \in \mathbb{Z}$ and is zero for all other indices. Define the subsampled signal f' by

$$f'_n = \begin{cases} f_n, & n = 0, \pm M, \pm 2M, \dots \\ 0, & \text{otherwise} \end{cases}$$

Then

$$f'_n = \delta_n^M f_n.$$

Finally define $g_n = f'_{nM} = f_{nM}$, $n \in \mathbb{Z}$.

f	$\cdots$	$f_0,$	$f_1, \dots$	$f_M,$	$f_{M+1}, \dots$	$f_{2M},$	$f_{2M+1}, \dots$	$\dots$
δ^M	$\cdots$	$1,$	$0, \dots, 0$	$1,$	$0, \dots, 0$	$1,$	$0, \dots, 0$	$\dots$
f'	$\cdots$	$f_0,$	$0, \dots 0$	$f_M,$	$0, \dots, 0$	$f_{2M},$	$0, \dots, 0$	$\dots$
g	$\cdots$	f_0	$= g_0$	f_M	$= g_1$	f_{2M}	$= g_2$	$\dots$

Since δ^M has period M, its Fourier transform is $(\mathcal{F}\delta^M)_k = 1/M$, $k = 0, \dots, M-1$, and it follows that δ^M can be expanded in its Fourier series

$$\delta_n^M = \frac{1}{M}\sum_{k=0}^{M-1} e^{i\frac{2\pi}{M}nk}.$$

(Check that indeed the right hand side is 1 for $n \in M\mathbb{Z}$ and 0 otherwise.) Thus

$$f'_n = \frac{1}{M}\sum_{k=0}^{M-1} f_n e^{i\frac{2\pi}{M}nk}$$

with Fourier transform

$$F'(e^{i\omega}) = \frac{1}{M}\sum_{k=0}^{M-1} F(e^{i(\omega - \frac{2\pi}{M}k)}).$$

Again we see that $F'(e^{i\omega})$ is the sum of M replicas of $F(e^{i\omega})$. These replicas are rotated on the unit circle and spaced $2\pi/M$ apart.

To go from f' to g, we have to compress the time axis by a factor M, which corresponds in the z-domain by repacing z by $z^{1/M}$. Indeed

$$G(z) = \sum_n f'_{nM} z^{-n} = \sum_k f'_k z^{-k/M} = \sum_k f'_k (z^{1/M})^k = F'(z^{1/M}).$$

Thus with $z = e^{i\omega}$, ω should be divided by M, so that

$$G(e^{i\omega}) = F'(e^{i\frac{\omega}{M}}) = \frac{1}{M}\sum_{k=0}^{M-1} F(e^{i(\frac{\omega - 2\pi k}{M})}).$$

Thus compressing the time domain by a factor M corresponds to stretching the ω-domain by a factor M. The subsampled signal g has a spectrum G that is the sum of M shifted replicas of the spectrum F which is stretched by a factor M and these replicas are spaced 2π apart. Thus to avoid overlap (aliasing), the bandwidth of F should not be larger than π/M, otherwise the replicas overlap, and the original signal can not be recovered from the subsampled one.

On the other hand, if a signal is filtered such that only a subband of bandwidth π/M remains, then this subband signal need not be stored at the original sampling rate because by storing only every Mth sample, we will still be able to reconstruct the subband signal.

Thus if we split a signal into two halfband signals, then if the original signal is stored with 10 000 samples, we should only keep 5 000 samples for each subband, giving the same amount of numbers containing the same amount of information.

The inverse operation of subsampling is *upsampling*. Now the given signal f is streched and the samples in between are interpolated with zeros. $f \to g$ i.e. $g_n = f_{n/M}$ for $n \in M\mathbb{Z}$ and $g_n = 0$ otherwise.

In the z-domain, this corresponds to

$$G(z) = F(z^M).$$

It is usual to denote a downsampling by M as $(\downarrow M)$ and upsampling by M as $(\uparrow M)$. Note that $(\uparrow M)(\downarrow M)f \neq f$, but we do have $(\downarrow M)(\uparrow M)f = f$. The operations $(\downarrow M)$ and $(\uparrow M)$ are conjugates in the sense that $\langle(\downarrow M)f, g\rangle = \langle f, (\uparrow M)g\rangle$. The matrix representation (w.r.t. the standard basis) for downsampling is the unit matrix in which we keep only every Mth row, the matrix representation for upsampling is the unit matrix in which we keep only every Mth column. They are each others transpose.

2.6 The Heisenberg uncertainty principle

As we said in the introduction, it is the purpose of a wavelet basis that it should be locallized in the time domain as well as in the frequency domain. Thus ideally, a wavelet basis function should have a compact support in the time domain while its Fourier transform should have a compact support in the frequency domain. However, the famous Heisenberg uncertainty principle says that it is impossible to have a signal with finite support on the time axis which is at the same time band limited.

Consider the basis function $e^{i\omega_0 t}$. This has a perfect localization in the frequency domain. It contains exactly one frequency ω_0 because its Fourier transform is $\sqrt{2\pi}\delta(\omega - \omega_0)$. It is zero everywhere, except for $\omega = \omega_0$. In the time domain however, the complex exponential represents a cosine and a sine which are essentially nonzero on the whole real axis.

On the other extreme, we can consider a delta function in the time domain: $\delta(t - t_0)$. This is perfectly localized in the time domain. But its Fourier transform is $e^{-i\omega t_0}/\sqrt{2\pi}$ which is essentially nonzero on the whole real ω-axis. This is not band-limited.

These are the two extremes: either we have perfect localization in the time domain, but then the signal will contain all the frequencies, or we have perfect localization in the frequency domain, but then, the signal will live on the whole real time-axis.

In general, let us define some measure for expressing the width of the support of a function and of its Fourier transform. Let f be a signal and F its Fourier transform. Define the weighted means in the time domain and in the frequency domains as ($|f|^2/\|f\|_2$ and $|F|^2/\|F\|_2$, can be considered as probability density functions)

$$t_0 = \frac{\int t|f(t)|^2 dt}{\int |f(t)|^2 dt} \quad \text{and} \quad \omega_0 = \frac{\int \omega |F(\omega)|^2 d\omega}{\int |F(\omega)|^2 d\omega}.$$

We say that the signal f is concentrated at (t_0, ω_0) in the time-frequency domain. (We assume that t_0 and ω_0 are finite.) The standard deviations s and S are then a measure for the width of the distribution of the time-frequency distribution of f around (t_0, ω_0). These s and S are given as the square roots of the variances

$$s^2 = \frac{\int (t - t_0)^2 |f(t)|^2 dt}{\int |f(t)|^2 dt} \quad \text{and} \quad S^2 = \frac{\int (\omega - \omega_0)^2 |F(\omega)|^2 d\omega}{\int |F(\omega)|^2 d\omega}.$$

These define somehow the measure we need (think of the Gauss function $f(t) = \exp(-\frac{(t-t_0)^2}{2\sigma^2})$ whose Fourier transform is another such Gaussian and the above width measure is related to the variance σ). Then the following theorem holds:

Theorem 2.6.1 (Heisenberg uncertainty principle). *Let F be the Fourier transform of f and let s^2 and S^2 be defined as above, then*

$$s^2 S^2 \geq \frac{1}{4}.$$

Proof. The proof simplifies considerably when we assume a coordinate transformation such that $t_0 = 0$ and $\omega_0 = 0$. If s^2 or S^2 are infinite, nothing has to be proved. So assume that $\|tf(t)\|$ and $\|\omega F(\omega)\|$ are finite, which implies that for example $\lim_{t\to\pm\infty} |tf(t)|^2 = 0$ and thus certainly $t|f(t)|^2 \to 0$ for $t \to \pm\infty$. Also $f' \in L_2(\mathbb{R})$. In that case, the proof goes as follows. Let $\mathcal{P}$ be the operator $\mathcal{P}f(t) = tf(t)$ and $\mathcal{Q}$ the operator $\mathcal{Q}f(t) = f'(t)$, then $\mathcal{QP} - \mathcal{PQ} = \mathcal{I}$ because

$$(\mathcal{QP} - \mathcal{PQ})f(t) = \frac{d}{dt}(tf(t)) - t\frac{df(t)}{dt} = f(t).$$

Therefore, we have

$$\|f\|^2 = \langle f, (\mathcal{QP} - \mathcal{PQ})f \rangle = \langle f, \mathcal{QP}f \rangle - \langle f, \mathcal{PQ}f \rangle .$$

Now by partial integration:

$$\langle f, \mathcal{QP}f \rangle = \int_{\mathbb{R}} \overline{f(t)} \frac{d}{dt}[tf(t)]dt = t|f(t)|^2 \Big|_{-\infty}^{\infty} - \int_{\mathbb{R}} t\overline{f'(t)} f(t) dt = -\langle \mathcal{Q}f, \mathcal{P}f \rangle$$

Since also $\langle f, \mathcal{PQ}f \rangle = \langle \mathcal{P}f, \mathcal{Q}f \rangle$, we get

$$\|f\|^2 = -2\mathrm{Re}\, \langle \mathcal{Q}f, \mathcal{P}f \rangle .$$

The Cauchy-Schwartz inequality gives

$$|\mathrm{Re}\,\langle f, \mathcal{PQ}f\rangle\,| \leq \|\mathcal{P}f\|\ \|\mathcal{Q}f\|$$

and this leads to

$$\frac{\|\mathcal{Q}f\|}{\|f\|}\frac{\|\mathcal{P}f\|}{\|f\|} \geq \frac{1}{2}.$$

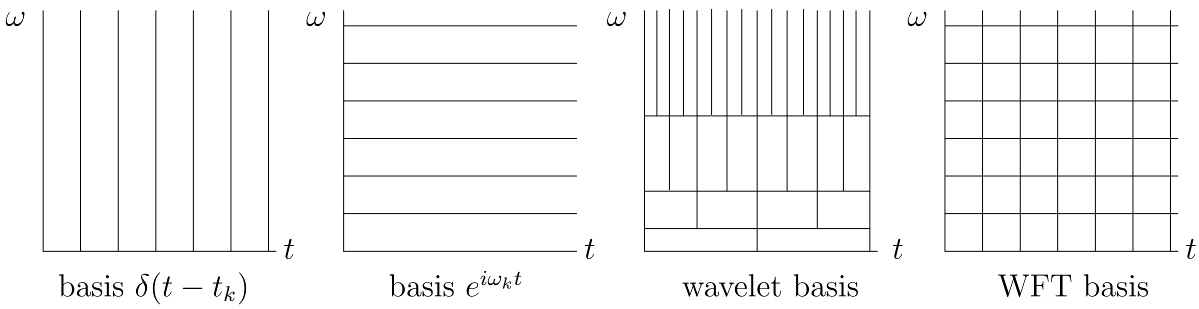
Obviously $s = \|\mathcal{P}f\|/\|f\|$ and by the isomorphism between the time domain and the Fourier domain, recalling that differentiation in the time domain corresponds to multiplication with $i\omega$ in the Fourier domain, it should be clear that $S = \|\mathcal{Q}f\|/\|f\|$, and the theorem is proved. □

The Gaussian function is the only function for which equality holds in the Heisenberg uncertainty principle. Indeed, one has equality when $\langle \mathcal{P}f, \mathcal{Q}f\rangle$ is real and negative, thus when $\mathcal{Q}f = -c\mathcal{P}f$ with c some positive constant. The only solution of this differential equation is a multiple of $e^{-ct^2/2}$ with $c > 0$.

2.7 Time-frequency plane

In the time-frequency plane, it follows from the Heisenberg uncertainty principle that, setting out the widths of a basis function and its Fourier transform, then wavelet basis functions will essentially contribute to the signal in a rectangle. To catch high frequencies, we need a small s, for low frequencies, we can use basis functions with a large s. The basis functions

Figure 2.6: Time-frequency plane



$\delta(t - t_k)$ represent vertical lines (infinitely thin and infinitely high rectangles). The basis functions $e^{i\omega_k t}$ represent horizontal lines (infinitely small and infinitely long rectangles). The wavelet basis will correspond to finite rectangles: narrow and high (thin rectangles) for high frequencies and wide and small (fat rectangles) for low frequencies.

A first attempt to reach such an objective is given by the *windowed Fourier transform* (WFT) or the *short time Fourier transform* (STFT). Here the basis functions $e^{i\omega t}$ are replaced by windowed versions $\psi_{\omega,b}(t) = g(t-b)e^{i\omega t}$, where g is a window (for example $g(t) = \exp(-t^2)$, in which case the corresponding transform is called the *Gabor transform*). Thus the WFT is a function of two variables:

$$\mathcal{F}_g f = F(\omega, b) = \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} \overline{\psi_{\omega,b}(t)} f(t) dt = \langle \psi_{\omega,b}, f \rangle_{L^2(\mathbb{R})} .$$

The inverse transform is derived from the inverse Fourier transform which gives

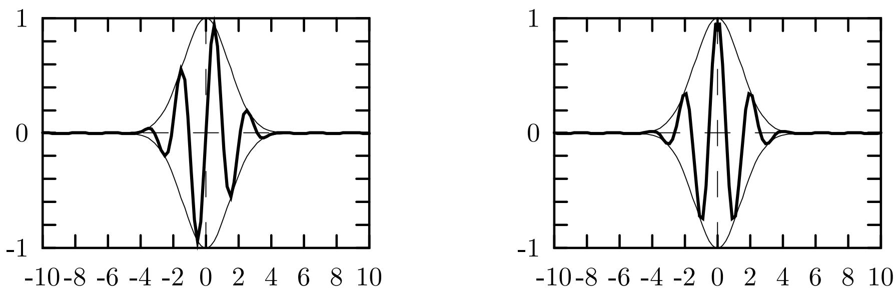
$$\overline{g(t-b)} f(t) = \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} F(\omega, b) e^{i\omega t} d\omega.$$

Multiplying with $g(t-b)$ and integrating w.r.t. b over $\mathbb{R}$ gives the inverse WFT:

$$f(t) = \frac{1}{2\pi \|g\|^2} \int\!\!\int_{\mathbb{R}^2} F(\omega, b) g(t-b) e^{i\omega t} d\omega db$$

Since in the WFT, the width of all the basis functions is given by the width of the window,

Figure 2.7: Gabor transform basis functions: $\sin(t)e^{-t^2}$, $\cos(t)e^{-t^2}$



which is constant, it is obvious that this basis will give equal rectangles for all time and all frequencies, whereas the wavelet basis gave also rectangles of equal area but where the scale changed logarithmically with time and frequency which is much more comform to the physiology of human observation (e.g. hearing for audio signals or vision for images). This corresponds to using a basis $\psi_{a,b}(t) = \sqrt{|a|}\psi(a(t-b))$. The continuous wavelet transform (CWT) is $(a, b \in \mathbb{R})$

$$F(a, b) = \frac{1}{\sqrt{C_\psi}} \int_{\mathbb{R}} \overline{\psi_{a,b}(t)} f(t) dt$$

while the inverse wavelet transform is given by:

$$f(t) = \frac{1}{\sqrt{C_\psi}} \int\!\!\int_{\mathbb{R}^2} F(a, b) \psi_{a,b}(t) da\, db,$$

where it is supposed that the constant

$$C_\psi = 2\pi \int_{\mathbb{R}} |\Psi(\omega)|^2 \frac{d\omega}{|\omega|},$$

is finite ($\Psi(\omega)$ is the Fourier transform of $\psi(t)$). Note that b refers to 'time' and a refers to 'scale', so that the continuous wavelet transform is defined in (a part of) the time-scale space; it is a time-scale representation of the signal.

For example the *Morlet wavelet* chooses a modulated Gaussian for $\psi(t)$:

$$\psi(t) = e^{-i\alpha t} e^{-t^2/2}, \quad \alpha = \pi\sqrt{\frac{2}{\ln 2}} \approx 5.336.$$

It looks much like the figures given for the Gabor transform. The difference is that in the Gabor transform, the support is constant and is only shifted by the parameter b. The different frequencies were obtained by changing ω explicitly. This gives the same exponential hull with different sine/cosines inside. Here, with Morlet wavelets, the parameter b has the same function: shifting in the time domain, but the parameter a will now stretch and compress the exponential hull in the figure above.

Both the WFT and the CWT are highly redundant in practical situations and therefore it is sufficient to sample the a and b and use only discrete values for it, for example $a = 2^n$ and $b = 2^{-n}k$, giving $\psi_{nk}(t) = 2^{n/2}\psi(2^n t - k)$. The wavelet transform is then a double sequence $w = (w_{n,k})_{n,k\in\mathbb{Z}^2} \in \ell^2(\mathbb{Z}^2)$ and the inverse transform is a double series: $f(t) = \sum_{n,k} w_{nk}\psi_{nk}(t)$. This discrete version of the wavelet transform is the kind of transform we shall study in greater detail.

2.8 Summary

We have several Hilbert spaces with several inner products:

$L^2(\mathbb{R})$	$\langle f, g\rangle = \frac{1}{\sqrt{2\pi}}\int_{\mathbb{R}} \overline{f(t)}g(t)dt$
$\ell^2(\mathbb{Z})$	$\langle f, g\rangle = \sum_{k\in\mathbb{Z}} \overline{f}_k g_k$
$\mathbb{C}^N$	$\langle f, g\rangle = \sum_{k=0}^{N-1} \overline{f}_k g_k$
L^2_T	$\langle f, g\rangle = \frac{1}{T}\int_{-T/2}^{T/2} \overline{f(t)}g(t)dt$
$L^2(\mathbb{T})$	$\langle f, g\rangle = \frac{1}{2\pi}\int_{\mathbb{T}} \overline{f(z)}g(z)d\theta, \quad z = e^{i\theta}$
$\equiv$	
$L^2_{2\pi}$	$\langle f, g\rangle = \frac{1}{2\pi}\int_{-\pi}^{\pi} \overline{f(t)}g(t)dt$

We consider two kinds of convolutions:

In $L^2(\mathbb{R})$: $g = h * u$	$\Leftrightarrow$ $g(t) = \dfrac{1}{\sqrt{2\pi}} \displaystyle\int_{\mathbb{R}} h(\tau)u(t-\tau)d\tau$	$\Leftrightarrow$	$G(\omega) = H(\omega)U(\omega)$
In $\ell^2(\mathbb{Z})$: $g = h * u$	$\Leftrightarrow$ $g_n = \displaystyle\sum_{k\in\mathbb{Z}} h_k u_{n-k}$	$\Leftrightarrow$	$G(e^{i\omega}) = H(e^{i\omega})U(e^{i\omega})$

We have several Fourier transforms:

$\mathcal{F} : L^2(\mathbb{R}) \to L^2(\mathbb{R})$	$F(\omega) = \langle e_\omega, f\rangle_{L^2(\mathbb{R})}$ $e_\omega(t) = e^{i\omega t}$ $\omega \in \mathbb{R}, t \in \mathbb{R}$	$f(t) = \langle e_t, F\rangle_{L^2(\mathbb{R})}$ $e_t(\omega) = e^{-i\omega t}$ $\omega \in \mathbb{R}, t \in \mathbb{R}$	$\langle f, g\rangle_{L^2(\mathbb{R})}$ $=$ $\langle F, G\rangle_{L^2(\mathbb{R})}$
$\mathcal{F} = \mathcal{Z} : \ell^2(\mathbb{Z}) \to L^2(\mathbb{T})$	$F(z) = \langle e_z, f\rangle_{\ell^2(\mathbb{Z})}$ $e_z(k) = z^k$ $z \in \mathbb{T}, k \in \mathbb{Z}$	$f(n) = \langle e_n, F\rangle_{L^2(\mathbb{T})}$ $e_n(z) = z^{-n}$ $z \in \mathbb{T}, n \in \mathbb{Z}$	$\langle f, g\rangle_{\ell^2(\mathbb{Z})}$ $=$ $\langle F, G\rangle_{L^2(\mathbb{T})}$
or with $\mathsf{F}(\omega) = F(e^{i\omega})$ $\mathcal{F} : \ell^2(\mathbb{Z}) \to L^2_{2\pi}$	$\mathsf{F}(\omega) = \langle e_\omega, f\rangle_{\ell^2(\mathbb{Z})}$ $e_\omega(k) = e^{ik\omega}$ $\omega \in [-\pi, \pi], k \in \mathbb{Z}$	$f(n) = \langle e_n, \mathsf{F}\rangle_{L^2_{2\pi}}$ $e_n(\omega) = e^{-in\omega}$ $\omega \in [-\pi, \pi], n \in \mathbb{Z}$	$\langle f, g\rangle_{L^2(\mathbb{R})}$ $=$ $\langle \mathsf{F}, \mathsf{G}\rangle_{L^2_{2\pi}}$
$\mathcal{F} : \mathbb{C}^N \to \mathbb{C}^N$	$F_k = \langle e_k, f\rangle_{\mathbb{C}^N}$ $e_k(n) = \frac{1}{N} e^{ikn\frac{2\pi}{N}}$ $k, n \in \{0, \ldots, N-1\}$	$f_n = \langle e_n, F\rangle_{\mathbb{C}^N}$ $e_n(k) = e^{-ikn\frac{2\pi}{N}}$ $k, n \in \{0, \ldots, N-1\}$	$\langle f, g\rangle_{\mathbb{C}^N}$ $=$ $N\langle F, G\rangle_{\mathbb{C}^N}$
$\mathcal{F} : L^2_T \to \ell^2(\mathbb{Z})$	$F_k = \langle e_k, f\rangle_{L^2_T}$ $e_k(t) = e^{ikt\frac{2\pi}{T}}$ $k \in \mathbb{Z}, t \in [-\frac{T}{2}, \frac{T}{2}]$	$f(t) = \langle e_t, F\rangle_{\ell^2(\mathbb{Z})}$ $e_t(k) = e^{-ikt\frac{2\pi}{T}}$ $k \in \mathbb{Z}, t \in [-\frac{T}{2}, \frac{T}{2}]$	$\langle f, g\rangle_{L^2_T}$ $=$ $\langle F, G\rangle_{\ell^2(\mathbb{Z})}$

Note the factor N in the case $\mathbb{C}^N$. We could avoid this when we would define the inner product in $\mathbb{C}^N$ as $\langle f, g\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \bar{f}_k g_k$.

2.9 Exercises

1. Prove that if $f \in \ell^2(\mathbb{Z})$ and $h = f_*$ (i.e., $h_k = \overline{f}_{-k}$), then

$$H(z) = \overline{F(1/\overline{z})} = F_*(z)$$

where $H(z) = \mathcal{Z}(h)$ and $F(z) = \mathcal{Z}(f)$ are the z-transforms of h and f respectively.

2. Prove that if $f, g \in \ell^2(\mathbb{Z})$, then

$$\langle f, g\rangle_{\ell^2(\mathbb{Z})} = \langle F, G\rangle_{L^2(\mathbb{T})} = \langle \mathsf{F}, \mathsf{G}\rangle_{L^2_{2\pi}}$$

where $\mathsf{F}(\omega) = F(e^{i\omega}) = \mathcal{F}(f)$ and $\mathsf{G}(\omega) = G(e^{i\omega}) = \mathcal{F}(g)$ are the Fourier transforms of f and g respectively.

3. If $f, g \in \ell^2(\mathbb{Z})$, prove that the z-transform of the convolution
$$h_n = (f * g)_n = \sum_k f_k g_{n-k}$$
is given by $\mathcal{Z}(h) = \mathcal{Z}(f)\mathcal{Z}(g)$.

4. If $f, g \in L^2_{2\pi}$, prove that the Fourier transform of the convolution
$$h(t) = (f * g)(t) = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(\tau) g(t - \tau) d\tau$$
is given by $\mathcal{F}(h) = \mathcal{F}(f)\mathcal{F}(g)$.

5. Prove that the Fourier transform of the pulse train $\delta^T(t) = \sum_{n \in \mathbb{Z}} \delta(t - nT)$ is equal to $\frac{1}{T} \sum_n e^{int\omega_s}$, $\omega_s = \frac{2\pi}{T}$. Then prove that the Fourier transform of the sampled signal $f_s(t) = f(t)\delta^T(t)$ is given by $F_s(\omega) = \frac{1}{T} \sum_n F(\omega - n\omega_s)$ where $F = \mathcal{F}(f)$.

6. (Poisson formula) Let $f \in L^2(\mathbb{R})$ and suppose that
$$\sum_{n \in \mathbb{Z}} f(t + 2n\pi)$$
converges to a continuous function $s \in L^2_{2\pi}$. Then we have the Poisson summation formula
$$s(t) = \sum_{n \in \mathbb{Z}} f(t + 2n\pi) = \frac{1}{\sqrt{2\pi}} \sum_{n \in \mathbb{Z}} F(n) e^{int}$$
where $F = \mathcal{F}(f)$ is the Fourier transform of f.
Hint: write the Fourier series for $s \in L^2_{2\pi}$: $s(t) = \sum_n S_n e^{int}$ with $S_n = \frac{1}{2\pi} \int_{-\pi}^{\pi} s(t) e^{-int} dt$.
Use the fact that the Fourier transform of $f(at)$ is given by $a^{-1}F(\omega/a)$ when $F(\omega)$ is the Fourier transform of $f(t)$, to rewrite the Poisson formula in the form
$$\sum_{k \in \mathbb{Z}} f(t + k) = \sqrt{2\pi} \sum_{k \in \mathbb{Z}} F(2\pi k) e^{i2\pi kt}.$$

7. (Sampling theorem) Suppose the sampling frequency for the signal $h(t)$ is $\omega_s = 2\omega_m = 2\pi/T$, and $H(\omega) = 0$ for $|\omega| > \omega_m$. Show that $H(\omega)$ has the Fourier series expansion
$$H(\omega) = \sum_k h_k e^{-ik\omega T}, \quad h_k = \frac{1}{2\omega_m} \int_{-\omega_m}^{\omega_m} H(\omega) e^{ik\omega T} d\omega = \frac{\sqrt{2\pi}}{2\omega_m} h(kT)$$
where $h(t) = \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} H(\omega) e^{i\omega t} d\omega$. By replacing $H(\omega)$ by the above Fourier expansion, prove that
$$h(t) = \sum_k h(kT) \frac{\sin \omega_m (t - kT)}{\omega_m (t - kT)}.$$

8. Prove that
$$\frac{1}{M}\sum_{k=0}^{M-1} e^{i\frac{2\pi}{M}nk} = 1 \text{ for } n \in M\mathbb{Z} \text{ and } 0 \text{ otherwise.}$$

9. Prove that the Gaussian function is the only function for which an equality holds in the Heisenberg uncertainty principle.

10. For the windowed Fourier transform we have
$$\overline{g(t-b)}f(t) = \frac{1}{\sqrt{2\pi}}\int_{\mathbb{R}} F(\omega,b)e^{i\omega t}d\omega.$$
Why is it not possible to divide by $\overline{g(t-b)}$ to recover $f(t)$?

11. Prove that
$$\frac{1}{N}\sum_{k=0}^{N-1} e^{-ikl\frac{2\pi}{N}} = \delta_l^N$$
that is, it is equal to 0, except when $l \in N\mathbb{Z}$, then the result is equal to 1.

12. Let $f \in L^2(\mathbb{R})$ have norm 1: $\|f\|_{L^2(\mathbb{R})} = 1$. Define g by $g(t) = f(2^n t - k)$ for $t \in \mathbb{R}$ and k an arbitrary real constant. Prove that $\|g\|_{L^2(\mathbb{R})} = 2^{-n/2}$, hence that $h(t) = 2^{n/2}g(t)$ has norm 1.

13. Prove that the Heisenberg product sS is not changed by (1) a dilation: $f(t) \mapsto 2^{j/2}f(2^j t)$, (2) a modulation: $f(t) \mapsto e^{i\omega t}f(t)$, (3) a translation: $f(t) \mapsto f(t-s)$.

14. If $\psi_{a,b}(t) = |a|^{1/2}\psi(a(t-b))$, prove that its Fourier transform is equal to $\Psi_{a,b}(\omega) = |a|^{-1/2}e^{-ib\omega}\Psi(\omega/a)$ where $\Psi(\omega)$ is the Fourier transform of $\psi(t)$.

Chapter 3

Filters

3.1 Definitions

In discrete time one usually speaks of *digital filters*. We shall restrict ourselves to a treatment in discrete time. The treatment of the continuous time case is completely analogous. We leave it as an exercise.

A *filter* is an operator which maps a signal into another signal.

A filter is linear, if it acts as a linear operator i.e.

$$\mathcal{H}(\alpha f + \beta g) = \alpha \mathcal{H} f + \beta \mathcal{H} g, \quad f, g \in \ell;\ \alpha, \beta \in \mathbb{C}.$$

A filter $\mathcal{H}$ is called *shift invariant* or *time invariant* if it commutes with the shift operator: $\mathcal{D}\mathcal{H} = \mathcal{H}\mathcal{D}$. This means that delaying the filtered signal is the same as filtering the delayed signal.

$$\boxed{\textit{All our filters will be assumed to be linear and time invariant.}}$$

The effect of a filter applied to an impulse δ is called its *impulse response*:

$$h = (h_n) = \mathcal{H}\delta.$$

The impulse response is also given by the filter coefficients.

The Fourier transform[1] $H(e^{i\omega}) = \sum_n h_n e^{-in\omega}$ of the impulse response is the *frequency response* of the filter. The frequency response is in general a complex function and has an amplitude and a phase:

$$H(e^{i\omega}) = \sum_n h_n e^{-in\omega} = |H(e^{i\omega})| \exp(i\varphi(\omega)).$$

For linear time invariant filters, we can write the filtering operation as a convolution with its impulse response: if $g = \mathcal{H} f$, then

$$g = \mathcal{H} f = \mathcal{H}(\sum_m f_m \mathcal{D}^m \delta) = \sum_m f_m \mathcal{D}^m \mathcal{H}\delta = \left(\sum_m f_m h_{n-m}\right) = h * f,$$

[1] recall that for digital signals, we often use the notation $\mathsf{H}(\omega) = H(e^{i\omega}) = \sum_n h_n e^{-in\omega} = \mathcal{F}(h_n)$. Note $\mathsf{H}(\omega + \pi) = H(-e^{i\omega})$.

so that such a filter is completely characterized by its impulse response. It can be defined as a linear combination of powers of the shift operator: $\mathcal{H} = \sum_m h_m \mathcal{D}^m$.

Note that if $g = \mathcal{H}f$, then, when $f_n = e^{in\omega}$, we get

$$g_n = \sum_k h_k f_{n-k} = \sum_k h_k e^{i(n-k)\omega} = e^{in\omega} H(e^{i\omega}).$$

Thus $g = H(e^{i\omega})f$. If H is linear and time invariant, then it will transform a sine (or cosine) into a sine (or cosine) (with the same frequency). The functions $e^{ik\omega}$ are the eigenfunctions of the filter.

In the z-domain, the filtering operation corresponds to a multiplication:

$$G(z) = H(z)F(z) \quad \text{or} \quad F(z) \longrightarrow \boxed{\quad H(z) \quad} \longrightarrow G(z).$$

The filter can be seen as a system with input F and output G. Therefore $H(z)$ is often called the *transfer function* of the filter (assuming that the formal series $H(z) = \sum_n h_n z^{-n}$ does indeed converge to a function).

Since the relation $\mathcal{H} \leftrightarrow h \leftrightarrow H$ is one-to-one we shall also speak of the filter H or the filter h, in order not to complicate the notation.

In linear algebra notation, a filtering operation corresponds to a multiplication with a Toeplitz matrix:

$$\mathbf{g} = \mathbf{T}_h \mathbf{f},$$

with

$$\mathbf{T}_h = \begin{bmatrix} & \ddots & \ddots & \ddots & & & \\ \cdots & h_2 & h_1 & \boxed{h_0} & h_{-1} & \cdots & \\ & \cdots & h_2 & h_1 & h_0 & h_{-1} & \cdots \\ & & & \ddots & \ddots & \ddots & \end{bmatrix}$$

(see (2.1)). Note that also matrix $\mathbf{T}_h \leftrightarrow h$ is one-to-one. Here, like in many other instances, we shall have different ways to describe the same phenomenon:

1. in the time domain a filter is an operator $\mathcal{H}$ defining a convolution with the impulse response;

2. in the z-domain a filter is a linear system with transfer function H;

3. and in linear algebra terms, a filter is a linear transformation described by a multiplication with the Toeplitz matrix $\mathbf{T}_h$.

A filter is called *causal* if it does not produce any output before there has been any input. Thus

$$s_n = 0, \quad \forall n < 0 \quad \Rightarrow \quad (\mathcal{H}s)_n = 0, \quad \forall n < 0.$$

A causal filter should have an impulse response with $(\mathcal{H}\delta)_n = 0$ for all $n < 0$. The Toeplitz matrix of a causal filter is lower triangular.

A filter is called *stable* if it transforms a finite energy signal into a finite energy signal:

$$\|f\|^2 < \infty \quad \Rightarrow \quad \|\mathcal{H}f\|^2 < \infty.$$

Since we used the term energy, we are implicitly working in the 2-norm (i.e. in ℓ^2), however this also holds for other norms. For example, if the signals are considered to belong to ℓ^∞, then $\|f\| < \infty$ means that for all $n \in \mathbb{Z}$, $|f_n| < M$ for some positive M. Then if $g = \mathcal{H}f$, it follows that

$$|g_n| \le \sum_k |h_k||f_{n-k}| \le M \sum_k |h_k|$$

and thus, the filter will be stable if and only if $\sum_k |h_k| < \infty$, thus if $h \in \ell^1$. This is sometimes called *bounded-input bounded-output* (BIBO) stability.

Thus for a stable and causal filter, the transfer function $H(z) = \sum_{n=0}^{\infty} h_n z^{-n}$ represents a function which is analytic in $\mathbb{E} = \{z \in \mathbb{C} : |z| > 1\}$. If we are working in ℓ^2, than $H \in L^2(\mathbb{T})$, which means that it is square integrable on $\mathbb{T}$ (it belongs to the Hardy space $H_2(\mathbb{E})$) or equivalently, $\sum_{n=0}^{\infty} |h_n|^2 < \infty$ or $\mathbf{T}_h$ is a lower triangular bounded Toeplitz operator on $\ell^2(\mathbb{Z})$.

A filter is called a *finite impulse response* (FIR) filter if its impulse response dies out after a finite number of time steps: $(\mathcal{H}\delta)_n = 0$ for all $n > N$. If this is not the case, the filter is called an *infinite impulse response* (IIR) filter.

Let a FIR filter be given by $H(e^{i\omega}) = |H(e^{i\omega})|e^{j\varphi(\omega)}$. The filter is called *linear phase* if the phase is linear in ω. Thus $\varphi(\omega) = -\alpha\omega$. It is called general linear phase if $\varphi(\omega) = \alpha\omega + c$ with c a constant. It can be shown that if the filter coefficients are given by $h_0, \ldots, h_N$, then the only nontrivial possibility is that $\alpha = N/2$. It also implies that the filter coefficients are symmetric or antisymmetric, i.e., $h_k = h_{N-k}$ or $h_k = -h_{N-k}$.

Example 3.1.1. [**Moving average filter**] Let us look at a simple example: Suppose the impulse response of the filter is $h_0 = 1/2$, $h_1 = 1/2$ and $h_m = 0$ for all other $m \in \mathbb{Z}$. Thus $\mathcal{H} = (\mathcal{I} + \mathcal{D})/2$. Then $\mathcal{H}s = f$ if $f_n = (s_n + s_{n-1})/2$, $n \in \mathbb{Z}$. The filter replaces all the samples by the average of the sample and its predecessor: it is a *moving average filter.* Taking z-transforms, we have $F(z) = H(z)S(z)$ where $H(z) = (1 + z^{-1})/2$. When $\mathbf{s}$ is the vector of the samples of s, then we can write the filtering operation as a multiplication with a bidiagonal Toeplitz matrix: $\mathbf{f} = \mathbf{Hs}$ where $\mathbf{H}$ is the Toeplitz matrix

$$\mathbf{H} = \begin{bmatrix} \ddots & \ddots & \ddots & & & \\ 0 & \frac{1}{2} & \boxed{\frac{1}{2}} & 0 & & \\ & 0 & \frac{1}{2} & \frac{1}{2} & 0 & \\ & & \ddots & \ddots & \ddots & \end{bmatrix}$$

The boxed element corresponds to the central element at position (0,0). This filter is sometimes called the *Moving average* or *Haar filter.* ◇

3.2 Inverse filter

A filter with z-transform $H(z)$ is invertible if $H(z) \neq 0$ for $z \in \mathbb{T}$. The z-transform of the inverse filter is then $1/H(z)$.

The condition $H(e^{i\omega}) \neq 0$ is necessary, because if we suppose that $H(e^{i\omega_0}) = 0$, then the frequency ω_0 is killed by the filter. Thus if $Y(e^{i\omega}) = H(e^{i\omega})X(e^{i\omega})$, and the input X contains the frequency ω_0, we shall not find this frequency in the output Y because $H(e^{i\omega_0}) = 0$, so that it is impossible to recover the frequency ω_0 in X from the output Y. The filter is not invertible.

Note that if the filter is causal and FIR, then $H(z)$ is a polynomial in z^{-1}, but the inverse of such a filter is in general an infinite series and hence, it will be an IIR filter.

Example 3.2.1. Consider the FIR filter $H(z) = 1 - \beta/z$. Its inverse is given by

$$\frac{1}{H(z)} = \frac{1}{1-\beta/z} = 1 + \beta z^{-1} + \beta^2 z^{-2} + \cdots$$

which converges in $\mathbb{E}$ if $|\beta| < 1$. The inverse is an IIR filter. If $|\beta| > 1$, then the above IIR causal filter is not stable, but we can use the expansion

$$\frac{1}{H(z)} = \frac{1}{1-\beta/z} = -\frac{z}{\beta} - \frac{z^2}{\beta^2} - \frac{z^3}{\beta^3} - \cdots$$

which is a stable, but noncausal filter. ◇

The observations of the previous example hold in general: A causal FIR filter

1. has no inverse if it has a zero on $\mathbb{T}$
2. has a causal inverse if it has all its zeros in $\mathbb{D}$ (the open unit disk)
3. has an anticausal inverse if it has all its zeros in $\mathbb{E}$ (outside the closed unit disk).

It is clear that a causal filter for which $H(z)$ is rational should have all its poles inside $\mathbb{D}$. If its inverse has also all its poles in $\mathbb{D}$, i.e., if $H(z)$ has all its zeros in $\mathbb{D}$, then the filter is called *minimal phase*[2].

3.3 Bandpass filters

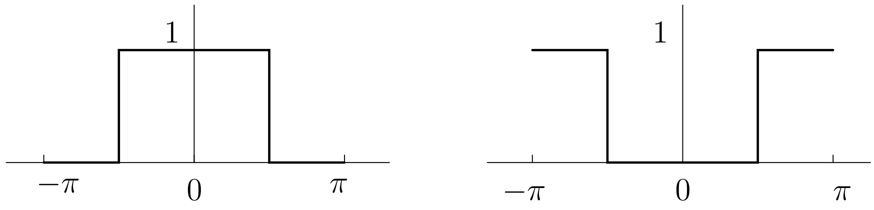
Consider a filter whose Fourier transform is equal to 1 for $|\omega| < \pi/2$ and zero for $|\omega| > \pi/2$. This is an ideal low pass filter. The region $[-\pi/2, \pi/2]$ where it is 1 is called the *passband* and the region $[-\pi, -\pi/2] \cup [\pi/2, \pi]$ where it is zero is called the *stopband*. From $Y(e^{i\omega}) = H(e^{i\omega})X(e^{i\omega})$, it is seen that all the frequencies of X in the passband are passed unaltered to Y, while all the frequencies in the stopband are killed.

Such an ideal filter can never be realized in practice, but in theory, it corresponds to a filter

$$\mathsf{H}(\omega) = H(e^{i\omega}) = \sum_k h_k e^{-ik\omega} = \begin{cases} 1, & 0 \leq |\omega| < \pi/2 \\ 0, & \pi/2 \leq |\omega| < \pi. \end{cases}$$

[2]It can be shown that for a minimal phase filter, the range of the phase angle is minimal among all such filters with the same amplitude response.

Figure 3.1: Ideal low pass and high pass filter



Since

$$h_k = \frac{1}{2\pi}\int_{-\pi}^{\pi} \mathsf{H}(\omega)e^{ik\omega}d\omega = \frac{1}{2\pi}\int_{-\pi/2}^{\pi/2} e^{ik\omega}d\omega = \frac{1}{k\pi}\sin\frac{k\pi}{2} = \frac{1}{2}\operatorname{sinc}\frac{k\pi}{2}$$

for $k \neq 0$ and $h_0 = 1/2$, it follows that this ideal low pass filter has an impulse response that is given by the samples of the sinc function.

$$h_n = \begin{cases} 1/2, & n = 0 \\ \pm 1/(n\pi), & n \text{ odd} \\ 0, & n \text{ even} \end{cases}$$

so that

$$\mathsf{H}(\omega) = H(e^{i\omega}) = \frac{1}{2} + \frac{e^{i\omega}+e^{-i\omega}}{\pi} - \frac{e^{3i\omega}+e^{-3i\omega}}{3\pi} + \frac{e^{5i\omega}+e^{-5i\omega}}{5\pi} + \cdots$$

Similarly, one can construct an ideal high pass filter G

$$\mathsf{G}(\omega) = G(e^{i\omega}) = \sum_k g_k e^{-ik\omega} = \begin{cases} 0, & 0 \le |\omega| < \pi/2 \\ 1, & \pi/2 \le |\omega| < \pi. \end{cases}$$

It has an impulse response

$$g_n = \begin{cases} 1/2, & n = 0 \\ \mp 1/(n\pi), & n \text{ odd} \\ 0, & n \text{ even} \end{cases}$$

Thus

$$\mathsf{G}(\omega) = G(e^{i\omega}) = \frac{1}{2} - \frac{e^{i\omega}+e^{-i\omega}}{\pi} + \frac{e^{3i\omega}+e^{-3i\omega}}{3\pi} - \frac{e^{5i\omega}+e^{-5i\omega}}{5\pi} + \cdots$$

Note that $\mathsf{G}(\omega) = G(e^{i\omega}) = H(-e^{i\omega}) = H(e^{i(\omega+\pi)}) = \mathsf{H}(\omega+\pi)$. These filters are "complementary" in the sense that H covers the lower half of the spectrum and G covers the upper half of the spectrum without overlapping. Together they cover the whole spectrum, a fact which is expressed by

$$|H(e^{i\omega})|^2 + |G(e^{i\omega})|^2 = 1 \quad \text{or} \quad |\mathsf{H}(\omega)|^2 + |\mathsf{G}(\omega)|^2 = 1.$$

Example 3.3.1. [Moving average filter] Let us reconsider the moving average filter

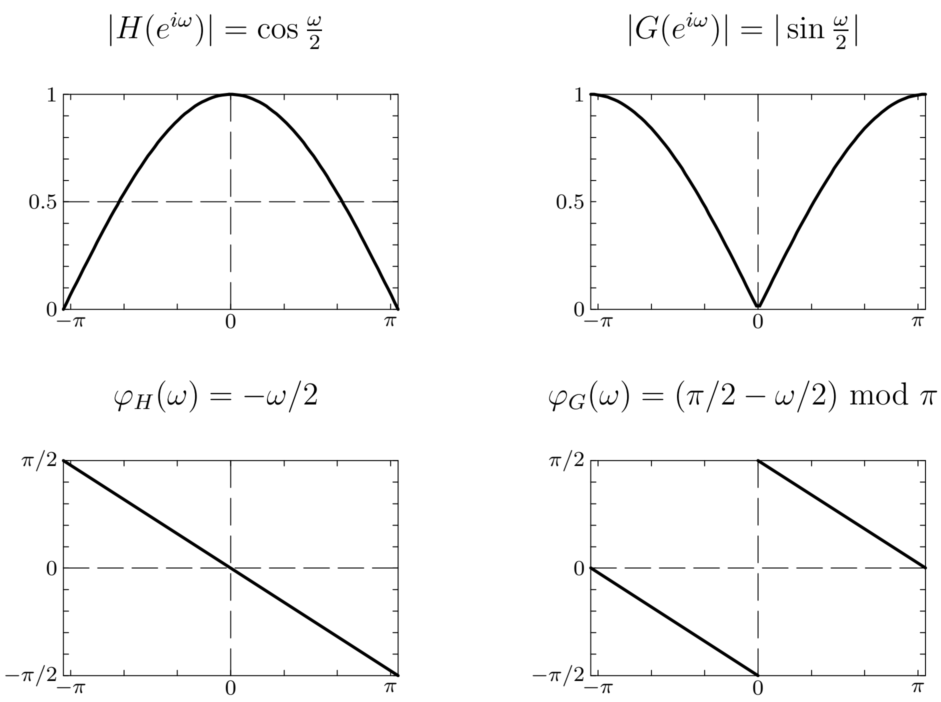
$$\mathsf{H}(\omega) = H(e^{i\omega}) = \frac{1}{2}(1+e^{-i\omega}) = \frac{e^{i\omega/2}+e^{-i\omega/2}}{2}e^{-i\omega/2} = \cos\frac{\omega}{2}e^{-i\omega/2}.$$

Thus amplitude and phase are given by

$$|\mathsf{H}(\omega)| = |H(e^{i\omega})| = \cos\frac{\omega}{2}, \quad \varphi(\omega) = -\frac{\omega}{2}.$$

From the amplitude plot, we see that this filter can be interpreted as an approximate low pass filter: The amplitude is near 1 for $|\omega|$ near zero while it is near zero for $|\omega|$ near π. $\diamond$

Figure 3.2: Amplitude and phase of moving average and moving difference filters



Example 3.3.2. [Moving difference filter] In complete analogy, one may consider the moving difference filter with transfer function $G(z) = \frac{1}{2}(1 - z^{-1})$. Since

$$\mathsf{G}(\omega) = G(e^{i\omega}) = \frac{1}{2}(1 - e^{-i\omega}) = \frac{e^{i\omega/2} - e^{-i\omega/2}}{2} e^{-i\omega/2} = (\sin\frac{\omega}{2})\ (ie^{-i\omega/2}) = \sin\frac{\omega}{2} e^{i(\frac{\pi}{2} - \frac{\omega}{2})}.$$

Thus $|\mathsf{G}(\omega)| = |G(e^{i\omega})| = |\sin\frac{\omega}{2}|$. For obvious reasons, this G can be considered as an approximate high pass filter. $\diamond$

An *all pass* filter $\mathcal{H}$ is a filter for which $|H(e^{i\omega})| = 1$ for all ω. It lets (amplitude of) all the frequencies pass unreduced (although it may change the phase). If $y = \mathcal{H}u$ and $\mathcal{H}$ is an all pass filter, then $|H(e^{i\omega})U(e^{i\omega})| = |U(e^{i\omega})|$ so that

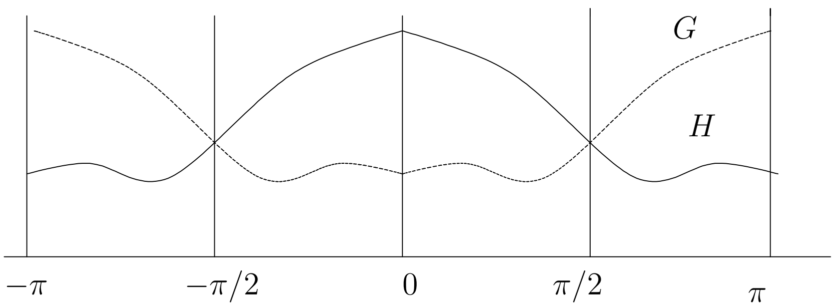
$$\|y\|^2 = \|Y\|^2 = \frac{1}{2\pi}\int_{-\pi}^{\pi} |Y(e^{i\omega})|^2 d\omega = \frac{1}{2\pi}\int_{-\pi}^{\pi} |U(e^{i\omega})|^2 d\omega = \|U\|^2 = \|u\|^2.$$

An all pass filter does not change the energy.

3.4 QMF and PCF filters

Two filters $\mathcal{F}$ and $\mathcal{G}$ are called quadrature mirror filters (QMF) if their frequency response amplitudes $|H(e^{i\omega})|$ and $|G(e^{i\omega})|$ are mirror images to each other with respect to the middle frequency $\frac{\pi}{2}$ (called the mirror frequency). See Figure 3.3. Simple examples are $G(z) = H(-z^{-1})$ (i.e. $G(e^{i\omega}) = H(e^{i(\pi-\omega)})$ or $\mathsf{G}(\omega) = \mathsf{H}(\pi - \omega)$). Or more generally $G(z) = z^N H(-z^{-1})$ or $G(z) = z^N H_*(-z)$ etc.

Figure 3.3: QMF filters



Two filters $\mathcal{F}$ and $\mathcal{G}$ are called *power complementary filters* (PCF) if

$$|\mathsf{H}(\omega)|^2 + |\mathsf{G}(\omega)|^2 \equiv |H(e^{i\omega})|^2 + |G(e^{i\omega})|^2 = \text{constant}. \tag{3.1}$$

The ideal low pass and the ideal high pass filters are QMF and PCF. Also the Haar filters or moving average and moving difference are QMF and PCF.

3.5 Exercises

1. If $H(z)$ is the rational transfer function of a digital filter, show that the filter is stable and causal when all the poles of $H(z)$ are outside the unit disk (i.e., they are all in $\mathbb{E}$). Under what conditions will the inverse of this filter be causal and stable?

2. Prove that for an analog signal, an ideal low pass filter, i.e., a filter for which $H(\omega) = 1$ for $|\omega| < \omega_0$ and $H(\omega) = 0$ for $|\omega| > \omega_0$ is given by

$$h(t) = \frac{2\omega_0}{\sqrt{2\pi}} \operatorname{sinc}(\omega_0 t).$$

3. Prove that we have QMF filters if $G(z) = H(-z^{-1})$ or $G(z) = z^N H(-z^{-1})$ or $G(z) = z^N H_*(-z)$. Show that if $g_k = (-1)^k \bar{h}_{N-k}$, then we have QMF filters.

4. **(linear phase filters)** Assume that the real filter coefficients $h_0, \ldots, h_N$ of a FIR filter $H(z)$ satisfy $h_k = h_{N-k}$. Moreover assume that N is even. Show that the amplitude

and phase are given by

$$|H(e^{i\omega})| = h_{N/2} + 2\sum_{k=0}^{N/2-1} h_k \cos\frac{(N-2k)\omega}{2} \quad \text{and} \quad \varphi(\omega) = -\frac{N}{2}\omega.$$

Show that if $h_k = -h_{N-k}$, then amplitude and phase are given by

$$|H(e^{i\omega})| = 2\sum_{k=0}^{N/2-1} h_k \sin\frac{(N-2k)\omega}{2} \quad \text{and} \quad \varphi(\omega) = -\frac{N}{2}\omega + \frac{\pi}{2}.$$

How about the case where N is odd?

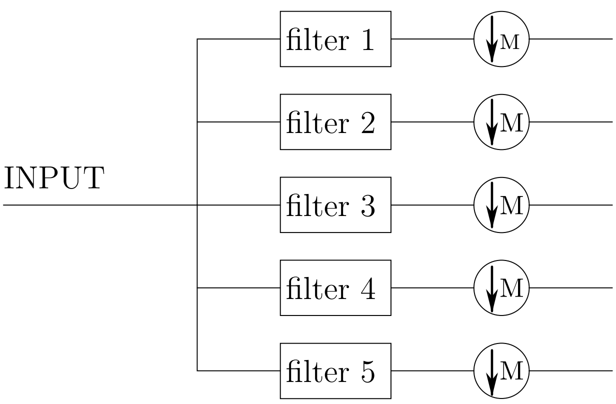
Chapter 4

Filter banks

4.1 Analysis and synthesis

Suppose we have a discrete or an analog signal s which is band limited with band width Ω. We want to write the signal as the sum of M signals, each of which have a band width Ω/M. In this way, a wide band signal can be split into M signals of a smaller band and transmitted over a channel with smaller band width. The receiver can reconstruct the original signal. In theory, one can compute the Fourier transform of the signal, cut this in M pieces and backtransform. In practice this is obtained by applying M filters to the signal, each of these filters generates one of the M signals with the limited band width. This is called an M channel filter bank.

Figure 4.1: Five channel filter bank

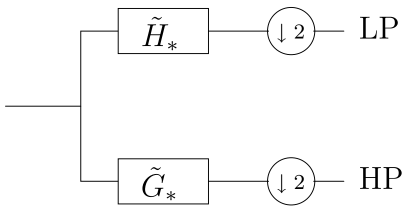


We shall restrict our discussion to the case $M = 2$ for discrete signals (although most of the discussion is true for analog signals as wel).

Suppose we have a discrete signal, then we can apply a low pass and a high pass filter to it, which splits the signal in two parts: the part which contains the low frequencies, which gives a low resolution idea of the signal and the other part, which contains the high frequencies and this part gives the detail information.

This is called a two-channel filter bank. It splits the frequency band in two subbands.

Figure 4.2: Two channel filter bank



Since in the ideal situation, each of the two filters take half of the frequency band of the original signal, it is possible to stretch the half bands again to the full bandwidth. This is obtained by downsampling the signal. Let us illustrate this idea.

If $s = (s_n)$ is a given signal, then $s' = \downarrow s$ if $s'_n = s_{2n}$. Thus we delete the odd samples and keep only the even ones. The arrow indicates that we *decimate* or *subsample* or *downsample* the signal. In general one writes $s' = (\downarrow M)s$ if $s'_n = s_{nM}$, but since we shall only subsample here by a factor of 2, we leave it out of the notation.

In the z-domain, this means that

$$s' = \downarrow s \quad \Leftrightarrow \quad S'(z^2) = \frac{S(z) + S(-z)}{2} \quad \Leftrightarrow \quad S'(z) = \frac{S(z^{1/2}) + S(-z^{1/2})}{2}.$$

In the frequency domain, this reads

$$S'(e^{i\omega}) = \frac{S(e^{i\omega/2}) + S(-e^{i\omega/2})}{2}$$

which clearly shows that if the bandwidth of S is π, then the bandwidth of S' is 2π.

For the high pass band $\pi/2 \leq |\omega| < \pi$, we have to shift the spectrum first to the low pass band $|\omega| < \pi/2$, which corresponds to adding π to ω (note that because of periodicity, we have a wrap around here). This means that for the shifted spectrum the frequency response is given by

$$\sum s_n e^{-in(\omega+\pi)} = \sum (-1)^n s_n e^{-in\omega}$$

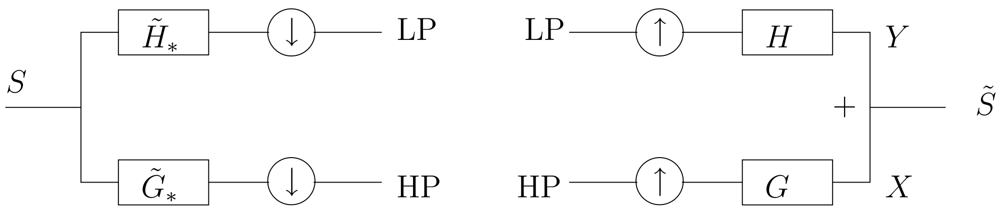
so that sampling at half the clock rate of this signal gives again the same formula as for the low pass band.

Thus on the analysis side of a two-channel filter bank, we have the application of the two filters $\tilde{H}_*$ and $\tilde{G}_*$ which are both followed by a decimation operation. We use the notation $\tilde{H}_*$ to denote that the transfer function of this filter is $\tilde{H}_*(z) = \sum_k \overline{\tilde{h}_k} z^k$ and similarly for $\tilde{G}_*$. Observe that $\tilde{H}_*(z)$ is the z-transform of the time reversed sequence $\tilde{h}_*$ where $\tilde{h} = (\tilde{h}_k)$.

On the synthesis side of the filter bank one finds the mirror image of the analysis side. First the signals are *upsampled*. This means that between every two samples a zero is introduced. This is denoted as

$$s' = \uparrow s \quad \Leftrightarrow \quad s'_{2n} = s_n \text{ and } s'_{2n+1} = 0 \quad \Leftrightarrow \quad S'(z) = S(z^2).$$

Figure 4.3: Analysis and synthesis



After the upsampling, some filters H and G are applied and the two resulting signals Y and X are added. Ideally the synthesis side should undo the analysis computations such that the resulting signal $\tilde{S}$ is again equal to the original S.

The operations of the filter bank can also be written in terms of (infinite) matrices. Indeed, filtering by $\tilde{H}_*$ means multiplication with the infinite Toeplitz matrix with entries the impulse response $\tilde{h}_* = (\bar{\tilde{h}}_{-n})$, i.e., the Fourier coefficients of $\tilde{H}_*$. Downsampling means that we skip every other row (the odd ones). Thus the operation in the upper branch of the analysis part gives $LP(z)$ as the result of multiplication with the *adjoint* of the matrix

$$\tilde{\mathbf{H}} = \begin{bmatrix} \ddots & & & & & & & & \\ \cdots & \tilde{h}_5 & \tilde{h}_3 & \tilde{h}_1 & \tilde{h}_{-1} & \cdots & & & \\ & \cdots & \tilde{h}_4 & \tilde{h}_2 & \boxed{\tilde{h}_0} & \tilde{h}_{-2} & \cdots & & \\ & & \cdots & \tilde{h}_3 & \tilde{h}_1 & \tilde{h}_{-1} & \tilde{h}_{-3} & \cdots & \\ & & & & & & & \ddots & \end{bmatrix} \tag{4.1}$$

A similar discussion holds for the lower branch. Thus the vector $\mathbf{p}$ of samples of $LP(z)$ and the vector $\mathbf{q}$ of samples of $HP(z)$ are obtained from the samples $\mathbf{s}$ of $S(z)$ as follows

$$\begin{bmatrix} \mathbf{p} \\ \mathbf{q} \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{H}}^* \\ \tilde{\mathbf{G}}^* \end{bmatrix} \mathbf{s} \equiv \tilde{\mathbf{K}}^* \mathbf{s}$$

On the synthesis side, the upsampling followed by filtering with H means that we multiply with the Toeplitz matrix whose entries are the impulse response coefficients (i.e. the Fourier coefficients of $H(z)$) and in which every other column is deleted (the odd ones). That is the matrix $\mathbf{H}$ which is defined like $\tilde{\mathbf{H}}$ but without the tildes. The matrix $\mathbf{G}$ can be defined similarly for the other branch on the synthesis side. The samples $\tilde{\mathbf{s}}$ of the result $\tilde{S}(z)$ are then computed from the samples of $LP(z)$ and $HP(z)$ by

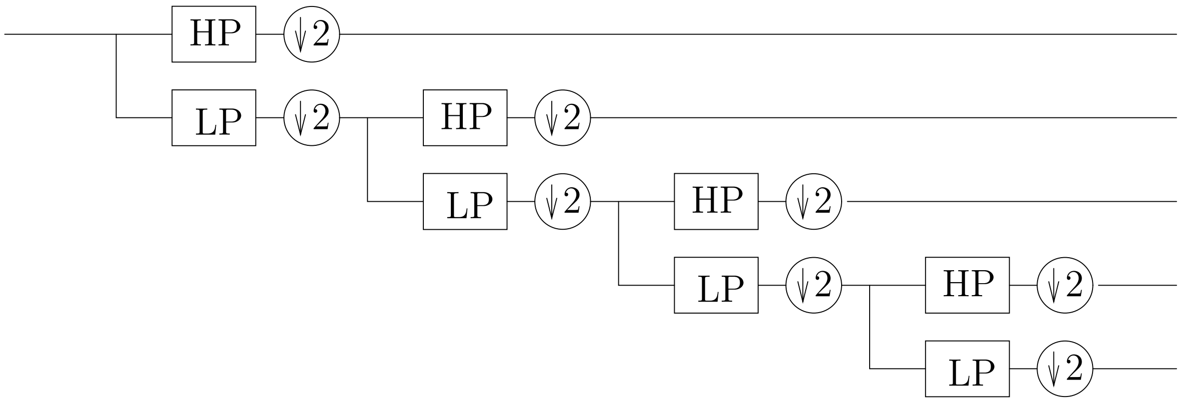
$$\tilde{\mathbf{s}} = \mathbf{H}\mathbf{p} + \mathbf{G}\mathbf{q} = [\mathbf{H} \;\; \mathbf{G}] \begin{bmatrix} \mathbf{p} \\ \mathbf{q} \end{bmatrix} \equiv \mathbf{K} \begin{bmatrix} \mathbf{p} \\ \mathbf{q} \end{bmatrix}.$$

We shall have $\tilde{\mathbf{s}} = \mathbf{s}$ if

$$\mathbf{K}\tilde{\mathbf{K}}^* = \mathbf{I}. \tag{4.2}$$

Before moving to the next section, we remark that the recursive application of a two channel filter bank as in Figure 4.4 also leads to an M channel filter bank. If the 2-channel filter banks split in equal band widths, then the band widths of the end channels will not be the same though.

Figure 4.4: Recursive 2 channel filter bank



4.2 Perfect reconstruction

After the analysis of the signal, all kinds of operations can be done. For example, the signal can be encoded, compressed and sent over a transmission channel. At the end, the receiver will reconstruct the signal by synthesis.

Ideally, if the operations in between do not loose information, then one should hope that the filters $\tilde{G}_*, \tilde{H}_*, G$ and H are designed such that the synthesized signal equals the original signal. This is called *perfect reconstruction* (PR) and the filter bank is then called a PR filter bank. What are the conditions to be imposed on the filters to guarantee a PR property? In terms of the matrices, we have already formulated the condition as (4.2). It is however not so easy to derive from (4.2) conditions for the four filters involved.

Therefore we do the analysis in the z-domain. Let us write down the operations of the analysis stage in the z-domain.

$$LP(z) = \frac{\tilde{H}_*(z^{1/2})S(z^{1/2}) + \tilde{H}_*(-z^{1/2})S(-z^{1/2})}{2}$$

and

$$HP(z) = \frac{\tilde{G}_*(z^{1/2})S(z^{1/2}) + \tilde{G}_*(-z^{1/2})S(-z^{1/2})}{2}.$$

Now extend the notion of paraconjugate for matrices as follows,

$$A(z) = \begin{bmatrix} a(z) & b(z) \\ c(z) & d(z) \end{bmatrix} \mapsto A_*(z) = \begin{bmatrix} a_*(z) & c_*(z) \\ b_*(z) & d_*(z) \end{bmatrix},$$

thus we take the paraconjugate of the entries and the transpose of the matrix. Recall that for a scalar function, the paraconjugate means $f_*(z) = \overline{f(1/\overline{z})}$. Then we can write

$$\begin{bmatrix} LP(z^2) \\ HP(z^2) \end{bmatrix} = \frac{1}{2}\tilde{M}_*(z)^t \begin{bmatrix} S(z) \\ S(-z) \end{bmatrix} \quad \text{where} \quad \tilde{M}(z) = \begin{bmatrix} \tilde{H}(z) & \tilde{H}(-z) \\ \tilde{G}(z) & \tilde{G}(-z) \end{bmatrix}.$$

This $\tilde{M}(z)$ is called the *modulation matrix* of the analysis part.

On the synthesis side, we have

$$Y(z) = H(z)LP(z^2) \quad \text{and} \quad X(z) = G(z)HP(z^2).$$

Thus the reconstructed signal is

$$\begin{aligned} \tilde{S}(z) &= T_0(z)S(z) + T_1(z)S(-z) \\ T_0(z) &= \frac{1}{2}[H(z)\tilde{H}_*(z) + G(z)\tilde{G}_*(z)] \\ T_1(z) &= \frac{1}{2}[H(z)\tilde{H}_*(-z) + G(z)\tilde{G}_*(-z)]. \end{aligned}$$

Thus

$$\tilde{S}(z) = [H(z)\ G(z)] \begin{bmatrix} LP(z^2) \\ HP(z^2) \end{bmatrix} = [H(z)\ G(z)]\frac{1}{2}\tilde{M}_*(z)^t \begin{bmatrix} S(z) \\ S(-z) \end{bmatrix}$$

By symmetry, we also have

$$\begin{bmatrix} \tilde{S}(z) \\ \tilde{S}(-z) \end{bmatrix} = M(z)^t \begin{bmatrix} LP(z^2) \\ HP(z^2) \end{bmatrix} = \frac{1}{2}M(z)^t\tilde{M}_*(z)^t \begin{bmatrix} S(z) \\ S(-z) \end{bmatrix}$$

where

$$M(z) = \begin{bmatrix} H(z) & H(-z) \\ G(z) & G(-z) \end{bmatrix}$$

is the modulation matrix on the synthesis side.

For PR we want $\tilde{S}(z) = cz^{-n_0}S(z)$. The c is a constant. Usually, it is equal to 1, but another constant would only represent a scaling, which of course does not loose information. We shall take $c = 1$. The z^{-n_0} represents a delay, which is only natural to allow because the computations with the filters will need at least a few clock cycles. However, we shall shift the filters such that $n_0 = 0$ (we can always shift them back later). Indeed, if the filters are FIR and causal, they should be polynomials in z^{-1}. However, by multiplication with a power of z, they become noncausal (Laurent polynomials containing powers of z and z^{-1}). For mathematical manipulation, this is the simplest formulation. When the filters are realized, one takes the filter coefficients of the Laurent polynomials, but uses them as coefficients of polynomials in z^{-1}, thus they are implemented as causal filters and this causes the delay in the filter bank. Thus, in conclusion, we impose the PR conditions by setting

$$M(z)^t\tilde{M}_*(z)^t = 2I_2 \quad \text{or} \quad \tilde{M}_*(z)M(z) = 2I_2.$$

4.3 Lossless filter bank

From the PR condition, it is seen that up to a shift and a scaling, M and $\tilde{M}_*$ should be inverses of each other. Assume that the filters G and H are FIR, which means that they are Laurent polynomials (in the causal case polynomials in z^{-1}). Since the inverse of a polynomial is not a polynomial, but gives rise to an IIR filter, $\tilde{M}_*$, being the inverse of a polynomial matrix, will contain IIR filters $\tilde{G}_*$ and $\tilde{H}_*$.

To avoid this problem, there are several solutions. A popular one is to require that the filter bank is lossless or paraunitary. This is expressed by the fact that the modulation matrix is proportional to a lossless or paraunitary matrix. In general, a polynomial matrix $A(z)$ is called *paraunitary* if it satisfies $A_*(z)A(z) = A(z)A_*(z) = I$. We shall require that $M(z)$ is paraunitary up to a scaling $\sqrt{2}$, namely we assume that $M_*(z)M(z) = 2I_2$.

For a paraunitary filter bank, we see that the PR condition becomes extremely simple, because we can choose

$$\tilde{M}(z) = M(z)$$

and thus $\tilde{M}_*(z)$ will contain FIR filters if $M(z)$ consists of FIR filters. For computational reasons, this is of course a most desirable situation. Note that it means that $\tilde{H}(z) = H(z)$ and $\tilde{G}(z) = G(z)$.

That paraunitarity is a strong restriction can be illustrated as follows. The filters from a paraunitary filter bank satisfy $M(z)M_*(z) = 2I$, i.e.

$$\begin{aligned} H_*(z)H(z) + H_*(-z)H(-z) &= 2 \\ G_*(z)G(z) + G_*(-z)G(-z) &= 2 \\ H_*(z)G(z) + H_*(-z)G(-z) &= 0 \end{aligned}$$

Thus, if $R_H(z) = H_*(z)H(z)$ is the power spectrum of H, and if $R_G(z) = G_*(z)G(z)$ is the power spectrum of G, and $R_{HG}(z) = H_*(z)G(z)$, then, the above conditions can be expressed as

$$\begin{aligned} R_H(z) + R_H(-z) &= 2 \\ R_G(z) + R_G(-z) &= 2 \\ R_{HG}(z) + R_{HG}(-z) &= 0 \end{aligned}$$

The first condition means

$$\sum_n r_n^H z^{-n} + \sum_n r_n^H (-1)^n z^{-n} = 2 \sum_n r_{2n}^H z^{-2n} = 2$$

or $r_0^H = 1$ and $r_{2n}^H = 0$. In other words,

$$r_{2n}^H \equiv \sum_k \overline{h}_k h_{k-2n} = \delta_n.$$

The impulse response and its even translates are orthonormal. This is called *double shift orthogonality.* Similarly, the second condition gives a double shift orthogonality for (g_k)

$$\sum_k \overline{g}_k g_{k-2n} = \delta_n.$$

The third condition leads to

$$\sum_k \bar{h}_k g_{k-2n} = 0. \tag{4.3}$$

For $n = 0$ this expresses that the filters H and G are orthogonal. To satisfy the condition (4.3) one often chooses $g_k = (-1)^k \bar{h}_{N-k}$ with N some odd integer. This relation is called *alternating flip*. For example, taking $N = 3$, we see that with this rule the inner product of the sequences

$$\begin{array}{ccccccccc} \cdots & \bar{h}_{-2} & \bar{h}_{-1} & \boxed{\bar{h}_0} & \bar{h}_1 & \bar{h}_2 & \bar{h}_3 & \cdots \\ \cdots & \bar{h}_5 & -\bar{h}_4 & \boxed{\bar{h}_3} & -\bar{h}_2 & \bar{h}_1 & -\bar{h}_0 & \cdots \end{array}$$

gives indeed zero. This means that G and H are QMF.

The previous relations can be easily expressed in terms of the matrices of Section 4.1:

$$\mathbf{H}^*\mathbf{H} = \mathbf{G}^*\mathbf{G} = \mathbf{I} \quad \text{and} \quad \mathbf{H}^*\mathbf{G} = \mathbf{0}.$$

Note also that $M_*(z)M(z) = 2I$ implies

$$H_*(z)H(z) + G_*(z)G(z) = 2,$$

so that G and H are PCF. By the alternating flip relation $G(z) = -z^{-N}H_*(-z)$, so that they are also QMF.

4.4 Polyphase matrix

Our previous description of the filter bank in terms of the modulation matrix is somewhat inefficient because it describes the analysis as applying filters $\tilde{H}_*$ and $\tilde{G}_*$ after which the filtered signals are subsampled. Thus half of the work is thrown away. From the description with the matrices $\tilde{\mathbf{H}}$ and $\tilde{\mathbf{G}}$ it became clear that this is not the way in which it is implemented. A more appropriate description will be given by the polyphase matrix which describes the same filter bank, but where (sub/up)sampling and filtering operations are interchanged, which turns out to be more efficient.

If $S(z)$ represents any signal, then

$$S(z) = \sum_n s_n z^{-n} = \sum_n s_{2n} z^{-2n} + z^{-1} \sum_n s_{2n+1} z^{-2n} = S_e(z^2) + z^{-1} S_o(z^2).$$

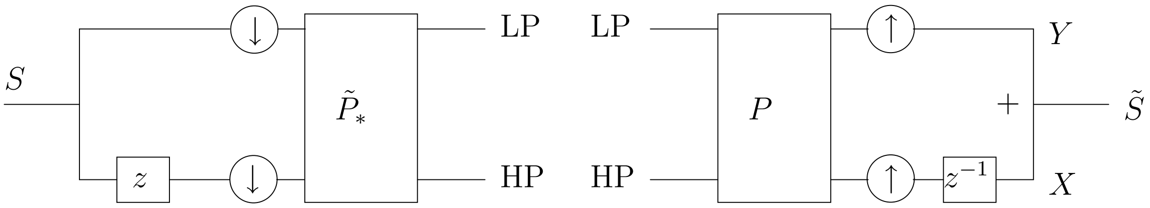
Note that

$$S_e(z^2) = \frac{1}{2}[S(z) + S(-z)] \quad \text{and} \quad S_o(z^2) = \frac{z}{2}[S(z) - S(-z)].$$

The signal S is split into 2 parts S_e and S_o because the signal in our filter bank has only 2 channels. This idea can be generalized to a filter bank with M channels. We restrict ourselves to a 2 channel case. We shall now give a polyphase (2-phase in our case) representation of the two-channel filter bank.

Our objective is to describe the filter bank by a block diagram like in Figure 4.5. It should be clear that the signals on the left obtained immediately after downsampling in the

Figure 4.5: Polyphase representation of filter bank



analysis part are $S_e(z)$ in the top branch and $S_o(z)$ in the lower branch. Similarly, it can be seen that the signals on the right obtained immediately after the application of P should be $\tilde{S}_e(z)$ in the top branch and $\tilde{S}_o(z)$ in the lower branch.

Thus if $S = \tilde{S}$ for PR, then we should have $P(z)\tilde{P}_*(z) = I$.

What are these transformation matrices $P(z)$ and $\tilde{P}_*(z)$?

Define the almost paraunitary matrix

$$T(z) = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ z & -z \end{bmatrix}, \quad \text{then} \quad T(z)T_*(z) = \frac{1}{2}I_2.$$

Then for any S

$$\begin{bmatrix} S_e(z^2) \\ S_o(z^2) \end{bmatrix} = T(z)\begin{bmatrix} S(z) \\ S(-z) \end{bmatrix}.$$

Note that

$$T(z)^{-1} = \begin{bmatrix} 1 & z^{-1} \\ 1 & -z^{-1} \end{bmatrix} = 2T_*(z).$$

Comparing

$$\begin{bmatrix} \tilde{S}(z) \\ \tilde{S}(-z) \end{bmatrix} = M(z)^t \begin{bmatrix} LP(z^2) \\ HP(z^2) \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} \tilde{S}_e(z^2) \\ \tilde{S}_o(z^2) \end{bmatrix} = P(z^2)\begin{bmatrix} LP(z^2) \\ HP(z^2) \end{bmatrix}$$

it follows by multiplying the first relation by $T(z)$ from the left and comparing with the second one that

$$P(z^2) = T(z)M(z)^t,$$

whence

$$P(z) = \begin{bmatrix} H_e(z) & G_e(z) \\ H_o(z) & G_o(z) \end{bmatrix}.$$

This $P(z)$ is called the *polyphase matrix* for the filters H and G.

Similarly, from

$$\begin{bmatrix} LP(z^2) \\ HP(z^2) \end{bmatrix} = \frac{1}{2}\tilde{M}_*(z)^t \begin{bmatrix} S(z) \\ S(-z) \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} LP(z^2) \\ HP(z^2) \end{bmatrix} = \tilde{P}_*(z^2)\begin{bmatrix} S_e(z^2) \\ S_o(z^2) \end{bmatrix}$$

it follows by using

$$\frac{1}{2}\tilde{M}_*(z)^t = \frac{1}{2}\tilde{M}_*(z)^t[2T_*(z)T(z)] = [T(z)\tilde{M}(z)^t]_*T(z),$$

that

$$\tilde{P}_*(z^2) = \tilde{M}_*(z)^t T_*(z) = [T(z)\tilde{M}(z)^t]_*.$$

so that the polyphase matrix for the analyis side is $\tilde{P}(z^2) = T(z)\tilde{M}(z)^t$ with

$$\tilde{P}(z) = \begin{bmatrix} \tilde{H}_e(z) & \tilde{G}_e(z) \\ \tilde{H}_o(z) & \tilde{G}_o(z) \end{bmatrix}.$$

In this notation, the perfect reconstruction condition $\tilde{M}_*(z)M(z) = 2I_2$ becomes the condition $P(z)\tilde{P}_*(z) = I$.

In the paraunitary case however where $M = \tilde{M}$, this becomes

$$P(z) = \tilde{P}(z)$$

and the PR condition is just $P(z)P_*(z) = I$ which is obviously satisfied if P is a paraunitary matrix.

From this condition, it is seen that the alternating flip is the way to relate H and G in the paraunitary case. If P is paraunitary and contains Laurent polynomials, then its inverse, which is $P^{-1} = P_*$, should also contain Laurent polynomials. Because $\det P^{-1} = 1/\det P$, it follows from Cramer's rule for the inversion of a 2 by 2 matrix that $\det P(z)$ should be a monomial, thus $\det P(z) = cz^{-m}$ where c is a constant that can only be unimodular: $|c| = 1$. Let us assume without loss of generality that $c = -1$. By $P^{-1} = P_*$, we have

$$-z^m \begin{bmatrix} G_o(z) & -G_e(z) \\ -H_o(z) & H_e(z) \end{bmatrix} = \begin{bmatrix} H_{e*}(z) & H_{o*}(z) \\ G_{e*}(z) & G_{o*}(z) \end{bmatrix}.$$

Thus

$$\begin{aligned} G(z) &= G_e(z^2) + z^{-1}G_o(z^2) \\ &= -z^{-2m}[-H_{o*}(z^2) + z^{-1}H_{e*}(z^2)] \\ &= -z^{-(2m+1)}H_*(-z). \end{aligned}$$

This is precisely the alternating flip relation. Note that this implies the previously derived relation $H(z) = -z^{-(2m+1)}G_*(-z)$.

Why is the polyphase implementation of the filter bank more interesting than the original one? This is a matter of efficiency. In the analysis phase, the filters H_* and G_* are applied to the given signal, and the two subbands are computed. Then both are subsampled. Thus half the work is thrown away. In the polyphase implementation, the signal is split into its even and its odd part, that is, the subsampling is done *before* the filters are applied, which gives a reduction in computation. The downsampling is moved from after the filters to before the filters. On the synthesis side, a symmetric image of this interchange is in order.

4.5 Note on orthogonality

The PR condition can be expressed in different ways:

Polyphase matrix condition	$P(z)\tilde{P}_*(z) = I$
modulation matrix condition	$M(z)\tilde{M}_*(z) = 2I$
matrix $\mathbf{K}$ condition	$\mathbf{K}\tilde{\mathbf{K}}^* = \mathbf{I}$.

For a paraunitary filter bank we have

Polyphase matrix is paraunitary	$P(z)P_*(z) = I$
modulation matrix is paraunitary up to factor 2	$M(z)M_*(z) = 2I$
matrix $\mathbf{K}$ is unitary	$\mathbf{K}\mathbf{K}^* = \mathbf{I}$.

Thus by choosing

$$\tilde{P}(z) = P(z) \quad \Leftrightarrow \quad \tilde{M}(z) = M(z) \quad \Leftrightarrow \quad \tilde{\mathbf{K}} = \mathbf{K}$$

we satisfy the PR condition

$$P(z)\tilde{P}_*(z) = I \quad \Leftrightarrow \quad \tilde{M}_*(z)M(z) = 2I \quad \Leftrightarrow \quad \mathbf{K}\tilde{\mathbf{K}}^* = \mathbf{I}.$$

The condition $\mathbf{K}\mathbf{K}^* = \mathbf{I}$ means that we have the double shift orthogonalities

$$\mathbf{H}^*\mathbf{H} = \mathbf{I} \;:\; \sum \bar{h}_n h_{n-2k} = \delta_k \tag{4.4}$$

$$\mathbf{H}^*\mathbf{G} = \mathbf{0} \;:\; \sum \bar{h}_n g_{n-2k} = 0 \tag{4.5}$$

$$\mathbf{G}^*\mathbf{G} = \mathbf{I} \;:\; \sum \bar{g}_n g_{n-2k} = \delta_k. \tag{4.6}$$

The condition (4.5) is satisfied by choosing alternating flips relating G and H: $g_n = (-1)^n \bar{h}_{N-n}$ with N odd. This is equivalent with $G(z) = -z^N H_*(-z)$: so that G and H are QMF. The condition (4.6) is then automatically satisfied if condition (4.4) holds: h should be orthogonal to its even shifts (double shift orthogonatyl).

In terms of the H and G from the modulation matrix, condition (4.4) translates into

$$H(z)H_*(z) + H(-z)H_*(-z) = 2 \quad \text{or} \quad |H(z)|^2 + |H(-z)|^2 = 2, \quad z \in \mathbb{T}.$$

In terms of H_e and H_o of the polyphase formulation, it reads

$$H_e(z)H_{e*}(z) + H_o(-z)H_{o*}(-z) = 1 \quad \text{or} \quad |H_e(z)|^2 + |H_o(-z)|^2 = 1, \quad z \in \mathbb{T}.$$

Thus, if in a paraunitary filter bank, one of the latter conditions are satisfied and if G and H are related by alternating flips, we have an (orthogonal) PR QMF filter bank.

It should be noted that taking a paraunitary filter bank to get PR is an overkill. This forces for instance the matrix $\mathbf{K}$ to be unitary. For PR we only needed $\mathbf{K}\tilde{\mathbf{K}}^* = \mathbf{I}$. Thus we can choose two different matrices $\mathbf{K}$ and $\tilde{\mathbf{K}}$ which are *biorthogonal*. We actually need

$$\begin{aligned}
H(z)\tilde{H}_*(z) + H(-z)\tilde{H}_*(-z) &= 2 \\
G(z)\tilde{G}_*(z) + G(-z)\tilde{G}_*(-z) &= 2 \\
H(z)\tilde{G}_*(z) + H(-z)\tilde{G}_*(-z) &= 0 \\
G(z)\tilde{H}_*(z) + G(-z)\tilde{H}_*(-z) &= 0.
\end{aligned}$$

The last two relations hold if we choose mixed alternating flips

$$H(z) = -z^N \tilde{G}_*(-z) \quad \text{and} \quad G(z) = -z^N \tilde{H}_*(-z)$$

with N odd. This leads to $\mathbf{H}^*\tilde{\mathbf{G}} = \mathbf{0} = \mathbf{G}^*\tilde{\mathbf{H}}$. One only has to satisfy then the biorthogonality relation

$$H(z)\tilde{H}_*(z) + H(-z)\tilde{H}_*(-z) = 2$$

or $\mathbf{H}\tilde{\mathbf{H}}^* = \mathbf{I}$ to get the required PR.

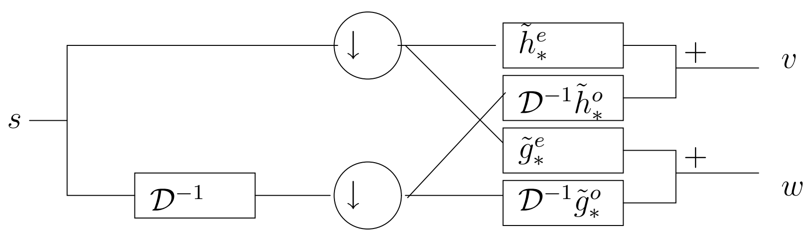
This idea will be the basis of the biorthogonal wavelets, which will be discussed later. First we shall discuss the orthogonal wavelets that correspond to the choice of a paraunitary filter bank. The mathematical framework will be multiresolution, considered in the next chapter.

4.6 Exercises

1. What is the matrix representation of the downsampling operation $(\downarrow 2)$? What is the operation that the transpose of this matrix will represent?

2. Can you go through our analysis for the two channel filter bank again, now assuming that it is an M channel filter bank?

3. Derive the polyphase description in the time domain. This means the following. Consider the filter bank of Figure 4.3. The input of the filter bank is the signal $s = (s_j)$. Define $s^e = (\downarrow 2)s$ and $s^o = (\downarrow 2)\mathcal{D}^{-1}s$. Thus $s^2_j = s_{2j}$ and $s^o_j = s_{2j+1}$. The filter $\tilde{H}_*$ has coefficients $\overline{\tilde{h}}_{-k}$ and the filter $\tilde{G}_*$ has coefficients $\overline{\tilde{g}}_{-k}$. If we denote the LP and HP signals as v and w respectively, then show that $s = \tilde{h}^e_* * s^e + \mathcal{D}\tilde{h}^o_* * s^o$ and $w = \tilde{g}^e_* * s^e + \mathcal{D}\tilde{g}^o_* * s^o$. Which gives the scheme



This is the scheme that corresponds to the "black box" representation $\tilde{P}_*$ of Figure 4.5. Check that this indeed the same! Do a similar analysis in the time domain for the synthesis phase.

Chapter 5

Multiresolution

5.1 Introduction

Consider a 2π-periodic signal, thus a signal from the time domain $L^2_{2\pi}$. As we know, this can be represented in terms of the basis functions $\{e^{ikt}\}_{k\in\mathbb{Z}}$: $f(t) = \sum_{k\in\mathbb{Z}} f_k e^{ikt}$ with Fourier coefficients $f_k = \left\langle e^{ikt}, f\right\rangle$. If we do not want basis functions with infinite support on the time axis, we should replace this by basis functions with a finite width on the time and on the frequency axis. If we want to include all frequencies, we can do this by contraction and stretching one "mother" function $\psi(t)$. This corresponds for example to considering basis functions of the form $\psi(2^n t)$, $n \in \mathbb{Z}$. To catch high frequencies, n should be large and thus these functions will have a narrow support on the time axis. This is confirmed by the Heisenberg uncertainty principle since a large width S in the frequency domain implies a small width s in the time domain. These narrow functions will not be able to cover the whole time support. Thus we have to translate these basis functions as well, so that they can cover the whole time-support. Therefore, we consider a two-parameter family of functions: $\psi(2^n t - k)$, $n, k \in \mathbb{Z}$. Suppose that in $L^2_{2\pi}$: $\|\psi\|^2 = 1$, then clearly $\|\psi(2^n t)\|^2 = 2^{-n}$, thus to have a normalized basis, we need to consider the basis functions

$$\psi_{nk}(t) = 2^{n/2}\psi(2^n t - k).$$

The dilation parameter n will stretch or compress the basis functions on the t-axis. It corresponds to a translation on the ω-axis of its Fourier transform. The translation parameter k translates the basis function on the t-axis and will stretch or compress the Fourier transform on the ω-axis.

The basis functions $\{\psi_{n,k}\}_{k\in\mathbb{Z}}$ will generate a subspace W_n which is called the space of resolution n. The projection of f on this space is a representation of f at resolution level n.

Thus the representation of a signal in the wavelet basis should be of the form

$$f(t) = \sum_{n,k\in\mathbb{Z}} a_{nk}\psi_{nk}(t). \tag{5.1}$$

If we would succeed in making the wavelet basis orthonormal, then the wavelet coefficients a_{nk} are given by $a_{nk} = \langle \psi_{nk}, f\rangle$. The representation of f at resolution n is given by $f_n(t) = \sum_k a_{nk}\psi_{nk}(t)$.

Exactly the same intuitive reasoning can be applied in the case of non-periodic signals. This leads to the same kind of wavelet transform.

5.2 Bases and frames

Before we start working with bases, we will first reflect on the notion of basis and generating set for infinite dimensional (function) spaces.

A basis is a set that is generating and free (i.e. linearly independent). In an infinite dimensional Hilbert space, we should be more careful since we have to deal with infinite sums and we have to take convergence aspects into account.

$\{\varphi_n\}$ is called a *Schauder basis* for a Hilbert space H if for all $f \in H$ there is a unique sequence $\{c_n\}$ such that $f = \sum c_n\varphi_n$. The equality means convergence in the norm of the Hilbert space. The basis is called *unconditional* if the convergence is uniform, i.e., independent of the order of the summation.

The most comfortable situation to work in is when we have an *orthonormal basis*, where $\langle\varphi_n, \varphi_m\rangle = \delta_{n-m}$. We have for all $f \in H$ that there is a unique sequence $c = \{c_k\}$ such that $f = \sum_k c_k\varphi_k$ with $c_k = \langle f, \varphi_n\rangle$, and the Parseval equality holds $\|f\|_H^2 = \sum |\langle f, \varphi_n\rangle|^2 = \|c\|^2_{\ell_2(\mathbb{Z})}$.

A *Riesz basis* is not really an orthonormal basis but it is "equally easy" to work with since it is topologically the same. This is the meaning of the following definition. A basis $\{\varphi_n\}$ is called a *Riesz basis* if it is topologically isomorphic with an orthonormal basis. Thus there is a topological isomorphism[1] $T : H \to H$ such that $\varphi_n = Tu_n$ with $\{u_n\}$ an orthonormal basis. In other words, a Riesz basis is "topologically equivalent" with an orthonormal basis.

In a much more general situation, consider a set of *atoms* $\{\varphi_n\} \subset H$ which need not be a basis. This set is sometimes called a *dictionary* of atoms. For $f \in H$, the expansion $f = \sum c_k\varphi_k$, is called a *representation* of f (with respect to $\{\varphi_n\}$). The mapping $L : H \to \ell_2(\mathbb{Z}) : f \mapsto \{\langle f, \varphi_n\rangle\}$ is the *representation operator* or the *analysis operator*.

If the dictionary $\{\varphi_n\}$ is an orthonormal basis, then the Parseval equality can be written as $\|f\|_H^2 = \|Lf\|^2_{L(H)}$ where $L(H) = \{Lf : f \in H\} \subset \ell^2(\mathbb{Z})$ and L is the representation operator for that basis. This operator L is a unitary operator in the case of an orthonormal basis and we have $f = L^*Lf = \sum \langle f, \varphi_n\rangle \varphi_n$. The *synthesis operator* is given by $L^* : L(H) \to H : c = (c_k) \mapsto \sum c_k\varphi_k$. In this case $L^* = L^{-1}$.

The following theorem characterizes an orthonormal basis.

Theorem 5.2.1. *If $\{\varphi_n\}$ is an orthonormal sequence in the Hilbert space H, then the following are equivalent.*

1. *$\{\varphi_n\}$ is an orthonormal basis for H*
2. *if $\langle f, \varphi_n\rangle = 0$ for all n, then $f = 0$*
3. *the analysis operator L is injective*
4. *$\{\varphi_n\}$ is complete: the closure of* $\operatorname{span}\{\varphi_n\}$ *is equal to H*

[1] A topological isomorphism is a bijective map T such that T and T^{-1} are continuous.

5. *Parseval equality holds:* $\|f\|_H^2 = \sum |\langle f, \varphi_n \rangle|^2$

6. *the analysis operator* L *is unitary.*

Example 5.2.1. For the space of all periodic bandlimited functions, i.e., with spectrum in $|\omega| \leq \omega_m$, set $\omega_s = 2\omega_m$, then $\sqrt{T}\,\mathrm{sinc}[\omega_s(t - nT)]$ with $T = 2\pi/\omega_s$ is an orthonormal basis. This follows from the sampling theorem. It can certainly generate all the functions in the space. On the other hand a basis is always minimal, i.e., it is not possible to remove one element and still generate the whole space. In this example, the fact that the Nyquist sampling frequency is used implies that the generating set is minimal, and no element can be left out. This is what makes it a basis. Orthonormality is easily checked. $\diamond$

For a Riesz basis, the situation is slightly more general. It is given by $\varphi_n = Tu_n$ with T the topological isomorphism, thus a bounded operator with bounded inverse. Here biorthogonality appears in a natural way. Indeed, define $\tilde{\varphi}_n = T^{-*}u_n$ where $T^{-*} = (T^{-1})^* = (T^*)^{-1}$. Then

$$\delta_{n-m} = \langle u_n, u_m \rangle = \left\langle T^{-1}Tu_n, u_m \right\rangle = \left\langle Tu_n, T^{-*}u_m \right\rangle = \langle \varphi_n, \tilde{\varphi}_m \rangle .$$

Thus $\{\varphi_n\}$ and $\{\tilde{\varphi}_n\}$ form biorthogonal sets. We have because $\{u_n\}$ is an orthonormal basis and T is linear

$$f = T(T^{-1}f) = T \sum \left\langle u_n, T^{-1}f \right\rangle u_n = \sum \left\langle T^{-*}u_n, f \right\rangle Tu_n = \sum \langle \tilde{\varphi}_n, f \rangle \varphi_n = L^*\tilde{L}f.$$

We have set

$$\tilde{L}f = \{\langle \tilde{\varphi}_n, f \rangle\}, \quad \text{and} \quad L^*c = \sum c_n \varphi_n.$$

Similarly one can derive that

$$f = T^{-*}(T^*f) = \sum \langle \varphi_n, f \rangle \tilde{\varphi}_n = \tilde{L}^*Lf$$

with

$$Lf = \{\langle \varphi_n, f \rangle\}, \quad \text{and} \quad \tilde{L}^*c = \sum c_n \tilde{\varphi}_n.$$

So that, because $\varphi_n = Tu_n = TT^*\tilde{\varphi}_n$, and therefore $L^* = (TT^*)^{-1}\tilde{L}^*$, we have

$$\tilde{L}^*L = L^*\tilde{L} = (TT^*)^{-1}L^*L = I_H$$

and thus $\tilde{L}^* = (TT^*)L^*$ gives the reconstruction of f from Lf: for all $f \in H$: $f = [(TT^*)L^*](Lf)$. Thus the analyis operator L is related to the reconstruction or synthesis operator $\tilde{L}^* = (TT^*)L^* = (L^*L)L^*$.

Now we come to the definition of a *frame*. This is a set $\{\varphi_n\}$ that is somewhat looser than a Riesz basis. The Riesz basis was still a basis and therefore it contained independent elements. This need not be true anymore for a frame. It is still a complete system for H, but the representation for a frame can be redundant. When a frame becomes minimal, i.e. if no more redundant elements can be removed, then it becomes a Riesz basis.

The set $\{\varphi_n\}$ is a *frame* if it is complete and if there exist constants $A, B > 0$ such that

$$\forall f \in H, \quad A\|f\|^2 \leq \sum |\langle \varphi_n, f \rangle|^2 \leq B\|f\|^2. \tag{5.2}$$

The constants A and B are called the *frame bounds* or Riesz bounds.

We note that a frame is *complete* in H, i.e., the closure of span$\{\varphi_n\}$ is equal to H, but it need not be a Riesz basis because it may contain redundant elements. A Riesz basis is always a frame though.

$$\text{orthonormal basis} \Rightarrow \text{Riesz basis} \Rightarrow \text{frame} \Rightarrow \text{complete set.}$$

For a given frame $\{\varphi_n\}$, we define the *frame operator* S as $Sf = \sum \langle \varphi_n, f\rangle \varphi_n$. By definition, the frame operator is a bounded operator with bounded inverse. In fact $\|S\| = B$ and $\|S^{-1}\| = A^{-1}$ give the best possible frame bounds.

The frame is called *tight* if $A = B$.

Example 5.2.2. Let $\{u_k\}$ be an orthonormal basis, then $\{u_1, u_1, u_2, u_2, u_3, u_3, \ldots\}$ is a tight frame ($A = B = 2$). ◇

An *exact frame* (i.e., a minimal frame where removing one element makes it incomplete) is a Riesz basis.

For a frame there is always a dual frame (which is not biorthogonal). Indeed, the frame operator S is a topological isomorhism, and thus, we can associate with a frame $\{\varphi_n\}$ the *dual frame* $\{\tilde{\varphi}_n = S^{-1}\varphi_n\}$. The frame bounds for $\{\tilde{\varphi}_n\}$ are B^{-1} and A^{-1}.

The frame operator can also be factored. Define the analysis and synthesis operators

$$Lf = \{\langle \varphi_n, f\rangle\} \quad \text{and} \quad L^*c = \sum c_n \varphi_n.$$

Then $Sf = \sum \langle \varphi_n, f\rangle \varphi_n = L^*Lf$. Thus $S = L^*L$. The operator $R = LL^*$ is called the *correlation operator* of the frame. It is positive semi-definite and self-adjoint and it maps $L(H)$ bijectively onto itself. Its matrix representation is $[\langle \varphi_i, \varphi_j\rangle]_{i,j}$. It can be shown that the analysis operators for primal and dual frame are related by $\tilde{L} = RL$.

The following characterizes when a frame is a Riesz basis.

Theorem 5.2.2. *Given a frame $\{\varphi_n\}$ of a Hilbert space and the corresponding operators as defined above then the following are equivalent*

1. *$\{\varphi_n\}$ is a Riesz basis*
2. *it is an exact frame*
3. *L is onto $\ell^2(\mathbb{Z})$, i.e. $\{\varphi_n\}$ is a Riesz-Fischer sequence*[2]
4. *L^* is one to one*
5. *R is a topological isomorphism on $\ell^2(\mathbb{Z})$*
6. *$R > 0$ (positive definite, hence invertible)*

[2] A Riesz-Fischer sequence means that for all $c \in \ell^2(\mathbb{Z})$ there is some $f \in H$ such that $c = \{\langle \varphi_n, f\rangle\}$, i.e., such that $c = Lf$.

To stress once more the meaning of the frame condition (5.2) we note the following. Two elements are independent if the first is not a multiple of the other, i.e., they are at a positive angle. The basis has independent elements if there is a positive angle between an element and any subspace spanned by other elements in the basis. However, in the infinite dimensional case, the angle between two sequences of basis elements can be positive but tend to zero. In that case the basis is called unstable. A Riesz basis is a stable basis where the latter does not happen. This can be expressed by the frame condition (5.2). Obviously, if the basis is orthogonal, then the angle is certainly not zero and this will always be a stable basis. The topological isomorphism can change the right angles and the lengths of the orthonormal basis but can not shrink them to zero (because it has a bounded inverse).

The reconstruction of f from its frame representation Lf is not obvious because the representation is redundant so that there exist precise relations between the numbers in Lf. The slightest perturbation to these data will make this the representation a sequence of numbers that is not the representation of any function for this frame. If the data are exact, then the function could be recovered from its representation by an iterative scheme. If the representation is not exact, then we could in principle recover several possible functions from several subsets. As an estimate of the original function, one can take an average over the different reconstructions. For some applications (like noise reduction) this has even an advantageous smoothing effect.

5.3 Discrete versus continuous wavelet transform

The array of coefficients $\{a_{nk}\}_{n,k\in\mathbb{Z}}$ in (5.1) is called the discrete wavelet transform (DWT) of f. We have chosen the dilation and translation parameter to be discrete.

The signal however was considered to be continuous. As we mentioned before, for practical computations, the signal will be sampled and will therefore be discrete (like in the example of the introduction). Thus $f \in \ell^2(\mathbb{Z})$ (in the non-periodic case) or f is just a finite vector of samples $f = [f_0, \ldots, f_{N-1}]$. Thus we work with complex vectors which are infinitely long or which have a finite length. In the above reasoning, the inner products become discrete sums over $\mathbb{Z}$ or over $\{0, \ldots, N-1\}$, but the basic idea is exactly the same. It is of course the latter situation which will occur in practical computations. For practical reasons (as in the discrete Fourier transform – DFT), one chooses for N most often a power of 2: $N = 2^K$. As a matter of fact, if the methods have to be implemented on a digital computer, then the integrals will be evaluated by quadrature formulas anyway, which means that they are replaced by sums.

Mathematically however, there is no reason why we should not go in the opposite direction and (like in Fourier analysis) let the dilation and translation parameters be continuous. Because of lack of time, and because we are mainly interested in practical implementations, we shall not discuss continuous wavelet transforms in depth. We briefly introduce the idea.

Calling the continuous parameters a and b, then one usually considers then atoms (they do not form a basis)

$$\psi_{a,b}(t) = \sqrt{|a|}\psi(a(t-b)), \quad a \neq 0, \quad a, b \in \mathbb{R}$$

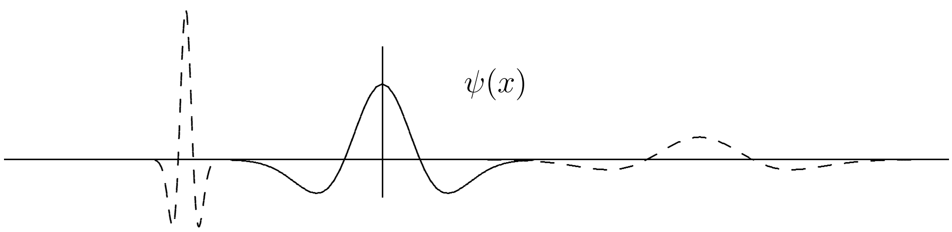
with $\psi \in L^2$.

Example 5.3.1. [Mexican hat] Consider

$$\psi(t) = (t^2 - 1)e^{-\frac{1}{2}t^2}.$$

Note that $\psi(t)$ is the second derivative of the function $\exp(-\frac{1}{2}t^2)$. Thus its Fourier transform is $\Psi(\omega) = (i\omega)^2 \mathcal{F}(e^{-t^2/2}) = -\omega^2 \exp(-\frac{1}{2}\omega^2)$. The function $\psi(t)$ and two of its dilations are plotted in Figure 5.1. One can see three of the basis functions $\psi_{a,b}(t)$. Note that neither $\psi(t)$

Figure 5.1: The mexican hat and two of its dilations



nor $\Psi(\omega)$ has a compact support, but they have a finite width and decay rapidly outside a finite interval. From left to right, b is increasing (shift more and more to the right) and $|a|$ is increasing (stretch more and more). $\diamond$

With continuous parameters, the continuous wavelet transform is a function of two real variables a and b:

$$\mathcal{W}_\psi f = F(a,b) = \frac{1}{\sqrt{2\pi C_\psi}} \int_{-\infty}^{\infty} f(t)\overline{\psi_{a,b}(t)}dt = \frac{1}{\sqrt{C_\psi}} \langle \psi_{a,b}, f \rangle, \quad f \in L^2(\mathbb{R}),$$

where it is supposed that

$$0 < C_\psi = \int_{-\infty}^{\infty} \frac{|\Psi(\omega)|^2}{|\omega|} d\omega < \infty$$

with Ψ the Fourier transform of ψ. This is called the *admissibility condition.* This C_ψ is some normalizing factor, similar to the factor 2π in the continuous Fourier transform. Note that $C_\psi < \infty$ implies $\Psi(0) = 0 = \frac{1}{\sqrt{2\pi}} \int_{\infty}^{\infty} \psi(t)dt$: if ψ is a wavelet, then its integral should be zero. Note that for $a = 2^n$ and $b = 2^{-n}k$, we are back in the discrete case.

Because b refers to 'time' and a refers to 'scale', thus (b, a) is a point in the time-scale space, the CWT is therefore sometimes caled a time-scale representation of the signal.

The CWT has the following properties

1. (linear): $\mathcal{W}_\psi(\alpha f + \beta g)(a,b) = \alpha(\mathcal{W}_\psi f)(a,b) + \beta(\mathcal{W}_\psi g)(a,b)$.

2. (time invariant): $\mathcal{W}_\psi(\mathcal{D}^u f)(a,b) = (\mathcal{W}_\psi f)(a, b-u)$.

3. (dilation): $(\mathcal{W}_\psi f_v)(a,b) = (\mathcal{W}_\psi f)(v^{-1}a, vb)$, $f_v(t) = \sqrt{|v|}f(vt)$.

The inverse transform is given by

$$\mathcal{W}_\psi^{-1} F = f(t) = \frac{1}{\sqrt{2\pi C_\psi}} \int_{-\infty}^{\infty} \left[\int_{-\infty}^{\infty} F(a,b)\psi_{a,b}(t)db \right] da$$

The Lebesgue space $L^2(\mathbb{R})$ can be decomposed into two Hardy spaces $L^2(\mathbb{R}) = H^2_-(\mathbb{R}) \oplus H^2_+(\mathbb{R})$, where

$$H^2_+(\mathbb{R}) = \{f \in L^2(\mathbb{R}) : \operatorname{supp} \mathcal{F}(f) \subset [0,\infty)\}, \quad H^2_-(\mathbb{R}) = \{f \in L^2(\mathbb{R}) : \operatorname{supp} \mathcal{F}(f) \subset (-\infty, 0]\}.$$

If we know that $F \in \mathcal{W}_\psi(H^2_+(\mathbb{R}))$, then we can restrict ourselves to a positive parameter $a > 0$, then the definition of C_ψ becomes

$$C_\psi = \int_0^\infty \frac{|\Psi(\omega)|^2}{\omega} d\omega < \infty$$

and the reconstruction formula is

$$f(t) = \frac{1}{\sqrt{2\pi C_\psi}} \int_0^\infty \left[\int_{-\infty}^{\infty} F(a,b)\psi_{a,b}(t)db \right] da.$$

Example 5.3.2. [**Morlet wavelet**] The Morlet wavelet is a modulated Gaussian: $\psi(t) = e^{i\alpha t}e^{-t^2/2}$. The parameter α is chosen appropriately. This wavelet however does not satisfy the condition $\hat{\psi}(0) = 0$. However it is satisfied up to a small error. For $\alpha \geq 5.5$, the error is numerically negligible. $\diamond$

Note that in both of the previous situations (whether the wavelet transform was discrete or not), we discuss a continuous signal f (periodic or not). For practical implementation, also the signal shall be discretized as we shall discuss later.

5.4 Definition of a multiresolution analysis

Let us now give the mathematical definition of a multiresolution analysis (MRA). We give it for the space $L^2 = L^2(\mathbb{R})$ of analog signals, but the same holds true for periodic signals or for digital signals with only small modifications.

Definition 5.4.1 (Multiresolution). A multiresolution analysis of L^2 is a nested sequence of subspaces $\cdots V_{-2} \subset V_{-1} \subset V_0 \subset V_1 \subset V_2 \subset \cdots$ such that[3]

1. $\bigvee_{n\in\mathbb{Z}} V_n = L^2$ and $\bigcap_{n\in\mathbb{Z}} V_n = \{0\}$.

2. (scale invariance): $f(t) \in V_j \Leftrightarrow f(2t) \in V_{j+1}$, $j \in \mathbb{Z}$.

[3]The notation $\bigvee_{n\in\mathbb{Z}} V_n$ means the closure of $V = \bigcup_{n\in\mathbb{Z}} V_n$ in the norm of L^2. Thus it adds to V all the limits of sequences of functions from V which converge in L^2. The statement $\bigvee_{n\in\mathbb{Z}} V_n = L^2$ is equivalent to saying that V is dense in L^2: any function from L^2 can be approximated arbitrary close (in L^2-norm) by elements from V.

3. (shift invariance): $f(t) \in V_0 \Leftrightarrow f(t-k) \in V_0$, $k \in \mathbb{Z}$.

4. (shift invariant basis): $\{\varphi(t-k)\}_{k\in\mathbb{Z}}$ forms a Riesz basis for V_0.

Example 5.4.1. [Haar wavelets] The example that we have seen in the introduction defines a MRA. There we have seen it for a finite interval, but we can easily generalize this for the whole real axis. Let us define the box functions χ_{nk} which are given by the characteristic functions for the intervals $I_{nk} = [2^{-n}k, 2^{-n}(k+1)]$. Thus $\chi_{nk}(t) = 1$ for $t \in I_{nk}$ and $\chi_{nk}(t) = 0$ for $t \notin I_{nk}$. If we define $V_n = \text{span}\{\chi_{nk} : k \in \mathbb{Z}\}$, then we have a MRA. In the introduction we have used the original basis χ_{0k} with coordinates that were nonzero only for $k = 1, \ldots, 8$. The fundamental function (called scaling function) $\varphi(t)$ is given by χ_{00} and $\chi_{0k}(t) = \varphi(t-k)$. $\diamond$

5.5 The scaling function or father function

The function φ in the definition of a MRA is called a *scaling function* or *father function.* Denoting its shifted versions by $\varphi_{0k}(t) = \varphi(t-k)$, then these should be a Riesz basis for V_0, so that any $f \in V_0$ can be written as

$$f(t) = \sum_k a_k \varphi_{0,k}(t), \quad (a_k) \in \ell^2.$$

Since also $\varphi(t) \in V_0$, and by the scale invariance property also $\varphi(t/2) \in V_0$, there should exist $(c_k) \in \ell^2$ such that

$$\varphi(\frac{t}{2}) = \sum_k c_k \varphi(t-k), \quad (c_k) \in \ell^2, \quad t \in \mathbb{R}.$$

Thus

$$\varphi(t) = \sum_k c_k \varphi(2t-k), \quad k \in \mathbb{Z}, \quad t \in \mathbb{R}.$$

This is called the *dilation equation* or *two-scale relation*

To avoid trivialities, we require $\int_{-\infty}^{\infty} \varphi(t)dt = \theta \neq 0$, i.e. $\Phi(0) = \theta/\sqrt{2\pi}$, then

$$2\int_{-\infty}^{\infty} \varphi(t)dt = \sum_n c_n \int_{-\infty}^{\infty} \varphi(2t-n)d(2t-n),$$

thus

$$\sum_n c_n = 2.$$

Suppose that we can solve the dilation equation for some choice of the coefficients c_k, then we may consider the functions

$$\boxed{\varphi_{nk}(t) = 2^{n/2}\varphi(2^n t - k), \quad n, k \in \mathbb{Z}.}$$

Note that $V_n = \text{span}\{\varphi_{nk} : k \in \mathbb{Z}\}$.

We observe also that the solution of the dilation equation is only defined up to a multiplicative constant. Thus, if $\varphi(t)$ is a solution, then $\theta\varphi(t)$ is also a solution for any constant $\theta \neq 0$. This is precisely the constant we have in $\int \varphi(t)dt = \theta \neq 0$. This θ will only be fixed if we impose a normalisation condition like for example $\|\varphi\| = 1$.

5.6 Solution of the dilation equation

We give some sample solutions of the dilation equation

$$\boxed{\varphi(t) = \sum_k c_k \varphi(2t-k), \quad \sum_k c_k = 2.}$$

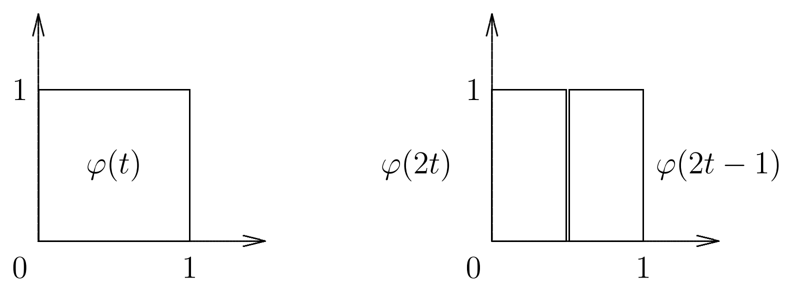
Example 5.6.1. Taking $c_0 = 2$ and all other $c_k = 0$, we see that $\varphi = \delta$ (the Dirac impulse) is a solution because it satisfies $\delta(t) = 2\delta(2t)$. This however is a pathological solution which does not have the usual properties for the solutions, so we shall not consider this to correspond to a wavelet. ◇

Example 5.6.2. [**Haar or box function**] For $c_0 = c_1 = 1$, the solution is a box function:

$$\varphi(t) = \chi_{[0,1[}(t) = \begin{cases} 1, & 0 \leq t < 1 \\ 0, & \text{otherwise} \end{cases}$$

The correctness can be checked on a picture (see Figure 5.2) ◇

Figure 5.2: The box function and the dilation equation



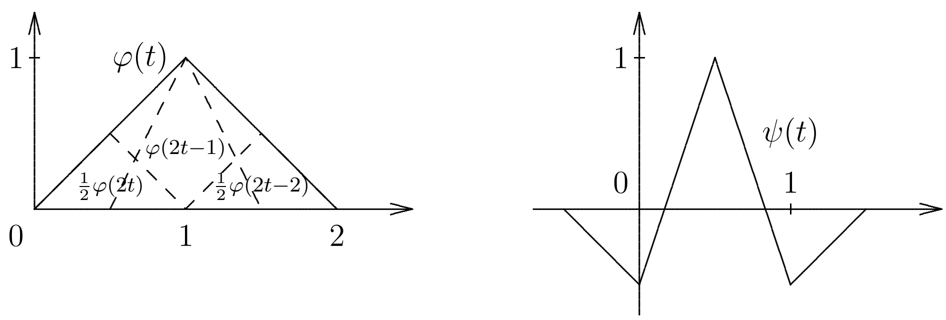
Example 5.6.3. [**piecewise linear spline or hat function**] For $c_1 = 1$ and $c_0 = c_2 = \frac{1}{2}$, the solution is the hat function:

$$\varphi(t) = \begin{cases} t, & 0 \leq t \leq 1 \\ 2-t, & 1 \leq t \leq 2 \\ 0, & \text{otherwise} \end{cases}$$

The correctness can again be checked graphically (see Figure 5.3 – the right figure is explained below). ◇

Such graphical verifications are only possible for simple examples. We need more systematic ways of solving the dilation equation. We give four methods.

Figure 5.3: The hat function and the dilation equation



5.6.1 Solution by iteration

One way to find $\varphi(t)$ is by iterating $\varphi^{[j]}(t) = \sum_k c_k \varphi^{[j-1]}(2t-k)$.
We remark however that $\varphi^{[j]}$ need not necessarily converge uniformly to the solution φ.

Example 5.6.4. Take for example with $\varphi_0 =$ the box function $\chi_{[0,1[}$.
For $c_0 = 2$, the box function gets taller and thinner, so it goes to the Dirac function.
For $c_0 = c_1 = 1$, the box remains invariant $\varphi^{[j]} = \varphi^{[0]}$, $j \geq 0$.
For $c_1 = 1$, $c_0 = c_2 = \frac{1}{2}$, the hat function appears as $j \to \infty$. ◇

Example 5.6.5. [**Quadratic spline**] Using a computer program with graphical possibilities, one can try the same with $c_0 = c_3 = \frac{1}{4}$, $c_1 = c_2 = \frac{3}{4}$. The solution is a quadratic spline.

$$\varphi(t) = \begin{cases} t^2, & 0 \leq t \leq 1 \\ -2t^2+6t-3, & 1 \leq t \leq 2 \quad \text{(quadratic spline)} \\ (3-t)^2, & 2 \leq t \leq 3 \\ 0 & \text{otherwise} \end{cases}$$

◇

Example 5.6.6. [**cubic B-spline**] Another example corresponds to the choice $c_0 = c_4 = \frac{1}{8}$, $c_1 = c_3 = \frac{1}{2}$, $c_2 = \frac{3}{4}$. The solution is the cubic B-spline. ◇

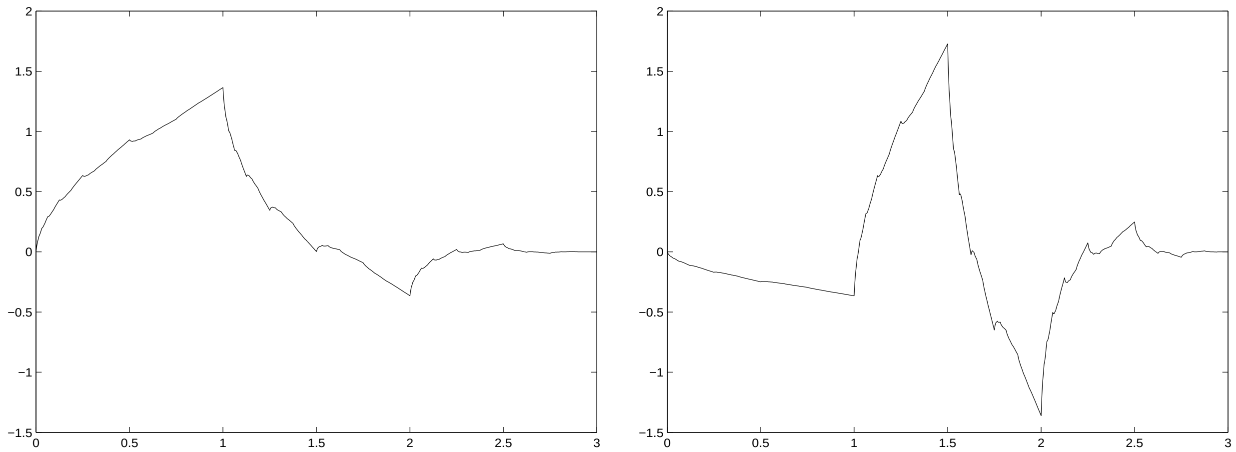
Example 5.6.7. [**D2 Daubechies**] An interesting example is obtained by choosing $c_0 = \frac{1}{4}(1+\sqrt{3})$, $c_1 = \frac{1}{4}(3+\sqrt{3})$, $c_2 = \frac{1}{4}(3-\sqrt{3})$, $c_3 = \frac{1}{4}(1-\sqrt{3})$. The solution is somewhat surprising. The corresponding wavelet is called D_2 (D for Daubechies and 2 because there are 2 vanishing moments – see later). The result is plotted in the first part of Figure 5.4. For the corresponding wavelet function ψ see below. ◇

As an application of this iteration method, we can immediately show the following property about the support[4] of the function φ.

Theorem 5.6.1. *If* $\varphi(t) = \sum_n c_n \varphi(2t-n)$ *with* $c_n = 0$ *for* $n < N_-$ *and* $n > N_+$*, then* $\text{supp}(\varphi) \subset [N_-, N_+]$.

[4]Recall that the support is the closure of the complement of the set where the function is zero.

Figure 5.4: The Daubechies D_2 scaling function and wavelet.



Proof. We use the construction by iteration and assume that the process converges. Let $\varphi^{[0]} = \chi_{[-\frac{1}{2},\frac{1}{2}[}$ and iterate

$$\varphi^{[j]}(t) = \sum_n c_n \varphi^{[j-1]}(2t-n)$$

Denote $\operatorname{supp}(\varphi^{[j]}) = [N_-^{[j]}, N_+^{[j]}]$, then

$$N_-^{[j]} = \frac{1}{2}(N_-^{[j-1]} + N_-) \quad , \quad N_+^{[j]} = \frac{1}{2}(N_+^{[j-1]} + N_+)$$

with

$$N_-^{[0]} = -\frac{1}{2} \quad , \quad N_+^{[0]} = \frac{1}{2}.$$

An easy induction shows that

$$N_-^{[j]} = 2^{-j} N_-^{[0]} + \left(\frac{1}{2} + \frac{1}{2^2} + \cdots + \frac{1}{2^j}\right) N_-,$$

which converges for $j \to \infty$ to N_-. A similar argument shows that $\lim_{j\to\infty} N_+^{[j]} = N_+$. This proves the theorem. $\square$

The iteration process described here may not always converge smoothly.

Example 5.6.8. Take $c_0 = c_3 = 1$ and all other $c_k = 0$. The solution of the dilation equation is $\varphi(t) = \chi_{[0,3[}$, but the iteration process does not converge uniformly. Check that. $\diamond$

5.6.2 Solution by Fourier analysis

Here the dilation equation is transformed in the Fourier domain and solved there. Defining the Fourier transform

$$\Phi(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \varphi(t) e^{-it\omega} dt,$$

the dilation equation gives

$$\begin{aligned} \Phi(\omega) &= \sum_n \frac{c_n}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \varphi(2t-n) e^{-it\omega} dt = \frac{C(\frac{\omega}{2})}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \varphi(\tau) e^{-i\tau\omega/2} d\tau \\ &= C(\tfrac{\omega}{2}) \Phi(\tfrac{\omega}{2}) \end{aligned}$$

where[5] $C(\omega) = \frac{1}{2} \sum_n c_n e^{-in\omega}$. Note that $C(0) = 1$. Thus the dilation equation in the Fourier domain reads

$$\Phi(2\omega) = C(\omega)\Phi(\omega).$$

Iterating the above result and using $\Phi(0) = 1/\sqrt{2\pi} \int_{-\infty}^{\infty} \varphi(t) dt = \theta/\sqrt{2\pi} \neq 0$, we find

$$\boxed{\Phi(\omega) = \frac{\theta}{\sqrt{2\pi}} \prod_{j=1}^{\infty} C(\omega/2^j).}$$

It can be shown rigorously that this infinite product makes indeed sense but we shall not do this here.

Example 5.6.9. $c_0 = 2$, then $C(\omega) = 1$, $\Phi(\omega) = \theta/\sqrt{2\pi}$ and this is indeed the Fourier transform of the Dirac function. ◇

Example 5.6.10. $c_0 = 1 = c_1$ (box function). The product of the C-functions ($C(\omega) = (1 + e^{-i\omega})/2$) is a geometric series.

$$C(\omega/2)C(\omega/4) = \frac{1}{4}(1 + e^{-i\omega/2})(1 + e^{-i\omega/4}) = \frac{1 - e^{-i\omega}}{4(1 - e^{-i\omega/4})}.$$

The product of N such functions is

$$\prod_{k=1}^{N} C(\omega/2^k) = \frac{1 - e^{-i\omega}}{2^N(1 - e^{-i\omega/2^N})}$$

which for $N \to \infty$ approaches

$$\frac{\sqrt{2\pi}}{\theta}\Phi(\omega) = \frac{1 - e^{-i\omega}}{i\omega} = \int_0^1 e^{-i\omega t} dt = \sqrt{2\pi}\mathcal{F}\chi_{[0,1[}$$

and this identifies Φ as θ times the Fourier transform of the box function. For $\theta = 1$ we get the box function $\chi_{[0,1[}$. Another choice of $\theta \neq 0$ gives the same function with another normalization. ◇

[5] Observe that $\mathsf{H}(\omega) = \sqrt{2}C(\omega) = \frac{1}{\sqrt{2}} \sum_k c_k e^{-ik\omega}$ is the Fourier transform of (h_k) with $h_k = \frac{1}{\sqrt{2}} c_k$.

The Fourier analysis approach now gives easily the following examples which you may check.

Example 5.6.11. The hat function comes from squaring the previous $C(\omega)$, hence squaring $\prod_1^\infty C(\omega/2^j)$. ◇

Example 5.6.12. The cubic spline comes from squaring again. ◇

Example 5.6.13. [**Shannon**] Suppose

$$C(\omega) = \begin{cases} 1, & |\omega| \leq \pi/2 \\ 0, & |\omega| > \pi/2 \end{cases} \quad \text{and} \quad \Phi(\omega) = \begin{cases} 1, & |\omega| \leq \pi \\ 0, & |\omega| > \pi \end{cases}$$

Then the dilation equation $\Phi(2\omega) = C(\omega)\Phi(\omega)$ is satisfied. In the time domain it becomes $\varphi(t) = \sum_k c_k \varphi(2t-k)$ with $c_0 = 1$, $c_{2k} = 0$ and $c_{2k+1} = (-1)^k \frac{2}{(2k+1)\pi}$. The solution is obviously

$$\varphi(t) = \sqrt{2\pi}\frac{\sin \pi t}{\pi t}.$$

This is related to the Shannon wavelet. ◇

5.6.3 Solution by recursion

Suppose $\varphi(t)$ is known at integer values $t = k$. Then the dilation equation defines $\varphi(t)$ at half integers $t = k/2$. Repeating this process yields $\varphi(t)$ at all dyadic points $t = k/2^j$. This is a fast algorithm and it is often used in practice.

We know that $\varphi(k) = 0$ for $k \notin \{N_-, \ldots, N_+\}$. For example, if $N_- = 0$ and $N_+ = 5$ we get

$$\begin{bmatrix} \varphi(0) \\ \varphi(1) \\ \varphi(2) \\ \varphi(3) \\ \varphi(4) \\ \varphi(5) \end{bmatrix} = \begin{bmatrix} c_0 & & & & & \\ c_2 & c_1 & c_0 & & & \\ c_4 & c_3 & c_2 & c_1 & c_0 & \\ & c_5 & c_4 & c_3 & c_2 & c_1 \\ & & & c_5 & c_4 & c_3 \\ & & & & & c_5 \end{bmatrix} \begin{bmatrix} \varphi(0) \\ \varphi(1) \\ \varphi(2) \\ \varphi(3) \\ \varphi(4) \\ \varphi(5) \end{bmatrix}.$$

Obviously, this matrix should have an eigenvalue 1. The function values $\varphi(k)$ are the components of the corresponding eigenvector. It also follows from $\varphi(N_-) = c_{N_-}\varphi(N_-)$ and $\varphi(N_+) = c_{N_+}\varphi(N_+)$ that $\varphi(N_-) = \varphi(N_+) = 0$ (unless $c_{N_-} = 1$ or $c_{N_+} = 1$).

Example 5.6.14. For D_2, we can find starting values at $\varphi(1)$ and $\varphi(2)$ as follows. We know by Theorem 5.6.1 that $\operatorname{supp}\varphi(t) \subset [0,3]$. We also know that at the boundaries, $\varphi(0) = \varphi(3) = 0$, so that of all values $\varphi(k)$, $k \in \mathbb{Z}$, only $\varphi(1)$ and $\varphi(2)$ are nonzero. Hence, the dilation equation gives

$$\begin{array}{rcl} \varphi(1) & = & c_1\varphi(1) + c_0\varphi(2) \\ \varphi(2) & = & c_3\varphi(1) + c_2\varphi(2) \end{array} \equiv \begin{bmatrix} \varphi(1) \\ \varphi(2) \end{bmatrix} = C \begin{bmatrix} \varphi(1) \\ \varphi(2) \end{bmatrix}$$

Thus $[\varphi(1) \quad \varphi(2)]^t$ is an eigenvector for the matrix $C = \begin{bmatrix} c_1 & c_0 \\ c_3 & c_2 \end{bmatrix}$. Its eigenvalues are $\lambda = 1$ and $\lambda = \frac{1}{2}$. For $\lambda = 1$, the eigenvector is $\varphi(1) = cc_0$, $\varphi(2) = cc_3$. The constant c is chosen to normalize the vector. As we shall see later in Theorem 5.7.1, the values of $\varphi(k)$ should sum up to θ. Hence, then $c = \theta/(c_0 + c_3)$. This eventually gives the required function values to start with. The next values at 1/2 and 3/2 are given by

$$\begin{aligned} \varphi(\frac{1}{2}) &= c_0\varphi(1) \\ \varphi(\frac{3}{2}) &= c_2\varphi(1) + c_1\varphi(2) \end{aligned}$$

etcetera. ◇

5.6.4 Solution by the cascade algorithm

Suppose $f \in V_j \subset V_{j+1}$ is given by

$$f(t) = \sum_k s_{jk}\varphi(2^j t - k) = f(t) = \sum_k s_{j+1,k}\varphi(2^{j+1}t - k).$$

From the dilation equation, we get

$$\varphi(2^j t - k) = \sum_i c_i\varphi(2^{j+1}t - (2k + i)) = \sum_l c_{l-2k}\varphi(2^{j+1}t - l).$$

Hence

$$f(t) = \sum_l \sum_k s_{jk}c_{l-2k}\varphi(2^{j+1}t - l)$$

so that

$$s_{j+1,l} = \sum_k c_{l-2k}s_{jk}.$$

Next observe that if we start the iteration with $s_{0k} = \delta_k$, then $f(t) = \varphi(t)$. Because the support of $\varphi(2^j t - k)$ become infinitely narrow for $j \to \infty$, it means that for j sufficiently large, the function value $\varphi(k/2^j)$ is approximately given by s_{jk}.

5.7 Properties of the scaling function

The dilation equation (and hence its solution φ) is completely defined by the coefficients (c_k), thus by the function

$$C(\omega) = \frac{1}{2}\sum_k c_k e^{-ik\omega} \in L^2_{2\pi}.$$

Therefore properties of φ correspond to properties of the filter coefficients (c_k) or equivalently of the function C.

5.7.1 General properties

We have already normalized the coefficients c_k such that

$$\boxed{\sum_k c_k = 2 \quad \Leftrightarrow \quad C(0) = 1.}$$

Let us show some properties of the scaling function.

Theorem 5.7.1 (partition of unity). *We have*

$$\boxed{\sum_k \varphi(t-k) = \sum_k \varphi(k) = \theta.}$$

Proof. Consider $w_j(t) = \sum_k \varphi(2^{-j}(t-k))$, then we have to prove that $w_0(t) = \theta$. Because this function has period 1, we may restrict ourselves to $0 \leq t \leq 1$. Now

$$\begin{aligned} w_j(t) &= \sum_k \varphi(2^{-j}(t-k)) \\ &= \sum_k \sum_l c_l \varphi(2^{-(j-1)}(t-k)-l) \\ &= \sum_l c_l \sum_k \varphi(2^{-(j-1)}(t-k-2^{j-1}l)) \qquad (k+2^{j-1}l = n) \\ &= \sum_l c_l \sum_n \varphi(2^{-(j-1)}(t-n)) = 2w_{j-1}(t) = 2^j w_0(t). \end{aligned}$$

Hence

$$w_0(t) = \frac{w_j(t)}{2^j} = \lim_{j\to\infty} \frac{w_j(t)}{2^j} = \lim_{j\to\infty} \sum_l 2^{-j}\varphi(2^{-j}(t+l)), \qquad \frac{t+l}{2^j} \in [\frac{l}{2^j}, \frac{l+1}{2^j}].$$

This is a Riemann sum for the integral $w_0(t) = \int \varphi(s)ds = \theta$. □

Equivalent forms are

Theorem 5.7.2. *The following properties are equivalent*

1. *The partition of unity*

$$\boxed{\sum_k \varphi(t-k) = \sum_k \varphi(k) = \theta.}$$

2. *The following condition of the filter coefficients*

$$\boxed{\sum_n (-1)^n c_n = 0 \quad \mathit{or} \quad 1 = \sum_k c_{2k} = \sum_k c_{2k+1}.} \tag{5.3}$$

3. The following condition of $C(\omega)$

$$\boxed{C(\pi) = 0.} \tag{5.4}$$

Proof. 1. (1) $\Rightarrow$ (2): By the dilation equation,

$$\begin{aligned}
\theta = \sum_l \varphi(t-l) &= \sum_l \left(\sum_k c_{2k}\varphi(2t-2k-2l) + \sum_k c_{2k+1}\varphi(2t-2k-2l-1) \right) \\
&= \sum_j \left(\sum_k c_{2k} \right) \varphi(2t-2j) + \sum_j \left(\sum_k c_{2k+1} \right) \varphi(2t-2j-1) \\
&= \sum_j \alpha_j \varphi(2t-j), \quad \alpha_{2j} = \sum_k c_{2k}, \quad \alpha_{2j+1} = \sum_k c_{2k+1}.
\end{aligned}$$

Because we know by hypothesis that also $\sum_j \varphi(2t-j) = \theta$, it follows by the independence of the $\varphi(2t-j)$ that $\alpha_j = 1$ for all j.

2. (2) $\Rightarrow$ (1): Set $w(t) = \sum_k \varphi(t-k)$ then using the dilation equation we find

$$\begin{aligned}
w(t) &= \sum_k \sum_n c_n \varphi(2t-2k-n) \\
&= \sum_k \left[\sum_{n \text{ even}} c_n \varphi(2t-(2k+n)) + \sum_{n \text{ odd}} c_n(2t-(2k+n)) \right] \\
&= \sum_k \left[\sum_\ell c_{2\ell}\varphi(2t-2(k+\ell)) + \sum_\ell c_{2\ell+1}\varphi(2t-2(k+\ell)-1) \right] \\
&= \sum_j \left[\sum_\ell c_{2\ell}\varphi(2t-2j) + \sum_\ell c_{2\ell+1}\varphi(2x-2j-1) \right] \\
&= \sum_j \varphi(2t-2j)(\sum_\ell c_{2\ell}) + \sum_j \varphi(2t-2j-1)(\sum_\ell c_{2\ell+1}) \\
&= \sum_j \varphi(2t-j) = w(2t)
\end{aligned}$$

This means that w has to be constant and thus independent of t, so that it equals $\sum_k \varphi(k)$. Now integrate $\sum_k \varphi(t-k) = c$ over $[0,1]$ then

$$c = \sum_k \int_0^1 \varphi(t-k)dx = \sum_k \int_{t-k}^{t-k+1} \varphi(\tau)d\tau = \int_{\mathbb{R}} \varphi(\tau)d\tau = \theta.$$

Hence $c = \theta$.

3. (2) $\Leftrightarrow$ (3): Because

$$C(\omega) = \frac{1}{2}\sum_k c_k e^{-ik\omega}, \quad C(\pi) = \frac{1}{2}\sum_k c_k(-1)^k.$$

Because $\sum_k c_k = 2$, we see that (2) and (3) are equivalent.

□

5.7.2 Orthogonality

By definition, we know that the system $\{\varphi_{n0}(t) = \varphi(t-n)\}$ is a Riesz basis for V_0. What can be said if we assume that it is an orthogonal basis?

First we have the following property about a normalization of an orthogonal basis.

Lemma 5.7.3 (normalization). *If $\{\varphi_{0k}\}$ is an orthogonal basis, then*

$$\left|\int \varphi(s)ds\right|^2 = \int |\varphi(s)|^2 ds.$$

Proof. By the orthogonality we have

$$\begin{aligned}\left|\int \varphi(s)ds\right|^2 &= \int \overline{\varphi(s)}\left(\int \varphi(\tau)d\tau\right)ds = \int \overline{\varphi(s)}\left(\sum_k \varphi(s-k)\right)ds \\ &= \sum_k \int \overline{\varphi(s)}\varphi(s-k)ds = \int |\varphi(s)|^2 ds\end{aligned}$$

□

We may conclude that $\sqrt{2\pi}\|\varphi\|^2 = \theta^2$. Also it is immediately verified that $\|\varphi(t)\| = \|\varphi(t-k)\|$. Thus, if we set $\|\varphi\| = 1$, i.e. $\theta = (2\pi)^{1/4}$ then an orthogonal basis $\{\varphi_{0k}\}$ will be orthonormal.

It is also clear that if $\{\varphi_{0k}\}$ is an orthonormal basis for V_0, then $\{\varphi_{nk}\}$ is an orthonormal basis for V_n.

We now express the orthonormality of the $\{\varphi_{0k}\}$ in terms of the coefficients c_k and in terms of the filter $C(\omega)$. The next theorem says that (5.5) is equivalent with this orthogonality. Note that the proof does not make use of the dilation equation.

Theorem 5.7.4. *The system $\{\varphi_{0k}\}$ is orthonormal if and only if the Fourier transform Φ satisfies*

$$\boxed{\sum_{k=-\infty}^{\infty} |\Phi(\omega + 2k\pi)|^2 = \frac{1}{\sqrt{2\pi}}.} \tag{5.5}$$

Proof. Use the fact that $\varphi(t-k)$ forms an orthonormal basis in V_0, then

$$\begin{aligned}\frac{1}{2\pi}\int_0^{2\pi} e^{im\omega}d\omega = \delta_m &= \frac{1}{\sqrt{2\pi}}\int_{\mathbb{R}} \varphi(t)\overline{\varphi}(t-m)dt \\ &= \frac{1}{\sqrt{2\pi}}\int_{\mathbb{R}} e^{im\omega}|\Phi(\omega)|^2 d\omega \\ &= \frac{1}{\sqrt{2\pi}}\sum_{k=-\infty}^{\infty}\int_{k2\pi}^{(k+1)2\pi} e^{im\omega}|\Phi(\omega)|^2 d\omega \\ &= \frac{1}{\sqrt{2\pi}}\int_0^{2\pi} e^{im\omega}\left(\sum_{k=-\infty}^{\infty}|\Phi(\omega+2k\pi)|^2\right)d\omega.\end{aligned}$$

The second line is because the Fourier transform defines an isomorphism, the last line because the Fourier transform is continuous. This proves (5.5). □

This has the following consequence.

Corollary 5.7.5. *The function* $C(\omega) = \frac{1}{2}\sum_k c_k e^{-ik\omega}$ *satisfies*

$$\boxed{|C(\omega)|^2 + |C(\omega+\pi)|^2 = 1} \tag{5.6}$$

Proof. Recall $\Phi(2\omega) = C(\omega)\Phi(\omega)$ so that

$$\frac{1}{\sqrt{2\pi}} = \sum_k |\Phi(\omega + 2k\pi)|^2 = \sum_k |\Phi(2\omega + 2k\pi)|^2 = \sum_k |C(\omega + k\pi)|^2 |\Phi(\omega + k\pi)|^2.$$

Because C is 2π-periodic,

$$\begin{aligned} \frac{1}{\sqrt{2\pi}} &= \sum_{k \text{ even}} + \sum_{k \text{ odd}} \\ &= |C(\omega)|^2 \sum_k |\Phi(\omega + 2k\pi)|^2 + |C(\omega+\pi)|^2 \sum_k |\Phi(\omega + (2k+1)\pi)|^2 \\ &= |C(\omega)|^2 \sum_k |\Phi(\omega + 2k\pi)|^2 + |C(\omega+\pi)|^2 \sum_k |\Phi(\omega + \pi + 2k\pi)|^2. \end{aligned}$$

Hence (5.6) follows. □

Example 5.7.1. [hat function] Note that the hat function ($c_0 = c_2 = 1$, $c_1 = \frac{1}{2}$) does not satisfy this relation. It is not orthogonal to its integer translates (check it). ◇

Corollary 5.7.6. *In terms of the* c_k *(5.6) is transformed into*

$$\boxed{\sum_n c_{n-2k}\overline{c}_{n-2\ell} = 2\delta_{k-\ell}.} \tag{5.7}$$

Proof. Note that (5.6) means that

$$\frac{1}{4}\sum_{k,l} c_k \overline{c}_l e^{-i(k-l)\omega} + \frac{1}{4}\sum_{k,l} c_k \overline{c}_l (-1)^{k-l} e^{-i(k-l)\omega} = 1.$$

Hence the odd terms drop out and we get

$$\sum_{k-l \text{ even}} c_k \overline{c}_l e^{-i(k-l)\omega} = \sum_j \gamma_j e^{-2ij\omega} = 2$$

with

$$\gamma_j = \sum_{k-l=2j} c_k \overline{c}_l.$$

This has to be true for all ω, so that $\gamma_j = 2\delta_j$. Thus $\sum_n c_n \overline{c}_{n-2j} = 2\delta_j$. This is of course equivalent with the statement to be proved. The complex conjugate drops out if the c_k are real. □

The property of the previous corollary:

$$\sum_n c_{n-2k}\overline{c}_n = 2\delta_k$$

is called *double shift orthogonality* for the coefficients c_k.

To make the circle complete, we should show that double shift orthogonality implies the orthogonality of the φ_{0k}. However this is not true in general.

Lemma 5.7.7. *If the iteration scheme coverges uniformly and if the c_k satisfy the double shift orthogonality then the system $\{\varphi_{0k}\}$ is orthogonal.*

Proof. The proof goes by starting the iteration scheme with an orthogonal set: for example the box functions. Then it is proved that orthogonality is preserved from iteration step to iteration step. If the iteration scheme converges, then the resulting $\varphi(t-k)$ will be orthogonal. To prove the induction step we use again the dilation equation

$$\begin{aligned}
r^{[i+1]}_{m,n} &= \frac{1}{\sqrt{2\pi}}\int \overline{\varphi^{[i+1]}(t-m)}\varphi^{[i+1]}(t-n)dt \\
&= \frac{1}{\sqrt{2\pi}}\int \left(\sum_p \overline{c}_p\overline{\varphi^{[i]}(2t-2m-p)}\right)\left(\sum_q c_q\varphi^{[i]}(2t-2n-q)\right)dt \\
&= \frac{1}{\sqrt{2\pi}}\int \left(\sum_p \overline{c}_p\overline{\varphi^{[i]}(2t-2m-p)}\right)\left(\sum_j c_{j-2l}\varphi^{[i]}(2t-2m-j)\right)dt, \quad \left(\begin{array}{c} n=l+m \\ j=2l+q \end{array}\right) \\
&= \sum_p\sum_j \overline{c}_p c_{j-2l}\frac{1}{\sqrt{2\pi}}\int \overline{\varphi^{[i]}(2t-2m-p)}\varphi^{[i]}(2t-2m-j)dt \\
&= \frac{1}{2}\sum_p\sum_j \overline{c}_p c_{j-2l} r^{[i]}_{p,j} = \frac{1}{2}\sum_j \overline{c}_j c_{j-2l} = \delta_l = \delta_{m-n}
\end{aligned}$$

where we used the induction hypothesis $r^{[i]}_{p,j} = \delta_{p-j}$. □

We thus may summarize

Theorem 5.7.8. *If $\int \varphi(t) = (2\pi)^{1/4}$ then the following are equivalent:*

1. *The system $\{\varphi_{0n}\}$ is orthonormal.*
2. *The Fourier transform $\Phi(\omega)$ satisfies $\sum_{k=-\infty}^{\infty}|\Phi(\omega+2k\pi)|^2 = \frac{1}{\sqrt{2\pi}}$*

This implies the following equivalent conditions

1. *The function $C(\omega)$ satisfies $|C(\omega)|^2+|C(\omega+\pi)|^2 = 1$*
2. *The coefficients c_k are double shift orthogonal: $\sum_n c_{n-2k}\overline{c}_{n-2\ell} = 2\delta_{k-\ell}$*

If the iteration scheme converges, then the 4 conditions are equivalent.

Remark 5.7.1. Note that in the previous theorem, the last two conditions do not imply the first two in general. For example the choice $c_0 = c_3 = 1$, $c_1 = c_2 = 0$ defines a φ which is not orthogonal to its translates (it is the box function on the interval $[0, 3]$), yet these coefficients satisfy (5.7) and (5.3) as can be easily checked.

There is an interesting practical consequence to the double shift orthogonality condition.

Corollary 5.7.9. *Orthogonal scaling functions with a compact support must have an even number of non-zero coefficients.*

Proof. Suppose that the coefficients c_k are zero for $k \notin \{N_-, \ldots, N_+\}$ and that $N_+ - N_- = 2p > 0$. This is impossible because then $\sum_n \bar{c}_n c_{n-2k} = 2\delta_k$ implies for $k = p$ that $(c_{N_-})(c_{N_+}) = 0$. $\square$

5.8 The wavelet or mother function

We know that in multiresolution analysis

$$V_n \subset V_{n+1}.$$

Suppose W_n is the orthogonal complement of V_n in V_{n+1}:

$$V_{n+1} = V_n \oplus W_n.$$

Thus

$$V_0 \oplus \sum_{k=0}^{n} W_k = \bigoplus_{-\infty}^{n} W_k = V_{n+1} \quad \text{and} \quad \bigoplus_{-\infty}^{\infty} W_k = L^2.$$

Now consider the function ψ defined by

$$\boxed{\psi(t) = \sum_n d_n \varphi(2t - n) \in V_1, \quad d_n = (-1)^n \bar{c}_{1-n}.} \tag{5.8}$$

We shall explain in the next section where this definition comes from. In this section we first try to get a bit familiar with the function ψ. It can be proved that (see section 5.9 below) if the φ_{0k} form an orthonormal basis for V_0, then

$$\psi_{0k}(t) = \psi(t - k)$$

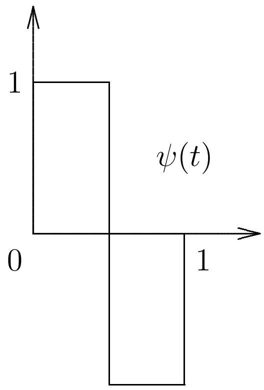
forms an orthonormal basis for W_0 and that more generally, the *wavelets*

$$\boxed{\psi_{nk}(t) = 2^{n/2}\psi(2^n t - k) \quad , \quad n, k \in \mathbb{Z}}$$

are such that $\{\psi_{nk} : k \in \mathbb{Z}\}$ forms an orthonormal basis for W_n.

The function $\psi(t)$ is called the *mother function* or the *wavelet* (function). The mother functions for the previous examples can now be considered. One can easily check the following examples (do it !).

Figure 5.5: The Haar wavelet.



Example 5.8.1. [Haar wavelet] For the box function ($c_0 = c_1 = 1$)

$$\psi(t) = \begin{cases} 1, & 0 \le t < 1/2 \\ -1, & 1/2 \le t < 1 \end{cases}$$

Figure 5.5 gives a plot of this wavelet. It is called the Haar wavelet. $\diamond$

Example 5.8.2. [piecewise linear wavelet] The hat function ($c_0 = c_2 = \frac{1}{2}, c_1 = 1$) leads to

$$\psi(t) = \begin{cases} -1/2 - t, & -1/2 \le t \le 0 \\ 3t - 1/2, & 0 \le t \le 1/2 \\ 5/2 - 3t, & 1/2 \le t \le 1 \\ t - 3/2, & 1 \le t \le 3/2 \end{cases}$$

The wavelet is plotted in Figure 5.3. $\diamond$

Note: The function ψ of example 5.8.2 is not really a wavelet according to our definition via MRA. The scaling function φ is not orthogonal to its integer translates (thus it is certainly not an orthogonal basis). The corresponding function ψ happens to be orthogonal to its integer translates. But the most important defect is that the $\psi(x-k)$ are not orthogonal to the $\varphi(t-l)$.

Example 5.8.3. For Daubechies D_2, $\psi(t-1)$ is plotted in Figure 5.4. $\diamond$

Theorem 5.8.1. *When*

$$c_n = 0 \quad \text{for} \quad n < N^- \quad \text{and} \quad n > N^+,$$

then

$$\operatorname{supp}(\psi) \subset \left[\frac{1}{2}(1 - N^+ + N^-), \frac{1}{2}(1 + N^+ - N^-)\right].$$

Proof. This follows from

$$\psi(t) = \sum_n (-1)^n \bar{c}_{1-n} \varphi(2t - n)$$

and the fact that $\text{supp}(\varphi) \subset [N^-, N^+]$. First note that $\text{supp}\,\varphi \subset [N^-, N^+]$ implies that $\text{supp}\,\varphi(2t-n)$ as a function of t is $[(N^- + n)/2, (N^+ + n)/2]$. On the other hand c_{1-n} is only nonzero for $n \in [1 - N^+, 1 - N^-]$. Therefore $\text{supp}\,\psi$ is exactly as stated. We leave the details to the reader. □

Let us develop now the analysis for ψ more exactly. It will give answers to questions such as: Where does the defining relation (5.8) of the ψ come from? Do the ψ_{nk} for $k \in \mathbb{Z}$ form indeed an orthonormal basis for the W_n? etc. We shall do this rather formally in section 5.9. This section can however be skipped. Section 5.10 gives a more informal approach.

5.9 Existence of the wavelet

Now let us prove the existence of the mother (wavelet) function.

We know that any $f \in V_0$ can be written as

$$f(t) = \sum_k a_k \varphi(t - k)$$

with $(a_k) \in \ell^2$ since the series should converge in $L^2(\mathbb{R})$. Taking Fourier transforms, this gives

$$F(\omega) = \sum_k a_k e^{ik\omega} \Phi(\omega) = A_f(\omega)\Phi(\omega), \quad A_f(\omega) = \sum_k a_k e^{-ik\omega}.$$

Clearly $A_f(\omega) \in L^2_{2\pi}$ and

$$\|A_f(\omega)\|^2_{L^2_{2\pi}} = \frac{1}{2\pi}\int_{-\pi}^{\pi} |A_f(\omega)|^2 d\omega = \|(a_k)\|^2_{\ell^2} < \infty.$$

Thus $f \in V_0 \Leftrightarrow F = A_f \Phi$ with $A_f \in L^2_{2\pi}$. Thus $f \in V_0$ is completely characterized by $A \in L^1_{2\pi}$. The proof of the following theorem can be skipped in first reading (a much simpler argument can be given in the case of discrete signals – see later). The basic idea can be summarized as follows: Just as $f \in V_0$ is characterized by $A \in L^2_{2\pi}$, any function of V_{-1} will be characterized by $A(2\omega)C(\omega)$ with $A \in L^2_{2\pi}$. The problem of constructing an orthonormal basis for W_{-1} is thus equivalent with the construction of an orthonormal basis for the orthogonal complement of functions $A(2\omega)C(\omega)$ in $L^2_{2\pi}$, which will be relatively easy.

Theorem 5.9.1. *There exists a function $\psi \in W_0$ such that $\{\psi(t-k)\}_{k\in\mathbb{Z}}$ forms an orthonormal basis for W_0.*

Proof. If V is a subspace of $L^2(\mathbb{R})$, then $\hat{V}$ will denote the subspace of $L^2(\mathbb{R})$ containing all the Fourier transforms of V.
By the Fourier isomorphism:

$$V_{-1} \oplus W_{-1} = V_0 \Leftrightarrow \hat{V}_{-1} \oplus \hat{W}_{-1} = \hat{V}_0.$$

We know that

$$\begin{aligned}\hat{V}_0 &= \{A(\omega)\Phi(\omega) : A \in L^2_{2\pi}\}\\ \hat{V}_{-1} &= \{A(2\omega)\Phi(2\omega) : A \in L^2_{2\pi}\}.\end{aligned}$$

By $\Phi(2\omega) = C(\omega)\Phi(\omega)$, we get

$$\hat{V}_{-1} = \{A(2\omega)C(\omega)\Phi(\omega) : A \in L^2_{2\pi}\}. \tag{5.9}$$

Define the operator (we show below that it is unitary)

$$S : \hat{V}_0 \to L^2_{2\pi} : A\Phi \mapsto A.$$

Note $S\hat{V}_0 = L^2_{2\pi}$. Instead of computing $\hat{W}_{-1}$ directly, we compute $S(\hat{W}_{-1})$ first, i.e. the orthogonal complement of $S(\hat{V}_{-1})$ in $L^2_{2\pi}$.

Lemma 5.9.2. *S is a unitary operator.*

Proof. It holds for any $f \in V_0$ that $F = A\Phi$ and that

$$\begin{aligned}\|F\|^2_{L^2(\mathbb{R})} &= \frac{1}{\sqrt{2\pi}}\int_{\mathbb{R}} |F(\omega)|^2 d\omega = \frac{1}{\sqrt{2\pi}}\int_{\mathbb{R}} |A_f(\omega)|^2|\Phi(\omega)|^2 d\omega\\ &= \sum_k \frac{1}{\sqrt{2\pi}}\int_{k2\pi}^{(k+1)2\pi} |A_f(\omega)|^2|\Phi(\omega)|^2 d\omega\\ &= \frac{1}{\sqrt{2\pi}}\int_0^{2\pi} |A_f(\omega)|^2(\sum_k |\Phi(\omega+2k\pi)|^2)d\omega = \frac{1}{2\pi}\int_0^{2\pi} |A_f(\omega)|^2 d\omega = \|A_f\|^2_{L^2_{2\pi}}.\end{aligned}$$

□

From (5.9) :

$$S(\hat{V}_{-1}) = \{A(2\omega)C(\omega) : A \in L^2_{2\pi}\}$$

Let $F \in L^2_{2\pi}$ be in the orthogonal complement of $S(\hat{V}_{-1})$, then

$$\int_0^{2\pi} A(2\omega)C(\omega)\overline{F(\omega)}d\omega = 0, \quad \forall A \in L^2_{2\pi}.$$

Thus

$$\int_0^{\pi} A(2\omega)[C(\omega)\overline{F(\omega)} + C(\omega+\pi)\overline{F(\omega+\pi)}]d\omega = 0 \quad , \quad \forall A \in L^2_{2\pi}$$

which implies

$$C(\omega)\overline{F(\omega)} + C(\omega+\pi)\overline{F(\omega+\pi)} = 0, \quad \forall \omega \in \mathbb{R}.$$

This means that in $\mathbb{C}^2$ the vector $\vec{h} = [C(\omega)\ C(\omega+\pi)]$ is orthogonal to the vector $\vec{f} = [F(\omega)\ F(\omega+\pi)]$:

$$\vec{h}\vec{f}^* = 0, \qquad \vec{h}\vec{h}^* = 1,$$

* means complex conjugate transpose. It is clear that

$$\begin{aligned} F(\omega) &= \overline{C(\omega+\pi)} \\ F(\omega+\pi) &= -\overline{C(\omega)} \end{aligned}$$

is a solution. More generally any solution is of the form

$$\begin{aligned} F(\omega) &= -\beta(\omega)\overline{C(\omega+\pi)} \\ F(\omega+\pi) &= \beta(\omega)\overline{C(\omega)}. \end{aligned}$$

For convenience, we choose $\beta(\omega) = \alpha(\omega)e^{-i\omega}$, because then

$$\begin{aligned} F(\omega) &= -\alpha(\omega)e^{-i\omega}\overline{C(\omega+\pi)} \\ F(\omega+\pi) &= \alpha(\omega)e^{i\omega}\overline{C(\omega)} \end{aligned}$$

implies that $\alpha(\omega)$ is π-periodic. Such a function can be written in terms of $e^{-i2k\omega}$. Thus we may choose

$$f_k(\omega) = -\sqrt{2}e^{-i\omega}\overline{C(\omega+\pi)}e^{-2ik\omega}, \quad k \in \mathbb{Z}$$

as a set of functions in $S(\hat{W}_{-1})$ that generates the whole space. These functions form an orthonormal basis because (we denote by $(\cdot,\cdot)$ the inner product for π-periodic functions) $(f_k, f_\ell) = (f_{k-\ell}, f_0)$ and using $|C(\omega+\pi)|^2 + |C(\omega)|^2 = 1$ and noting that $e^{-2ik\omega}$ is π-periodic, we get

$$\begin{aligned} (f_k, f_0) &= \frac{1}{\pi}\int_0^{2\pi} e^{-2ik\omega}|C(\omega+\pi)|^2 d\omega \\ &= \frac{1}{\pi}\int_0^{\pi} e^{-2ik\omega}[|C(\omega+\pi)|^2 + |C(\omega)|^2] d\omega \\ &= \frac{1}{\pi}\int_0^{\pi} e^{-2ik\omega} d\omega \\ &= \frac{1}{2\pi}\int_0^{2\pi} e^{-ik\eta} d\eta \\ &= \delta_k \end{aligned}$$

Taking the S^{-1} transform we find that

$$\Psi_{-1,k}(\omega) = -\sqrt{2}e^{-i\omega}\overline{C(\omega+\pi)}\Phi(\omega)e^{-2ki\omega} \quad , \quad k \in \mathbb{Z}$$

is an orthonormal basis for $\hat{W}_{-1}$.

Choosing a function $\psi \in W_0$ with Fourier transform Ψ satisfying

$$\Psi(2\omega) = D(\omega)\Phi(\omega), \quad D(\omega) = -e^{-i\omega}\overline{C(\omega+\pi)}, \tag{5.10}$$

we just found that $\sqrt{2}\Psi(2\omega)e^{-2ki\omega}$ forms an orthonormal basis for $\hat{W}_{-1}$. Taking the inverse Fourier transform reveals that $\psi_{-1,k}(t) = \frac{1}{\sqrt{2}}\psi(\frac{t}{2} - k)$ forms an orthonormal basis for W_{-1}. Moreover, after rescaling, we find that $\psi_{0k}(t) = \psi(t-k)$ is an orthonormal basis for W_0.

This concludes the proof of Theorem 5.9.1. □

Because $e^{i\omega}\Psi(2\omega)$ is the Fourier transform of $\frac{1}{2}\psi(\frac{t+1}{2})$ and

$$-\overline{C(\omega+\pi)}\Phi(\omega) = \frac{1}{2}\sum_k (-1)^{k+1}\bar{c}_k e^{ik\omega}\Phi(\omega)$$

we get, after taking inverse Fourier transforms of (5.10)

$$\frac{1}{2}\psi(\frac{t+1}{2}) = \frac{1}{2}\sum_k (-1)^{k+1}\bar{c}_k\varphi(t+k)$$

or

$$\psi(t) = \sum_k d_k\varphi(2t-k), \quad d_k = (-1)^k\bar{c}_{1-k}.$$

It is not difficult to accept that

$$\{\psi_{nk}(t) = 2^{n/2}\psi(2^n t - k): \quad k \in \mathbb{Z}\}$$

gives an orthonormal basis for W_n, and after taking the limit $L^2(\mathbb{R}) = \bigoplus_n W_n$, we find that

$$\{\psi_{nk}(t) = 2^{n/2}\psi(2^n t - k) : k, n \in \mathbb{Z}\}$$

forms an orthonormal wavelet basis for $L^2(\mathbb{R})$.

5.10 A more informal approach

Suppose we accept the following theorem

Theorem 5.10.1. *Suppose that $\{\varphi(t-n)\}$ forms an orthonormal basis for V_0, then there exists a function ψ such that $\{\psi(t-k)\}$ forms an orthonormal basis for $W_0 = V_1 \ominus V_0$.*

Since $\psi \in W_0 \subset V_1$, there must exist d_k such that

$$\psi(t) = \sum_k d_k\varphi(2t-k).$$

After Fourier transform, this is $\Psi(2\omega) = D(\omega)\Phi(\omega)$ where $D(\omega) = \frac{1}{2}\sum_k d_k e^{ik\omega}$.

Because $W_0 \perp V_0$, we have

$$\begin{aligned} 0 &= \int \overline{\psi(t)}\varphi(t-k)dt \\ &= \int \left(\sum_n \bar{d}_n \overline{\varphi(2t-n)}\right)\left(\sum_m c_m\varphi(2t-2k-m)\right)dt \\ &= \sum_n\sum_m \bar{d}_n c_m \int \overline{\varphi(2t-n)}\varphi(2t-2k-m)dt \end{aligned}$$

By orthogonality of the φ_{0k}, this implies

$$\sum_n \bar{d}_n c_{n-2k} = 0.$$

Similarly, by the orthogonality of the ψ_{0k}, we get

$$\sum_n \bar{d}_n d_{n-2k} = 2\delta_k.$$

A solution for these equations is $d_k = (-1)^k \bar{c}_{1-k}$, as one can easily verify.

It then follows from $\sum_k (-1)^k c_k = 0$, that also $\sum_k d_k = 0$ and hence, using $\int \varphi(t)dt = \theta = (2\pi)^{1/4}$, it follows that

$$\int \psi(t)dt = \sum_k d_k \int \varphi(2t-k)dt = 0.$$

5.11 Summary

We summarize our results and use this occasion to switch to a different normalization. Suppose we set

$$h_k = \frac{c_k}{\sqrt{2}} \quad \text{and} \quad g_k = \frac{d_k}{\sqrt{2}}$$

then $\sqrt{2}C(\omega) = \mathsf{H}(\omega) = H(e^{i\omega})$ and $\sqrt{2}D(\omega) = \mathsf{G}(\omega) = G(e^{i\omega})$ where

$$\mathsf{H}(\omega) = H(e^{i\omega}) = \sum_k h_k e^{-ik\omega} \quad \text{and} \quad \mathsf{G}(\omega) = G(e^{i\omega}) = \sum_k g_k e^{-ik\omega}.$$

We had $g_k = (-1)^k \bar{h}_{1-k}$. Thus

$$\begin{aligned} \Phi(2\omega) &= C(\omega)\Phi(\omega) = \frac{1}{\sqrt{2}}\mathsf{H}(\omega)\Phi(\omega) \\ \Psi(2\omega) &= D(\omega)\Phi(\omega) = \frac{1}{\sqrt{2}}\mathsf{G}(\omega)\Phi(\omega) \quad \text{and} \quad \mathsf{G}(\omega) = -e^{-i\omega}\overline{\mathsf{H}(\omega+\pi)}, \end{aligned}$$

so that

$$\mathsf{M}(\omega) = \begin{bmatrix} \mathsf{H}(\omega) & \mathsf{H}(\omega+\pi) \\ \mathsf{G}(\omega) & \mathsf{G}(\omega+\pi) \end{bmatrix} = \begin{bmatrix} \mathsf{H}(\omega) & -e^{-i\omega}\overline{\mathsf{G}(\omega)} \\ \mathsf{G}(\omega) & -e^{-i\omega}\overline{\mathsf{H}(\omega)} \end{bmatrix}. \tag{5.11}$$

This matrix M is the continuous analog of the modulation matrix of a 2-channel filter bank (see also in the next chapter).

Using orthogonality, it can be shown that $\mathsf{M}\mathsf{M}^* = 2I$: a relation which catches several properties implied. For example $|\mathsf{H}(\omega)|^2 + |\mathsf{G}(\omega)|^2 = 2$ and $\mathsf{G}(\omega)\overline{\mathsf{H}(\omega)} = \mathsf{H}(\omega)\overline{\mathsf{G}(\omega)} = 0$. The first relation is the continuous analog of (3.1) and it shows that H and G describe power complementary filters (PCF). The second relation shows that these filters are orthogonal, which is obvious because the filter H (see below) will give the V_{k-1} part of the signal and the filter G will give the W_{k-1} part and these spaces are orthogonal.

As an exercise one can prove in analogy with the scaling function that one has

$$\Psi(\omega) = D(\omega/2)\prod_{k=2}^{\infty} C(\omega/2^k),$$

and by the orthogonality of the $\psi(t-k)$ we get

$$\sum_k |\Psi(\omega + 2k\pi)|^2 = \frac{1}{\sqrt{2\pi}},$$

implying that

$$|\mathsf{G}(\omega)|^2 + |\mathsf{G}(\omega+\pi)|^2 = 2, \tag{5.12}$$

and by the orthogonality of φ_{nk} to ψ_{ml} one gets

$$\mathsf{H}(\omega)\mathsf{G}(-\omega) + \mathsf{H}(\omega+\pi)\mathsf{G}(-(\omega+\pi)) = 0. \tag{5.13}$$

In fact the last 2 equalities, together with (5.6) are equivalent with $[\mathsf{M}(\omega)][\mathsf{M}(\omega)]^* = 2I$.

The relation with the previous chapter on filter banks is now obvious. Writing $H(z) = \mathsf{H}(\omega)$ and $G(z) = \mathsf{G}(\omega)$ for $z = e^{i\omega}$, the relations are identical. Indeed the relations (5.6), (5.12), (5.13) become

$$\begin{aligned} H(z)H_*(z) + H(-z)H_*(-z) &= 2 \\ G(z)G_*(z) + G(-z)G_*(-z) &= 2 \\ H(z)G_*(z) + H(-z)G_*(-z) &= 0 \end{aligned}$$

showing that

$$M(z) = \begin{bmatrix} H(z) & H(-z) \\ G(z) & G(-z) \end{bmatrix}$$

satisfies $M(z)M_*(z) = 2I$, which corresponds to a paraunitary filter bank. We shall come back to this later.

5.12 Exercises

1. Prove that the hat function is the convolution of the box function with itself, hence that its Fourier transform is the square of the Fourier is the box function. The hat function is a piecewise linear polynomial, i.e., a spline of order 1. This is a way to construct polynomial B-splines. A polynomial B-spline of order p is a convolution of $p+1$ box functions. Therefore, if $\Phi(\omega)$ is the Fourier transform of the box function, then $\Phi(\omega)^2$ is the Fourier transform of the hat function and $\Phi(\omega)^4$ is the Fourier transform of the cubic B-spline, etc.

2. Prove that $C(\omega) = 2$ for $|\omega| \leq \pi/2$ and $\Phi(\omega) = 0$ for $|\omega| > \pi/$ is the Fourier transform of the sequence $c = (c_k)$ wit $c_0 = 1$, $c_{2k} = 0$, and $c_{2k+1} = (-1)^k \frac{2}{(2k+1)\pi}$.

3. Check that the hat function is not orthogonal to its integer translates.

4. Show that the piecewise linear wavelet of Example 5.8.2 is not orthogonal to its integer translates.

5. Fill in details of the proof of Theorem 5.8.1.

6. If $\psi(t) \in L^2(\mathbb{R})$ is the inverse Fourier transform of $\Psi(\omega)$, prove that the inverse Fourier transform of $\sqrt{2}\Psi(2\omega)e^{-i2k\omega}$ is equal to $\frac{1}{\sqrt{2}}\psi(\frac{t}{2} - k)$.

7. Let $\varphi \in L^2(\mathbb{R})$. Prove that the following two conditions are equivalent

   (a) $\{\varphi(\cdot - k) : k \in \mathbb{Z}\}$ satisfies the Riesz condition that for any $c = (c_k) \in \ell^2(\mathbb{Z})$, we have
   $$A\|c\|^2_{\ell^2(\mathbb{Z})} \leq \left\| \sum_{k\in\mathbb{Z}} c_k \varphi(\cdot - k) \right\|^2_{L^2(\mathbb{R})} \leq B\|c\|^2_{\ell^2(\mathbb{Z})}.$$

   (b) The Fourier transform Φ of φ satisfies
   $$A \leq \sum_{k\in\mathbb{Z}} |\Phi(\omega + 2k\pi)|^2 \leq B \quad \text{a.e.}$$

   Hint: The middle term in the first inequality is by Parseval $\|\mathsf{C}(\omega)\Phi(\omega)\|^2_{L^2(\mathbb{R})}$, $\mathsf{C}(\omega) = \mathcal{F}(c)$. Rewrite this as
   $$\frac{1}{\sqrt{2\pi}} \int_{-\pi}^{\pi} |\mathsf{C}(\omega)|^2 K(\omega) d\omega$$
   where $K(\omega) = \sum_k |\Phi(\omega + 2k\pi)|^2$. Hence the first inequality is equivalent with
   $$A \leq \frac{1}{\sqrt{2\pi}} \int_{-\pi}^{\pi} g(\omega) K(\omega) d\omega \leq B, \quad g(\omega) = \frac{|\mathsf{C}(\omega)|^2}{\|c\|_{\ell^2(\mathbb{Z})}}.$$
   Now prove that this is equivalent with the second inequality.

8. Consider $\mathbb{R}^2$ with the standard inner product: $\langle u, v \rangle = u_1v_1 + u_2v_2$. Let L^2 be the space of real valued functions for which the inner product is $\langle f, g \rangle = \sum_{k=1}^3 f(k)g(k)$. It is isomorphic to $\mathbb{R}^3$. Define the vectors $e = \{e_i\}_{i=1}^3$ in $\mathbb{R}^2$ as $e_1 = (1, 0)$, $e_2 = (0, 1)$, $e_3 = (a, b)$ with $a, b \in \mathbb{R}$. The analysis operator is $T : \mathbb{R}^2 \to \mathbb{R}^3 : u \mapsto v = (\langle u, e_i \rangle)_{i=1}^3$. What is the matrix representing T w.r.t. the standard basis? Prove that $G = T^*T = \sum_{k=1}^3 e_k e_k^*$. Show that $\|Tu\|^2_{\mathbb{R}^3} = \|u\|^2_{\mathbb{R}^2} + |\langle e_3, u \rangle_{\mathbb{R}^2}|^2$ and hence that $\|u\|^2_{\mathbb{R}^2} \leq \|Tu\|^2_{\mathbb{R}^3} \leq (1 + \|e_3\|^2_{\mathbb{R}^2})\|u\|^2_{\mathbb{R}^2}$. This proves that e is a frame. What are the frame bounds?

9. If $\{u_k\}$ and $\{v_k\}$ are two orthonormal bases, prove that $\{u_k, v_k\}$ forms a tight frame. What is the frame bound?

10. Prove that by adding the zero vector to a frame one gets a new frame with the same frame bounds.

11. Prove that a tight frame is an orthonoral basis if it consists of normalized vectors and if the frame bound is 1.

12. Consider in $\mathbb{R}^2$ the 3 vectors $e_1 = (0,1)$, $e_2 = (\sqrt{3}/2, -1/2)$ and $e_3 = (-\sqrt{3}/2, -1/2)$. They are all of length 1 and they are at angles $2\pi/3$. If for a vector $v \in \mathbb{R}^2$, one computes the coefficients $a_k = \langle e_k, v\rangle$, $k = 1,2,3$, (in the natural Euclidean inner product), How can v be reconstructed from the numbers (a_1, a_2, a_3)?

13. In the Shannon sampling theorem, show that the functions $\operatorname{sinc} \omega_m(t - nT)$ forms an orthogonal basis when $\omega_s = 2\omega_m$. When $\omega_s > 2\omega_m$, then these functions are redundant. They form a frame. Introduce the redundancy factor R as

$$R = \frac{\pi}{\omega_m T} = \frac{\omega_s}{2\omega_m} > 1, \quad \text{where} \quad T = \frac{2\pi}{\omega_s}.$$

Then it can be shown that by restricting the sinc functions to the bandwidth, then

$$f(t) = \frac{1}{R} \sum f(nT) \operatorname{sinc}[\frac{\pi}{RT}(t - nT)].$$

The functions $\operatorname{sinc}[\frac{\pi}{RT}(t - nT)]$ are redundant. Show that they form a tight frame.

14. If φ_1 is a solution of a dilation equation with coefficients c_1 and φ_2 is a solution of a dilation equation with coefficients c_2, Prove that $\varphi_1 * \varphi_2$ is a silution of a dilation equation with coefficients $c_1 * c_2$.

15. Work out example 5.6.8. Write a matlab program and observe the kind of convergence you get.

16. Check the wavelet functions ψ given in the examples of Section 5.8.

17. Work out the proof of Theorem 5.8.1.

Chapter 6

Wavelet transform and filter banks

To come to the wavelet transform, we should be able to decompose $f_n \in V_n = V_{n-1} \oplus W_{n-1}$ into its components $f_{n-1} \in V_{n-1}$ and $g_{n-1} \in W_{n-1}$. Thus we should be able to transform the scaling coefficients v_{nk} in the expansion of

$$f_n = \sum_k v_{nk}\varphi_{nk} \in V_n = V_{n-1} \oplus W_{n-1}$$

into the scaling coefficients $v_{n-1,k}$ and wavelet coefficients $w_{n-1,k}$ such that

$$f_{n-1} = \sum_k v_{n-1,k}\varphi_{n-1,k} \in V_{n-1} \quad \text{and} \quad g_{n-1} = \sum_k w_{n-1,k}\psi_{n-1,k} \in W_{n-1}.$$

This corresponds to a basis transformation in V_n: we change from the basis $\{\varphi_{nk}\}_k$ to the basis $\{\varphi_{n-1,k}\}_k \cup \{\psi_{n-1,k}\}_k$.

This will be equivalent with the analysis part of a (2-channel) filter bank. The inverse transform should do the opposite and this corresponds to the synthesis part of the filter bank.

We shall first describe this in the continuous case for signals in $L^2(\mathbb{R})$.

6.1 Wavelet expansion and filtering

Let f_n be in V_n. Because $V_n = V_{n-1} \oplus W_{n-1}$, we can decompose f_n uniquely as

$$f_n = f_{n-1} + g_{n-1} \quad \text{with} \quad f_{n-1} \in V_{n-1}, \quad g_{n-1} \in W_{n-1}.$$

If we repeat this, then

$$f_n = g_{n-1} + g_{n-2} + \cdots + g_{n-m} + f_{n-m}, \quad f_j \in V_j, \quad g_j \in W_j.$$

The integer m is large enough when f_{n-m} is sufficiently "blurred".

Now suppose that

$$\begin{aligned} f_j(t) &= \sum_k v_{jk}\varphi_{jk}(t), \quad v_j = (v_{jk}) \in \ell^2(\mathbb{Z}) \\ g_j(t) &= \sum_k w_{jk}\psi_{jk}(t), \quad w_j = (w_{jk}) \in \ell^2(\mathbb{Z}) \end{aligned}$$

The decomposition algorithm will decompose v_n into v_{n-1} and w_{n-1}, then v_{n-1} again into v_{n-2} and w_{n-2} etc. We have a recursive filter bank like in Figure 4.4. The decomposition is given in Figure 6.1. When we want to reconstruct the p_n, the algorithm should perform

Figure 6.1: The decomposition scheme

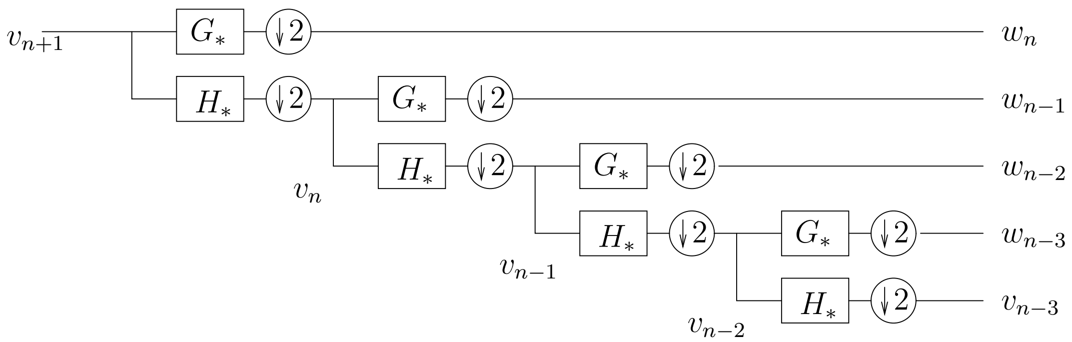
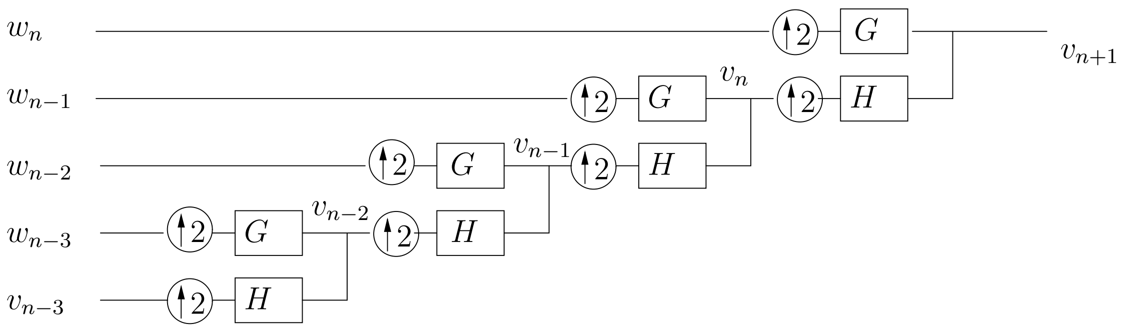


Figure 6.2: The reconstruction scheme



operations represented schematically in Figure 6.2. The purpose of this section is to find v_{n-1} and w_{n-1} from v_n (decomposition) and to recover v_n from v_{n-1} and w_{n-1} (reconstruction).

We recall that $h_k = c_k/\sqrt{2}$ and $g_k = d_k/\sqrt{2}$ so that

$$\Phi(2\omega) = \frac{1}{\sqrt{2}}\mathsf{H}(\omega)\Phi(\omega), \qquad \mathsf{H}(\omega) = \sum_k h_k e^{-ik\omega}$$
$$\Psi(2\omega) = \frac{1}{\sqrt{2}}\mathsf{G}(\omega)\Phi(\omega), \qquad \mathsf{G}(\omega) = \sum_k g_k e^{-ik\omega}$$

with $g_k = (-1)^k \overline{h}_{1-k}$ so that $\mathsf{G}(\omega) = -e^{-i\omega}\overline{\mathsf{H}(\omega+\pi)}$. Moreover $V_n = \operatorname{span}\{\varphi_{nk} : k \in \mathbb{Z}\}$ and $W_n = \operatorname{span}\{\psi_{nk} : k \in \mathbb{Z}\}$ where the

$$\varphi_{nk}(t) = 2^{n/2}\varphi(2^n t - k), \quad \psi_{nk}(t) = 2^{n/2}\psi(2^n t - k)$$

are orthonormal bases. The projection P_n on V_n and Q_n on W_n are given by

$$P_n f = \sum_k v_{nk}(f)\varphi_{nk}, \qquad v_{nk}(f) = \langle \varphi_{nk}, f\rangle$$
$$Q_n f = \sum_k w_{nk}(f)\psi_{nk}, \qquad w_{nk}(f) = \langle \psi_{nk}, f\rangle .$$

We want to relate v_{nk} and w_{nk} to $v_{n+1,k}$. We first prove

Lemma 6.1.1. *We have*

$$\varphi_{nk}(t) = \sum_l h_{l-2k}\varphi_{n+1,l}(t).$$

Proof. First note that

$$\varphi(\frac{t}{2}) = \sum_k c_k \varphi(t-k).$$

Then

$$\begin{aligned}
\varphi_{nk}(t) &= 2^{n/2}\varphi(2^n t - k) \\
&= 2^{n/2}\sum_i c_i \varphi(2^{n+1}t - 2k - i) \\
&= \frac{1}{\sqrt{2}}\sum_i c_i 2^{(n+1)/2}\varphi(2^{n+1}t - (2k+i)) \\
&= \frac{1}{\sqrt{2}}\sum_l c_{l-2k}\varphi_{n+1,l}(t) = \sum_l h_{l-2k}\varphi_{n+1,l}(t)
\end{aligned}$$

which proves the result. □

From their definitions, it thus follows that

$$v_{nk} = \sum_l \bar{h}_{l-2k} v_{n+1,l}$$

From our discussion of filter banks, it should be clear that this corresponds to applying a filter with transfer function $H_*(z)$ where $H(z) = \sum_k h_k z^k$, followed by downsampling (compare with the matrix (4.1)). If we denote this filter as $\mathcal{H}^*$, and the combination $\downarrow \mathcal{H}^*$ as $\hat{\mathcal{H}}^*$, then we have

$$v_n = (\downarrow \mathcal{H}^*)\ v_{n+1} = \hat{\mathcal{H}}^* v_{n+1}.$$

We do something similar for the w_{nk} :

Lemma 6.1.2. *We have*

$$\psi_{nk} = \sum_l g_{l-2k}\varphi_{n+1,l}.$$

Proof. This is along the same lines as the previous one

$$\begin{aligned}
\psi_{nk}(t) &= 2^{n/2}\psi(2^n t - k) \\
&= 2^{n/2}\sum_j d_j \varphi(2^{n+1}t - 2k - j) \\
&= \frac{1}{\sqrt{2}}\sum_j d_j 2^{(n+1)/2}\varphi(2^{n+1}t - (2k+j)) \\
&= \frac{1}{\sqrt{2}}\sum_l d_{l-2k}\varphi_{n+1,l}(t) = \sum_l g_{l-2k}\varphi_{n+1,l}(t).
\end{aligned}$$

□

Defining the filter $\mathcal{G}^*$ with transfer function $G_*(z)$ where $G(z) = \sum_k g_k z^k$, then it follows along the same lines as above that

$$w_n = (\downarrow \mathcal{G}^*)\ v_{n+1} = \hat{\mathcal{G}}^* v_{n+1}.$$

For the reconstruction we can easily prove that

Lemma 6.1.3. *The following relations hold*

$$\begin{aligned}
\langle \varphi_{n+1,k}, \varphi_{nl}\rangle_{L^2(\mathbb{R})} &= h_{k-2l} \\
\langle \varphi_{n+1,k}, \psi_{nl}\rangle_{L^2(\mathbb{R})} &= g_{k-2l}.
\end{aligned}$$

Proof. Also this one is trivial. For example from

$$\varphi_{nl}(t) = \sum_j h_{j-2l}\varphi_{n+1,j}(t)$$

we find that

$$\langle \varphi_{n+1,k}, \varphi_{nl}\rangle_{L^2(\mathbb{R})} = h_{k-2l}.$$

□

We can therefore express v_{n+1} in terms of v_n and w_n as follows.

$$\begin{aligned}
v_{n+1,k} &= \langle \varphi_{n+1,k}, f\rangle_{\ell^2(\mathbb{Z})} \\
&= \langle \varphi_{n+1,k}, P_{n+1}f\rangle_{\ell^2(\mathbb{Z})} \\
&= \langle \varphi_{n+1,k}, P_n f + Q_n f\rangle_{\ell^2(\mathbb{Z})} \\
&= \left\langle \varphi_{n+1,k}, \sum_l v_{nl}\varphi_{nl}\right\rangle_{\ell^2(\mathbb{Z})} + \left\langle \varphi_{n+1,l}, \sum_l w_{nl}\psi_{nl}\right\rangle_{\ell^2(\mathbb{Z})} \\
&= \sum_l h_{k-2l}v_{nl} + \sum_l g_{k-2l}w_{nl}.
\end{aligned}$$

Thus

$$v_{n+1} = (\mathcal{H}\uparrow)v_n + (\mathcal{G}\uparrow)w_n = \hat{\mathcal{H}}v_n + \hat{\mathcal{G}}w_n.$$

Here $\mathcal{H}$ and $\mathcal{G}$ are the filters with transfer function $H(z)$ and $G(z)$ respectively and $\hat{\mathcal{H}} = \mathcal{H}\uparrow$ and $\hat{\mathcal{G}} = \mathcal{G}\uparrow$.

Note: The notation with a superstar (which means adjoint) is justified as follows. If $\mathcal{F}$ is an operator (filter) on ℓ^2, then the *adjoint* $\mathcal{F}^* : \ell^2 \to \ell^2$ is defined by

$$\langle \mathcal{F}a, b\rangle_{\ell^2(\mathbb{Z})} = \langle a, \mathcal{F}^* b\rangle_{\ell^2(\mathbb{Z})}$$

The matrix representation of the adjoint operator is the Hermitian conjugate of the matrix representation of the operator. Recall also that the adjoint of $\downarrow$ is $\uparrow$, so that the adjoint of $\downarrow \mathcal{F}$ is $\mathcal{F}^* \uparrow$. Also, note that the adjoint of $\mathcal{F}$ with transfer function $F(z) = \sum_k f_k z^{-k}$ is the operator $\mathcal{F}^*$ with transfer function $F_*(z)$. A combination of these observations shows that $\hat{\mathcal{H}}^* = (\mathcal{H}\uparrow)^*$ is indeed the adjoint of $\hat{\mathcal{H}} = \mathcal{H}\downarrow$ and the same holds for $\hat{\mathcal{G}}$.

We can use the conditions on the c_k we have found before to see that $\hat{\mathcal{H}}^*\hat{\mathcal{H}} = \hat{\mathcal{G}}^*\hat{\mathcal{G}} = \mathcal{I}$ (the identity) and $\hat{\mathcal{H}}\hat{\mathcal{G}}^* = \hat{\mathcal{G}}\hat{\mathcal{H}}^* = \mathcal{O}$ (the zero operator). Somewhat more difficult is to show that $\hat{\mathcal{H}}^*\hat{\mathcal{H}} + \hat{\mathcal{G}}^*\hat{\mathcal{G}} = \mathcal{I}$. Thus with $\hat{\mathcal{K}}^* = [\hat{\mathcal{H}}^* \;\; \hat{\mathcal{G}}^*]$, we have $\hat{\mathcal{K}}\hat{\mathcal{K}}^* = \mathcal{I}$ and $\hat{\mathcal{K}}^*\hat{\mathcal{K}} = \mathcal{I}$.

6.2 Filter bank interpretation

We can interpret the previous decomposition and reconstruction as a 2-channel orthogonal filter bank.

We first recall that the matrix representation of the filter $\hat{\mathcal{H}}^*$ is given by

$$\mathbf{H}^* = \begin{bmatrix} \ddots & & & & & & & \\ \cdots & \bar{h}_{-1} & \bar{h}_0 & \bar{h}_1 & \bar{h}_2 & \cdots & & \\ & \cdots & \bar{h}_{-2} & \bar{h}_{-1} & \boxed{\bar{h}_0} & \bar{h}_1 & \cdots & \\ & & \cdots & \bar{h}_{-3} & \bar{h}_{-2} & \bar{h}_{-1} & \bar{h}_0 & \cdots \\ & & & & & & & \ddots \end{bmatrix}$$

(the framed element is the (0,0) entry). It corresponds to application of a filter with transfer function $H_*(z) = \sum_k \bar{h}_k z^k$ followed by decimation with a factor 2. Note that for $z = e^{i\omega}$, this is related to the function $C(\omega)$ of the multiresolution analysis by $H_*(z) = \sqrt{2}\,\overline{C(\omega)}$.

Similarly, the application of the filter $\hat{\mathcal{G}}$ corresponds to an application of a filter with transfer function $G_*(z) = \sum_k \bar{g}_k z^k$, followed by a decimation. Also $G_*(z) = \sqrt{2}\,\overline{D(\omega)}$ for $z = e^{i\omega}$.

The relation $g_k = (-1)^k \bar{h}_{1-k}$ immediately shows that the impulse responses $h = (h_k)$ and $g = (g_k)$ are orthogonal: $\langle h, g\rangle_{\ell^2(\mathbb{Z})} = 0$. More generally, it is easily checked that h is orthogonal to all the even translates of g: $\left\langle h, \mathcal{D}^{2k} g\right\rangle_{\ell^2(\mathbb{Z})} = 0$. This means that $\mathbf{H}^*\mathbf{G} = 0$. It also implies that we could as well have chosen $g_k = (-1)^k \bar{h}_{N-k}$ and still have orthogonality, as long as N is odd. The only essential thing is that the signs alternate and that the order of the filter coefficients is flipped and conjugated, then they can be shifted over any odd number of samples. We recognise this as the *alternating flip* relation. Note that a FIR filter with coefficients $\bar{h}_0, \bar{h}_1, \bar{h}_2, \ldots, \bar{h}_N$ is orthogonal to the FIR filter with coefficients $h_N, -h_{N-1}, h_{N-2}, \ldots, -h_0$ for N odd (see Corollary 5.7.9).

By (5.7), h and its even translates, are orthonormal: $\langle h, \mathcal{D}^{2k}h\rangle_{\ell^2(\mathbb{Z})} = \delta_k$. This is called double shift orthogonality and it corresponds to the fact that $\mathbf{HH}^* = \mathbf{I}$. Similarly $\mathbf{GG}^* = \mathbf{I}$.

In a completely analogous way, it is seen on the synthesis side that the filter $\mathcal{H}$ corresponds to an upsampling by a factor of 2 followed by application of the filter with transfer function $H(z) = \sum h_k z^{-k}$ while application of $\mathcal{G}$ corresponds to an upsampling followed by application of the filter with transfer function $G(z) = \sum g_k z^{-k}$.

This places the previous section in the context of a 2-channel filter bank. Writing down the modulation matrix (compare with (5.11))

$$M(z) = \begin{bmatrix} H(z) & H(-z) \\ G(z) & G(-z) \end{bmatrix}$$

and using the relation $G(z) = -z^{-1}H_*(-z)$ (compare with $D(\omega) = -e^{-i\omega}\overline{C(\omega+\pi)}$ of the multiresolution analysis) it is easily shown that $M_*(z)M(z) = 2I$: we have a paraunitary filter bank and we have perfect reconstruction.

The double shift orthogonality condition $\sum_n h_n \overline{h}_{n-2k} = \delta_k$ or equivalently $\mathbf{H}^*\mathbf{H} = \mathbf{I}$ was derived from the orthogonality of the $\varphi(x-k)$ and by the choice $g_n = (-1)^n \overline{h}_{1-n}$ it also implies the orthogonality of the $\psi(x-k)$ i.e., $\mathbf{G}^*\mathbf{G} = \mathbf{I}$. One may check that for all the simple examples we have seen whether this condition is satisfied. The box function and D_2 are the only ones which satisfy them.

The box function was the first known (orthogonal) scaling father function. Excluding the delta function, we find that none of the other examples (except D_2) satisfies the above condition and hence none of them is guaranteed to generate wavelets orthogonal to their translates.

6.3 Fast Wavelet Transform

We describe here the Fast Wavelet Transform (FWT) which is a method to compute the Discrete Wavelet transform (DWT) just like the Fast Fourier Transform (FFT) is a method to compute the Discrete Fourier Transform (DFT).

We suppose in the rest of this chapter that we work with real data and real filters. For this section we also assume that we work with orthogonal compactly supported wavelets, i.e. $h_k = 0$ for $k < 0$ and $k \geq 2N$ and the coefficients h_k are real.
We want to invert the relation

$$v_{n+1} = [\hat{\mathcal{H}}\ \hat{\mathcal{G}}] \begin{bmatrix} v_n \\ w_n \end{bmatrix}$$

which is done by

$$\begin{bmatrix} \hat{\mathcal{H}}^* \\ \hat{\mathcal{G}}^* \end{bmatrix} v_{n+1} = \begin{bmatrix} v_n \\ w_n \end{bmatrix}$$

In general the matrices corresponding to $\hat{\mathcal{H}}$ and $\hat{\mathcal{G}}$ are infinite dimensional. However, for practical reasons we work with discrete data vectors of finite length $M = 2^K$. The operators for analysis and synthesis as described above in terms of the filters $\hat{\mathcal{H}}$ and $\hat{\mathcal{G}}$ will transform vectors of length 2^K into vectors of the same length by multiplication with a $2^K \times 2^K$ matrix.

In DFT, this transform can be made fast (FFT) because the transformation is represented as a product of sparse elementary matrices. Moreover, the transformation matrix is orthogonal, so that its inverse is just equal to its transpose. This matrix is made orthogonal by choosing the unit vectors as basis in the "time" domain and the sines/cosines as basis in the frequency domain.

A similar observation holds if we want to generate a Fast Wavelet Transform (FWT) for the DWT.

For finite dimensional data, we shall have to truncate the infinite dimensional matrices. So for $M = 2^K$ we make $\mathbf{H}^*$, $\mathbf{G}^*$ of dimension $2^{K-1} \times 2^K$. E.g., for $K = 3$ and $N = 2$

$$\mathbf{H}^* = \begin{bmatrix} h_0 & h_1 & h_2 & h_3 & & & & \\ & & h_0 & h_1 & h_2 & h_3 & & \\ & & & & h_0 & h_1 & h_2 & h_3 \\ & & & & & & h_0 & h_1 \end{bmatrix}; \mathbf{G}^* = \begin{bmatrix} h_3 & -h_2 & h_1 & -h_0 & & & & \\ & & h_3 & -h_2 & h_1 & -h_0 & & \\ & & & & h_3 & -h_2 & h_1 & -h_0 \\ & & & & & & h_3 & -h_2 \end{bmatrix}$$

However this will cause some edge effects (orthogonality is lost). Therefore we suppose that the data are periodically extended, which amounts to reenter cyclically the data in H that were chopped off (a similar technique is used in DFT). So instead of the previous $\mathbf{H}^*$ and $\mathbf{G}^*$ for $K = 3$ and $N = 2$, we use the matrices

$$\mathbf{H}^* = \begin{bmatrix} h_0 & h_1 & h_2 & h_3 & & & & \\ & & h_0 & h_1 & h_2 & h_3 & & \\ & & & & h_0 & h_1 & h_2 & h_3 \\ h_2 & h_3 & & & & & h_0 & h_1 \end{bmatrix}; \mathbf{G}^* = \begin{bmatrix} h_3 & -h_2 & h_1 & -h_0 & & & & \\ & & h_3 & -h_2 & h_1 & -h_0 & & \\ & & & & h_3 & -h_2 & h_1 & -h_0 \\ h_1 & -h_0 & & & & & h_3 & -h_2 \end{bmatrix}$$

Now $\mathbf{G}$ and $\mathbf{H}$ are "orthogonal" again in the sense $\mathbf{HH}^* = \mathbf{I}$, $\mathbf{GG}^* = \mathbf{I}$ and $\mathbf{HG}^* = \mathbf{GH}^* = \mathbf{0}$. Thus, with $\mathbf{K}^* = [\mathbf{H}^* \;\; \mathbf{G}^*]$: $\mathbf{KK}^* = \mathbf{K}^*\mathbf{K} = \mathbf{I}$.
Suppose we have a data vector $\mathbf{v} = (v_l)_{l=0}^{2^K-1}$ of length 2^K. We can write it in terms of 2^K basis vectors φ_{0k}

$$\mathbf{v} = \sum_{k=0}^{2^K-1} v_{k0}\varphi_{0k}$$

with coefficient vector $\mathbf{v} = [v_{00}, \ldots, v_{2^K-1,0}]^t$, where

$$v_{k0} = \langle \mathbf{v}, \varphi_{0k} \rangle = \mathbf{v}^t \varphi_{0k} = \sum_{l=0}^{2^K-1} v_l \varphi_{0k}(l).$$

The expansion thus has to be understood as

$$\begin{bmatrix} v_0 \\ v_1 \\ \vdots \\ v_{2^K-1} \end{bmatrix} = \sum_{k=0}^{2^K-1} v_{k0} \begin{bmatrix} \varphi_{0k}(0) \\ \varphi_{0k}(1) \\ \vdots \\ \varphi_{0k}(2^K-1) \end{bmatrix}$$

E.g., consider $\mathbf{v} = [9\ 1\ 2\ 0]^t$, i.e., $K = 2$ and use the Haar wavelet, thus $h_0 = h_1 = \frac{1}{\sqrt{2}}$ and all other h_k are zero. Hence $\varphi = (\delta_l)_{l=0}^{2^K-1}$ is the first unit vector and φ_{0k} is the kth unit vector.

Note that $\varphi_{0l}(k) = \delta_{k-l}$ and thus $v_{k0} = \langle \mathbf{v}, \varphi_{0k} \rangle = v_k$.

$$\mathbf{v} = 9 \begin{bmatrix} \varphi_{00}(0) \\ \varphi_{00}(1) \\ \varphi_{00}(2) \\ \varphi_{00}(3) \end{bmatrix} + 1 \begin{bmatrix} \varphi_{01}(0) \\ \varphi_{01}(1) \\ \varphi_{01}(2) \\ \varphi_{01}(3) \end{bmatrix} + 2 \begin{bmatrix} \varphi_{02}(0) \\ \varphi_{02}(1) \\ \varphi_{02}(2) \\ \varphi_{02}(3) \end{bmatrix} + 0 \begin{bmatrix} \varphi_{03}(0) \\ \varphi_{03}(1) \\ \varphi_{03}(2) \\ \varphi_{03}(3) \end{bmatrix}.$$

We can also expand it in terms of

$$\varphi_{-1,k} \text{ and } \psi_{-1,k} \quad , \quad k = 0, 1, \ldots, 2^{K-1} - 1$$

with coefficients $\mathbf{v}_1 = (v_{k1})$ and $\mathbf{w}_1 = (w_{k1})$ given by

$$v_{k1} = \langle \mathbf{v}, \varphi_{-1,k} \rangle \quad , \quad w_{k1} = \langle \mathbf{v}, \psi_{-1,k} \rangle .$$

We could directly compute them by evaluating the inner products. However, by our previous analysis, we can also find them as

$$\begin{bmatrix} \mathbf{v}_1 \\ \mathbf{w}_1 \end{bmatrix} = \begin{bmatrix} \mathbf{H}_K^* \\ \mathbf{G}_K^* \end{bmatrix} \mathbf{v}_0.$$

For our previous example,

$$\mathbf{H}_2^* = \begin{bmatrix} h_0 & h_1 & & \\ & & h_0 & h_1 \end{bmatrix}; \quad \mathbf{G}_2^* = \begin{bmatrix} h_1 & -h_0 & & \\ & & h_1 & -h_0 \end{bmatrix}$$

so that

$$\begin{bmatrix} v_{01} \\ v_{11} \\ w_{01} \\ w_{11} \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 & & \\ & & 1 & 1 \\ 1 & -1 & & \\ & & 1 & -1 \end{bmatrix} \begin{bmatrix} 9 \\ 1 \\ 2 \\ 0 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 10 \\ 2 \\ 8 \\ 2 \end{bmatrix}$$

and we can check that this gives indeed the correct decomposition

$$\begin{aligned} \mathbf{v} &= \frac{10}{\sqrt{2}} \begin{bmatrix} \varphi_{-1,0}(0) \\ \varphi_{-1,0}(1) \\ \varphi_{-1,0}(2) \\ \varphi_{-1,0}(3) \end{bmatrix} + \frac{2}{\sqrt{2}} \begin{bmatrix} \varphi_{-1,1}(0) \\ \varphi_{-1,1}(1) \\ \varphi_{-1,1}(2) \\ \varphi_{-1,1}(3) \end{bmatrix} + \frac{8}{\sqrt{2}} \begin{bmatrix} \psi_{-1,0}(0) \\ \psi_{-1,0}(1) \\ \psi_{-1,0}(2) \\ \psi_{-1,0}(3) \end{bmatrix} + \frac{2}{\sqrt{2}} \begin{bmatrix} \psi_{-1,1}(0) \\ \psi_{-1,1}(1) \\ \psi_{-1,1}(2) \\ \psi_{-1,1}(3) \end{bmatrix} \\ &= \frac{10}{2} \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} + \frac{2}{2} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} + \frac{8}{2} \begin{bmatrix} 1 \\ -1 \\ 0 \\ 0 \end{bmatrix} + \frac{2}{2} \begin{bmatrix} 0 \\ 0 \\ 1 \\ -1 \end{bmatrix} = \begin{bmatrix} 9 \\ 1 \\ 2 \\ 0 \end{bmatrix}. \end{aligned}$$

The first and second term, are the components of v along $\varphi_{-1,0}$ and $\varphi_{-1,1}$. Together they form the part of v that is in V_{-1}. This can again be partitioned and written in terms of

$$\varphi_{-2,k} \text{ and } \psi_{-2,k} \quad , \quad k = 0, 1, \ldots, 2^{K-2} - 1$$

which in our example is

$$\varphi_{-2,0} \text{ and } \psi_{-2,0} \quad \rightarrow \quad \text{coefficients } v_{02} \text{ and } w_{02}.$$

This is for our example given by

$$\begin{bmatrix} v_{02} \\ w_{02} \end{bmatrix} = \begin{bmatrix} \mathbf{H}_1^* \\ \mathbf{G}_1^* \end{bmatrix} \begin{bmatrix} v_{01} \\ v_{11} \end{bmatrix} = \begin{bmatrix} h_0 & h_1 \\ h_1 & -h_0 \end{bmatrix} \begin{bmatrix} v_{01} \\ v_{11} \end{bmatrix}$$

thus, explicitly

$$\begin{bmatrix} v_{02} \\ w_{02} \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \frac{1}{\sqrt{2}} \begin{bmatrix} 10 \\ 2 \end{bmatrix} = \begin{bmatrix} 6 \\ 4 \end{bmatrix}$$

and indeed

$$\begin{bmatrix} 5 \\ 5 \\ 1 \\ 1 \end{bmatrix} = 6 \begin{bmatrix} \varphi_{-2,0}(0) \\ \varphi_{-2,0}(1) \\ \varphi_{-2,0}(2) \\ \varphi_{-2,0}(3) \end{bmatrix} + 4 \begin{bmatrix} \psi_{-2,0}(0) \\ \psi_{-2,0}(1) \\ \psi_{-2,0}(2) \\ \psi_{-2,0}(3) \end{bmatrix} = \frac{6}{2} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} + \frac{4}{2} \begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \end{bmatrix}$$

is equal to the sum of the first two terms in the previous decomposition. Thus we have written $\mathbf{v}$ as

$$\begin{aligned} \mathbf{v} &= 6\varphi_{-2,0} + 4\psi_{-2,0} + \frac{8}{\sqrt{2}}\psi_{-1,0} + \frac{2}{\sqrt{2}}\psi_{-1,1} \\ &= 6 \begin{bmatrix} 1/2 \\ 1/2 \\ 1/2 \\ 1/2 \end{bmatrix} + 4 \begin{bmatrix} 1/2 \\ 1/2 \\ -1/2 \\ -1/2 \end{bmatrix} + \frac{8}{\sqrt{2}} \begin{bmatrix} 1/\sqrt{2} \\ -1/\sqrt{2} \\ 0 \\ 0 \end{bmatrix} + \frac{2}{\sqrt{2}} \begin{bmatrix} 0 \\ 0 \\ 1/\sqrt{2} \\ -1/\sqrt{2} \end{bmatrix} \end{aligned}$$

Note that in this simple example we didn't need the wrap around of the H and G matrix. The two transformations together give the result $[v_{02}\ w_{02}\ w_{01}\ w_{11}]^t$ in terms of $[v_{00}\ v_{10}\ v_{20}\ v_{30}]^t$ as

$$\begin{aligned} \begin{bmatrix} v_{02} \\ w_{02} \\ w_{01} \\ w_{11} \end{bmatrix} &= \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 & & \\ 1 & -1 & & \\ & & \sqrt{2} & \\ & & & \sqrt{2} \end{bmatrix} \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 & & \\ & & 1 & 1 \\ 1 & -1 & & \\ & & 1 & -1 \end{bmatrix} \begin{bmatrix} v_{00} \\ v_{10} \\ v_{20} \\ v_{30} \end{bmatrix} \\ &= \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ \sqrt{2} & -\sqrt{2} & 0 & 0 \\ 0 & 0 & \sqrt{2} & -\sqrt{2} \end{bmatrix} \begin{bmatrix} 9 \\ 1 \\ 2 \\ 0 \end{bmatrix} = \begin{bmatrix} 6 \\ 4 \\ 8/\sqrt{2} \\ 2/\sqrt{2} \end{bmatrix} \end{aligned}$$

6.4 Wavelets by linear algebra

The filters (matrices) $\mathbf{G}^*$ and $\mathbf{H}^*$ are often intertwined to give a matrix like e.g.

$$\mathbf{T}^* = \begin{bmatrix} h_0 & h_1 & h_2 & h_3 & & & & & & & \\ h_3 & -h_2 & h_1 & -h_0 & & & & & & & \\ & & h_0 & h_1 & h_2 & h_3 & & & & & \\ & & h_3 & -h_2 & h_1 & -h_0 & & & & & \\ & & & & & \ddots & & & & & \\ & & & & & & h_0 & h_1 & h_2 & h_3 \\ & & & & & & h_3 & -h_2 & h_1 & -h_0 \\ h_2 & h_3 & & & & & & & h_0 & h_1 \\ h_1 & -h_0 & & & & & & & h_3 & -h_2 \end{bmatrix} \tag{6.1}$$

Assume for simplicity that the filter coefficients are real. If this matrix has to be orthogonal, (expressing $\mathbf{H}^*\mathbf{H} = \mathbf{I}$, $\mathbf{G}^*\mathbf{G} = \mathbf{I}$, and $\mathbf{H}^*\mathbf{G} = \mathbf{0}$) then, multiplying it with its transpose should give the identity.
This results for our example in only two independent relations

$$\begin{cases} h_0^2 + h_1^2 + h_2^2 + h_3^2 = 1 \\ h_2 h_0 + h_3 h_1 = 0 \end{cases}$$

If we require in addition the approximation to be of order[1] 2, then we require additionally

$$\begin{cases} h_0 - h_1 + h_2 - h_3 = 0 & (\mathsf{H}(\pi) = 0) \\ 0h_0 - h_1 + 2h_2 - 3h_3 = 0 & (\mathsf{H}'(\pi) = 0) \end{cases}$$

A solution of these 4 equations is given by

$$\begin{aligned} h_0 &= \frac{1+\sqrt{3}}{4\sqrt{2}}, & h_1 &= \frac{3+\sqrt{3}}{4\sqrt{2}} \\ h_2 &= \frac{3-\sqrt{3}}{4\sqrt{2}}, & h_3 &= \frac{1-\sqrt{3}}{4\sqrt{2}} \end{aligned}$$

which is Daubechies D_2.
When similarly introducing 6 coefficients $h_0, \ldots, h_5$, the orthogonality requirement gives 3 conditions, so that we can require the order to be 3 giving 3 more conditions.
A solution is given by D_3:

$$\begin{aligned} h_0 &= \frac{(1+\sqrt{10}+\sqrt{5+2\sqrt{10}})}{16\sqrt{2}}, & h_1 &= \frac{5+\sqrt{10}+3\sqrt{5+2\sqrt{10}}}{16\sqrt{2}} \\ h_2 &= \frac{10-2\sqrt{10}+2\sqrt{5+2\sqrt{10}}}{16\sqrt{2}}, & h_3 &= \frac{10-2\sqrt{10}+2\sqrt{5+2\sqrt{10}}}{16\sqrt{2}} \\ h_4 &= \frac{5+\sqrt{10}-3\sqrt{5+2\sqrt{10}}}{16\sqrt{2}}, & h_5 &= \frac{1+\sqrt{10}+\sqrt{5+2\sqrt{10}}}{16\sqrt{2}} \end{aligned}$$

One can check this as an exercise.

[1] This means that $\mathsf{H}(\pi) = \mathsf{H}'(\pi) = 0$. The partition of unity requires $\mathsf{H}(\pi) = 0$. The second condition is a smoothness condition. The general notion of order will be explained in Section 7.1

6.5 The wavelet crime

To move to practical computations, we shall assume that $f(t)$ is given by a number of samples. In practice there are a finite number, but for notational convenience, suppose that we have a countable number of values

$$f(t_k) = f(k\Delta t), \qquad k \in \mathbb{Z}.$$

Assume that $\Delta t = 2^{-N}$ so that $t_k = t_{Nk} = k\Delta t = k2^{-N}$. One could think that it is sufficient to feed the sample values to the recursive filter bank. However, this assumes that the function values are scaling coefficients (at level N), which is obviously not true. What we should feed into the filter bank are the coefficients v_{Nk} which are the scaling coefficients for the projection $f_N = P_N f$ of f on V_N:

$$f_N(t) = \sum_k v_{Nk}\varphi_{Nk}(t), \quad v_{Nk} = \langle \tilde{\varphi}_{Nk}, f \rangle .$$

(Note: we used here the $\tilde{\varphi}$ for a biorthogonal basis – see next section. For the moment one can think of $\tilde{\varphi}$ as being just φ.)
Replacing the coefficients v_{Nk} by samples of f is called the “wavelet crime” by G. Strang [21].

On the other hand, assuming that N is large and hence the φ_{Nk} and $\tilde{\varphi}_{Nk}$ have a verry narrow support (say width Δt centered at t_{Nk}), we can think of the v_{Nk} as (scaled) function values f_k. Indeed, since $\|\tilde{\varphi}_{Nk}\|^2 = \frac{1}{\sqrt{2\pi}}\int |\tilde{\varphi}_{Nk}(t)|^2 dt = 1 \approx \frac{\Delta t}{\sqrt{2\pi}}|\tilde{\varphi}_{Nk}(t_{Nk})|^2$, we should have $|\tilde{\varphi}_{Nk}(t_{Nk})|^2 \approx \sqrt{2\pi}/\Delta t$ or $\tilde{\varphi}_{Nk}(t_{Nk}) \approx \theta/\sqrt{\Delta t}$ with $\theta^2 = \sqrt{2\pi}$. So, replacing v_{Nk} by $(\sqrt{\Delta t}/\theta)\hat{v}_{Nk}$ with $\hat{v}_{Nk} = f(t_{Nk})$ makes sense because

$$v_{Nk} = \langle \tilde{\varphi}_{Nk}, f \rangle = \frac{1}{\sqrt{2\pi}} \int \tilde{\varphi}_{Nk}(t) f(t) dt \approx \frac{\sqrt{\Delta t}}{\theta} f(t_{Nk}).$$

Hence

$$\begin{aligned} f_N(t) \approx \hat{f}_N(t) &= \frac{2^{-N/2}}{\theta}\sum_k \hat{v}_{Nk}\varphi_{Nk}(t) = \frac{1}{\theta}\sum_k \hat{v}_{Nk}\varphi(2^N t - k), \qquad \hat{v}_{Nk} = f(t_{Nk}) \\ &= \frac{1}{\theta}\sum_l f(t - l\Delta t)\varphi(l). \end{aligned}$$

Letting $N \to \infty$, i.e., $\Delta t \to 0$ and knowing that $\sum_l \varphi(l) = \theta$, we see that $\lim_{N\to\infty} \hat{f}_N(t) = f(t)$. However this convergence is slow, as one can see by a taylor series expansion of $f(t - l\Delta t)$ around t, this convergence is only $O(\Delta t)$. It is therefore recommended to give as input the coefficients v'_{Nk} which are approximants for

$$v_{Nk} = \langle \tilde{\varphi}_{Nk}, f \rangle = \frac{1}{\sqrt{2\pi}} \int f(t) 2^{N/2} \overline{\tilde{\varphi}(2^N t - k)} dt$$

given by the Riemann sum

$$\begin{aligned} v'_{Nk} &= \frac{1}{\sqrt{2\pi}} \sum_n f_n 2^{N/2} \overline{\tilde{\varphi}(2^N n\Delta t - k)} \Delta t, \quad \Delta t = 2^{-N} \\ &= \frac{2^{-N/2}}{\sqrt{2\pi}} \sum_n f_n \overline{\tilde{\varphi}(n-k)}. \end{aligned}$$

Thus the function values are convolved with the values $\{\overline{\tilde{\varphi}(-n)}\}$, which is a low pass filtering operation.

6.6 Biorthogonal wavelets

The orthogonality requirement of the wavelets is however rather restrictive.

Theorem 6.6.1. *A symmetric FIR filter which is orthogonal can only have two nonzero coefficients.*

Proof. Consider for example the filter coefficients $c = [h_0\ h_1\ h_2\ h_2\ h_1\ h_0]$ with $h_0 \neq 0$. Orthogonality to its even translates gives $h_1 h_0 = 0$ (shift over 4 positions) and hence $h_1 = 0$, while also $h_2 h_0 = 0$ (shift over 2 positions) so that also $h_2 = 0$. This leaves us with $h = \frac{1}{\sqrt{2}}[1\ 0\ 0\ 1]$. Similar derivations show that we can only have solutions of the form $h = \frac{1}{\sqrt{2}}[1\ 1]$, with possibly an even number of zeros in between. Thus essentially the Haar wavelet is the only possible solution. $\square$

Only the Haar coefficients $h = \frac{1}{\sqrt{2}}[1\ 1]$ will lead to orthogonal wavelets.

Orthogonality also restricts smoothness, because imposing orthogonality conditions, leaves less freedom to impose conditions of the form $\mathsf{H}(\pi) = 0, \mathsf{H}'(\pi) = 0, \mathsf{H}''(\pi) = 0, \ldots$. As will be explained in Section 7.1, this corresponds to vanishing moments and to the smoothness of the wavelets.

Therefore weaker forms exist, like e.g., biorthogonal wavelets. We say that in a Hilbert space H the bases $\{e_k\}_k$ and $\{\tilde{e}_k\}_k$ are biorthogonal if $\langle e_k, \tilde{e}_l \rangle_H = \delta_{k-l}$, $k, l \in \mathbb{Z}$. Note that then for $f \in H$, we have two possible expansions:

$$\begin{aligned} f &= \sum_n a_n e_n, \quad a_n = \langle \tilde{e}_n, f \rangle_H \\ f &= \sum_n \tilde{a}_n \tilde{e}_n, \quad \tilde{a}_n = \langle e_n, f \rangle_H \end{aligned}$$

For biorthogonal wavelets, we have the functions φ and ψ which generate the bases $\{\varphi_{nk}\}$ and $\{\psi_{nk}\}$ (these are used on the synthesis side), but we also need some dual functions $\tilde{\varphi}$ and $\tilde{\psi}$ to generate the biorthogonal bases $\{\tilde{\varphi}_{nk}\}$ and $\{\tilde{\psi}_{nk}\}$ (used on the analysis side). These $\tilde{\varphi}$ and $\tilde{\psi}$ are defined by

$$\tilde{\varphi}(t) = \sqrt{2} \sum_k \tilde{h}_k \tilde{\varphi}(2t-k) \quad \text{and} \quad \tilde{\psi}(t) = \sqrt{2} \sum_k \tilde{g}_k \tilde{\varphi}(2t-k)$$

or, taking Fourier transforms

$$\tilde{\Phi}(2\omega) = \frac{1}{\sqrt{2}}\tilde{\mathsf{H}}(\omega)\tilde{\Phi}(\omega) \quad \text{and} \quad \tilde{\Psi}(2\omega) = \frac{1}{\sqrt{2}}\tilde{\mathsf{G}}(\omega)\tilde{\Phi}(\omega)$$

with

$$\tilde{\mathsf{H}}(\omega) = \sum_k \tilde{h}_k e^{-ik\omega} \quad \text{and} \quad \tilde{\mathsf{G}}(\omega) = \sum_k \tilde{g}_k e^{-ik\omega}.$$

It is now required that the following biorthogonality relations hold

$$\langle \varphi_{nk}, \tilde{\varphi}_{nl} \rangle_{L^2(\mathbb{R})} = \delta_{k-l}, \; n,k,l \in \mathbb{Z} \quad \text{and} \quad \left\langle \psi_{nk}, \tilde{\psi}_{ml} \right\rangle_{L^2(\mathbb{R})} = \delta_{m-n}\delta_{k-l}, \; n,m,k,l \in \mathbb{Z}$$

and

$$\left\langle \varphi_{nk}, \tilde{\psi}_{nl} \right\rangle = 0 \quad \text{and} \quad \langle \tilde{\varphi}_{nk}, \psi_{nl} \rangle = 0, \quad n,k,l \in \mathbb{Z}.$$

Set

$$\begin{array}{ll} \tilde{H}(z) = \sum_k \tilde{h}_k z^{-k}, & \tilde{G}(z) = \sum_k \tilde{g}_k z^{-k}, \\ H(z) = \sum_k h_k z^{-k}, & G(z) = \sum_k g_k z^{-k}, \end{array}$$

thus

$$\tilde{H}(e^{i\omega}) = \tilde{\mathsf{H}}(\omega), \quad H(e^{i\omega}) = \mathsf{H}(\omega), \quad \tilde{G}(e^{i\omega}) = \tilde{\mathsf{G}}(\omega), \quad \text{and} \quad G(e^{i\omega}) = \mathsf{G}(\omega).$$

Choosing $\tilde{H}$ and G and also $\tilde{G}$ and H as QMF by using (mixed) alternating flips

$$\bar{g}_n = (-1)^n \tilde{h}_{1-n}, \quad \tilde{g}_n = (-1)^n \bar{h}_{1-n}. \tag{6.2}$$

Then these conditions lead to the modulation matrix formulation

$$\begin{bmatrix} H(z) & H(-z) \\ G(z) & G(-z) \end{bmatrix}^t \begin{bmatrix} \tilde{H}_*(z) & \tilde{H}_*(-z) \\ \tilde{G}_*(z) & \tilde{G}_*(-z) \end{bmatrix} = 2 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Thus the filter bank is PR.

In terms of filter coefficients we have the orthogonality conditions

$$\sum_n \bar{g}_n \tilde{g}_{n-2k} = \delta_k \quad \text{and} \quad \sum_n \bar{h}_n \tilde{h}_{n-2k} = \delta_k.$$

In fact only one of them is needed, the other follows by the alternating flips.

The relevant spaces are

$$\begin{array}{ll} V_n = \operatorname{span}\{\varphi_{nk} : k \in \mathbb{Z}\}, & \tilde{V}_n = \operatorname{span}\{\tilde{\varphi}_{nk} : k \in \mathbb{Z}\} \\ W_n = \operatorname{span}\{\psi_{nk} : k \in \mathbb{Z}\}, & \tilde{W}_n = \operatorname{span}\{\tilde{\psi}_{nk} : k \in \mathbb{Z}\}. \end{array}$$

We have

$$V_n \perp \tilde{W}_n \quad \text{and} \quad W_n \perp \tilde{V}_n$$

for all $n \in \mathbb{Z}$. Also

$$V_n \oplus W_n = V_{n+1} \quad \text{and} \quad \tilde{V}_n \oplus \tilde{W}_n = \tilde{V}_{n+1}$$

i.e., we have complementary spaces (direct sums), but they are not orthogonal complements anymore.

The projection operators P_n on V_n and Q_n on W_n are given by

$$P_n f(t) = \sum_k \langle \tilde{\varphi}_{nk}, f \rangle \varphi_{nk}(t) = \sum_k v_{nk}\varphi_{nk}(t)$$

and

$$Q_n f(t) = \sum_k \left\langle \tilde{\psi}_{nk}, f \right\rangle \psi_{nk}(t) = \sum_k w_{nk}\psi_{nk}(t).$$

Analysis and reconstruction formulas are

$$\left[\begin{array}{c} v_n \\ w_n \end{array}\right] = \left[\begin{array}{c} \widehat{\tilde{\mathcal{H}}}^* \\ \widehat{\tilde{\mathcal{G}}}^* \end{array}\right] v_{n+1} \quad \text{and} \quad v_{n+1} = [\hat{\mathcal{H}} \ \ \hat{\mathcal{G}}] \left[\begin{array}{c} v_n \\ w_n \end{array}\right].$$

As before $\widehat{\tilde{\mathcal{H}}} = \tilde{\mathcal{H}} \uparrow$ and $\widehat{\tilde{\mathcal{G}}} = \tilde{\mathcal{G}} \uparrow$ while $\hat{\mathcal{H}} = \mathcal{H} \uparrow$ and $\hat{\mathcal{G}} = \mathcal{G} \uparrow$. Note that in this biorthogonal scheme, the projections on the analysis side are onto the spaces V_n and W_n, spanned by the primal functions φ_{nk} and ψ_{nk}. However the coefficients are computed as inner products with $\tilde{\varphi}_{nk}$ and $\tilde{\psi}_{nk}$ and thus one should use the filters $\tilde{H}$ and $\tilde{G}$ on the analysis side.

These formulas can be easily put into an algorithm. The following algorithms can be found in [13]. We suppose that all the coefficients are real and that c_k and $\tilde{c}_k$ are nonzero for $-L \leq k \leq L$ and that the $\tilde{d}_k$ and d_k are nonzero for $-M \leq k \leq M$. Moreover, suppose that $L = 2L' + 1$ and $M = 2M' + 1$ are odd. The "signal" is given as a vector of 2^K coefficients $v_{n,k}$, $k = 0, \ldots, 2^K - 1$.

The analysis is the result of the following DWT algorithm.

for $n = K - 1(-1)0$
  for $k = 0(1)2^n - 1$

$$v_{nk} = \sum_{i=-L}^{L} \tilde{h}_i v_{n+1,(i+2k) \bmod 2^{n+1}}$$

$$w_{nk} = \sum_{i=-M}^{M} \tilde{g}_i v_{n+1,(i+2k) \bmod 2^{n+1}}$$

  endfor
endfor

The inverse DWT is given by the following reconstruction algorithm

for $n = 1(1)K$
  for $k = 0(1)2^n - 1$
    if k even **then**

$$v_{nk} = \sum_{i=-L'}^{L'} h_{2i} v_{n-1,(k/2-i) \bmod 2^{n-1}} + \sum_{i=-M'}^{M'} g_{2i} w_{n-1,(k/2-i) \bmod 2^{n-1}}$$

    else k odd

$$v_{nk} = \sum_{i=-L'-1}^{L'} h_{2i+1} v_{n-1,((k-1)/2-i) \bmod 2^{n-1}} + \sum_{i=-M'-1}^{M'} g_{2i+1} w_{n-1,((k-1)/2-i) \bmod 2^{n-1}}$$

    endif
  endfor
endfor

To conclude we mention the following theorem about the support of biorthogonal wavelets.

Theorem 6.6.2. *[7] If*

$$H(z) = \sum_{k=N_1}^{N_2} h_k z^k \quad \text{and} \quad \tilde{H}(z) = \sum_{k=\tilde{N}_1}^{\tilde{N}_2} \tilde{h}_k z^k,$$

$h_{N_1} \neq 0 \neq h_{N_2}$*, and* $\tilde{h}_{\tilde{N}_1} \neq 0 \neq \tilde{h}_{\tilde{N}_2}$*, then*

$$\mathrm{supp}(\varphi) = [N_1, N_2], \quad \mathrm{supp}(\tilde{\varphi}) = [\tilde{N}_1, \tilde{N}_2],$$

and

$$\mathrm{supp}(\psi) = [\frac{1}{2}(N_1 - \tilde{N}_2 + 1), \frac{1}{2}(N_2 - \tilde{N}_1 + 1)], \quad \mathrm{supp}(\tilde{\psi}) = [\frac{1}{2}(\tilde{N}_1 - N_2 + 1), \frac{1}{2}(\tilde{N}_2 - N_1 + 1)].$$

6.7 Semi-orthogonal wavelets

In the biorthogonal case we had the different spaces

$$\begin{array}{ll} V_n = \mathrm{span}\{\varphi_{nk} : k \in \mathbb{Z}\}, & \tilde{V}_n = \mathrm{span}\{\tilde{\varphi}_{nk} : k \in \mathbb{Z}\} \\ W_n = \mathrm{span}\{\psi_{nk} : k \in \mathbb{Z}\}, & \tilde{W}_n = \mathrm{span}\{\tilde{\psi}_{nk} : k \in \mathbb{Z}\}. \end{array}$$

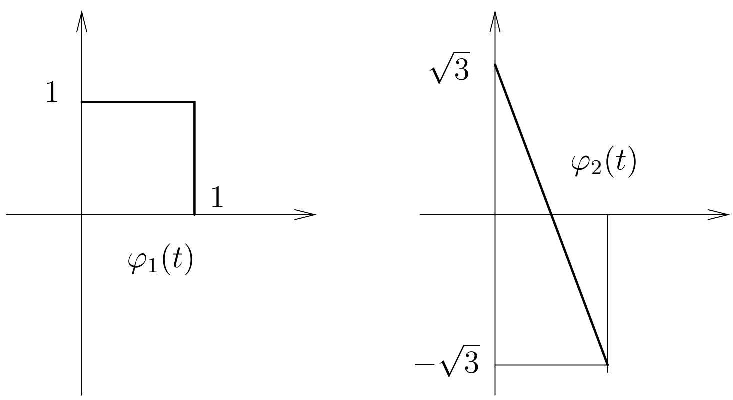
However, it may happen that $V_0 = \tilde{V}_0$, without φ and $\tilde{\varphi}$ being the same. In that case we have $V_n = \tilde{V}_n$ for all n and because $V_n \perp \tilde{W}_n$ and $\tilde{V}_n \perp W_n$, we also have $W_n = \tilde{W}_n$ for all n. In that case we do have an orthogonal multiresolution analysis, but the basis functions are not orthogonal. They are called *semi-orthogonal* in that case. We still have $W_j \perp W_i$ for $i \neq j$, it follows that $\psi_{jk} \perp \psi_{il}$ for $i \neq j$, but orthogonality in the same resolution level may not hold.

6.8 Multiwavelets

More freedom can also be introduced by having V_n generated by the translations of not just one scaling function, but by the translates of two or more scaling functions. Thus one may consider an R-vector of scaling functions $\varphi(t)^T = [\varphi_1(t), \ldots, \varphi_R(t)]$ and we then set $V_n = \text{span}_k\{\varphi_{r,n,k} : r = 1, \ldots, R\}$ where $\varphi_{r,n,k}(t) = 2^{n/2}\varphi_r(2^n t - k)$.

Example 6.8.1. A very simple example is given by the 2 scaling functions shown in Figure 6.3 The first φ_1 is the Haar scaling function. For the second, one can check that

Figure 6.3: Example of multiwavelet scaling functions



$$\varphi_2(t) = \frac{\sqrt{3}}{2}\varphi_1(2t) + \frac{1}{2}\varphi_2(2t) - \frac{\sqrt{3}}{2}\varphi_1(2t-1) + \frac{1}{2}\varphi_2(2t-1).$$

So we have

$$\begin{bmatrix} \varphi_1(t) \\ \varphi_2(t) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \sqrt{3}/2 & 1/2 \end{bmatrix} \begin{bmatrix} \varphi_1(2t) \\ \varphi_2(2t) \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ -\sqrt{3}/2 & 1/2 \end{bmatrix} \begin{bmatrix} \varphi_1(2t-1) \\ \varphi_2(2t-1) \end{bmatrix}.$$

$\diamond$

In general we have for multiwavelets a matrix version of the dilation equation:

$$\varphi(t) = \sqrt{2}\sum_k h_k \varphi(2t-k)$$

where $\varphi(t)$ is a 2×1 vector and the h_k are 2×2 matrices.

For the complementary wavelet spaces $W_n = V_{n+1} \ominus V_n = \text{span}_k\{\psi_{r,n,k} : r = 1, \ldots, R\}$ with $\psi_{r,n,k}(t) = 2^{n/2}\psi(2^n t - k)$. The multiwavelet $\psi(t) = [\psi_1, \ldots, \psi_R]^T$ satisfies an equation of the form $(R = 1)$

$$\psi(t) = \sqrt{2}\sum_k g_k \psi(2t-k)$$

where $\psi(t)$ is a 2×1 vector and the g_k are 2×2 matrices. The basics of the whole theory can be repeated for this matrix-vector generalization. For example a necessary (but not sufficient) condition for orthogonality is expressed by equations

$$\begin{aligned} \mathsf{H}(\omega)\mathsf{H}(\omega)^* + \mathsf{H}(\omega+\pi)\mathsf{H}(\omega+\pi)^* &= I_R \\ \mathsf{G}(\omega)\mathsf{G}(\omega)^* + \mathsf{G}(\omega+\pi)\mathsf{G}(\omega+\pi)^* &= I_R \\ \mathsf{H}(\omega)\mathsf{G}(\omega)^* + \mathsf{H}(\omega+\pi)\mathsf{G}(\omega+\pi)^* &= 0_R \end{aligned}$$

where $\mathsf{H}(\omega) = \sum_k h_k e^{\mathbf{i}\omega k}$ and $\mathsf{G}(\omega) = \sum_k g_k e^{\mathbf{i}\omega k}$. However, the relation between h_k and g_k is now more complicated than taking an alternating flip. On the other hand, this leaves us more freedom to design wavelets, even if the scaling functions are already fixed.

In general, there is no straightforward relation between the number of nonzero coefficients h_k and g_k and the support of the multiwavelets.

Derive a FWT algorithm for multiwavelets as an exercise.

6.9 Exercises

1. Prove that if $\mathcal{H}$ represents a filter operator, $\downarrow$ is a downsampling operator, and $\uparrow$ the corresponding upsampling operator, then the adjoint operator $(\mathcal{H}\uparrow)^*$ is given by $\hat{\mathcal{H}}^* = \downarrow \mathcal{H}^*$.

2. Derive the filter coefficients for the Daubechies wavelets given in section 6.4.

3. Prove Theorem 6.6.2.

4. For orthogonal multiwavelets ψ and scaling functions φ, define

$$v_{r,n,k} = \langle \varphi_{r,n,k}, f \rangle, \quad w_{r,n,k} = \langle \psi_{r,n,k}, f \rangle$$

and set

$$v_{nk} = [v_{1,n,k}, \ldots, v_{R,n,k}]^T, \quad w_{nk} = [w_{1,n,k}, \ldots, w_{R,n,k}]^T.$$

Prove that

$$v_{n,k} = \sum_l h^*_{l-2k} v_{n+1,l}, \quad w_{n,k} = \sum_j g^*_{l-2k} v_{n+1,l},$$

and

$$v_{n+1,k} = \sum_j (h_{k-2l} v_{n,l} + g_{k-2l} w_{nl}).$$

5. (**Wavelet crime**) Prove that $v_{Nk} = (2^{-N/2}/\sqrt[4]{2\pi})\hat{v}_{Nk} + O(2^{-3N/2})$ where $v_{N,k} = \langle f, \tilde{\varphi}_{Nk} \rangle$ and $\hat{v}_{Nk} = f(2^{-N}k)$.
Hint: Therefore write first

$$v_{Nk} = \frac{2^{-N/2}}{\sqrt{2\pi}} \int_{\mathbb{R}} \tilde{\varphi}(t) f(2^{-N}(t+k))dt.$$

Next expand $f(2^{-N}(t+k))$ in a Taylor series at the point $t_{Nk} = 2^{-N}k$.

Chapter 7

Approximating properties and wavelet design

7.1 Smoothness

The condition $\sum_n (-1)^n h_n = 0$ ensured that $\mathsf{H}(\pi) = 0$. This is a special case of a set of more general conditions which require $\mathsf{H}(\omega)$ to have a zero of order $p-1$ at $\omega = \pi$. This would give

$$\sum_n (-1)^n n^k h_n = 0 \quad , \quad k = 0, 1, \ldots, p-1.$$

One can show that for the box function $p = 1$, for the hat function $p = 2$ and for D_2, $p = 2$. The quadratic spline has $p = 3$, the cubic spline $p = 4$.

What do such conditions mean and where do they come from? We consider the biorthogonal case. We say that a multiresolution analysis $\{V_n\}$ is *of order p* or it has *p vanishing moments* if $t^k \in V_0$ for $k = 0, \ldots, p-1$. We first note that $t^k \in V_0$ implies that $t^k \in V_j$ for all $j \in \mathbb{Z}$. Indeed, $t^k \in V_0 \Leftrightarrow (2^j t)^k \in V_j$. Since V_j is a linear space, $t^k = 2^{-jk}(2^j t)^k \in V_j$. The following lemma shows why the term "vanishing moments" is justified.

Lemma 7.1.1. *If $t^k \in V_0$ then the kth derivative of the Fourier transform of $\tilde{\psi}$ vanishes at the origin: $\tilde{\Psi}^{(k)}(0) = 0$.*

Proof. Because $t^k \in V_0 \perp \tilde{W}_0$, we have $\left\langle t^k, \tilde{\psi} \right\rangle = 0$. Now the Fourier transform is

$$\tilde{\Psi}(\omega) = \frac{1}{\sqrt{2\pi}} \int \tilde{\psi}(t) e^{-i\omega t} dt$$

so that the derivative is

$$\tilde{\Psi}^{(k)}(\omega) = \frac{1}{\sqrt{2\pi}} \int \tilde{\psi}(t)(-it)^k e^{-i\omega t} dt = (-i)^k \frac{1}{\sqrt{2\pi}} \int \tilde{\psi}(t) t^k e^{-i\omega t} dt$$

Therefore $\int t^k \tilde{\psi}(t) dt = 0$ is equivalent with $\tilde{\Psi}^{(k)}(0) = 0$. $\square$

Theorem 7.1.2. *If the multiresolution analysis has order $p > 0$, then $\tilde{\mathsf{G}}(\omega)$ has a zero of order $p-1$ at the origin and $\mathsf{H}(\omega)$ has a zero of order $p-1$ at π. Here $\tilde{\mathsf{G}}(\omega) = \tilde{G}(e^{-i\omega}) = \sum_k \tilde{g}_k e^{ik\omega}$ and $\mathsf{H}(\omega) = H(e^{i\omega}) = \sum_k h_k e^{-ik\omega}$.*

Proof. This follows from the dilation equation in the Fourier domain: $\tilde{\Psi}(2\omega) = \frac{1}{\sqrt{2}}\tilde{\mathsf{G}}(\omega)\tilde{\Phi}(\omega)$. Differentiating $p-1$ times and using $\tilde{\Psi}^{(k)}(0) = 0$ for $k = 0, \ldots, p-1$ gives the result for $\tilde{\mathsf{G}}$.

Next recall that $\tilde{\mathsf{G}}(\omega) = -e^{-i\omega}\overline{\mathsf{H}(\omega+\pi)}$, so that by differentiating $p-1$ times, the result for H follows from the previous result for $\tilde{\mathsf{G}}$. □

By the previous result, it follows that we may write

$$\mathsf{H}(\omega) = H(e^{i\omega}) = \left(\frac{1+e^{-i\omega}}{2}\right)^p \mathsf{Q}(\omega).$$

Note also that in terms of the filter coefficients, $\mathsf{H}^{(k)}(\pi) = 0$ can be written as $\sum_n (-1)^n n^k h_n = 0$.

So far we have only considered the MRA $\{V_j\}$, but one could make a similar discussion for the MRA $\{\tilde{V}_j\}$. Also this may have a number of vanishing moments, say q, with q not necessarily equal to p. It is then equivalent with $\mathsf{G}(\omega)$ having a zero of order q at the origin or also with $\tilde{H}(\omega)$ having a zero of order q at π.

7.2 Approximation

Let $P_j f$ be the oblique projection of $f \in L_2$ onto V_j, parallel to $\tilde{V}_j^{\perp}$, thus

$$P_j f(t) = \sum_k \langle \tilde{\varphi}_{jk}, f\rangle \, \varphi_{jk}(t).$$

Note that this projection is only orthogonal if $\tilde{\varphi} = \varphi$. One can prove (see [21])

Theorem 7.2.1. *If $\mathsf{H}(\omega)$ has a zero of order p in π and if $f \in L_2$ is smooth enough (its pth derivative $f^{(p)}$ is in L_2), then there is some $C > 0$ such that*

$$\|f - P_j f\| \leq C 2^{-jp} \|f^{(p)}\|.$$

This theorem shows that wavelets as approximation tools are well suited for piecewise smooth functions. Singularities in the higher derivatives are well localized in t at specific resolution levels.

Moreover, the higher the order, the better their approximating abilities. These observations thus show that among the simple examples of wavelets, the splines are best in approximating, but ... they are not orthogonal! The wavelet D_2 is as good as the linear spline and it is orthogonal. Of course a sine/cosine system is also good in approximation and it is orthogonal, but they do not have compact support as D_2 has. Therefore D_2 is in some sense the simplest nontrivial compactly supported orthogonal wavelet one can imagine.

7.3 Design properties: overview

There are several properties for the design of a wavelet basis that one could want to be fulfilled.

1. orthogonality: is sometimes too restrictive
2. compact support: defined by the length of the filters
3. rational coefficients: this might be an issue for hardware implementation
4. symmetry: The wavelet transform of the mirror of an image is not the mirror of the wavelet transform, unless the wavelets are symmetric.
5. smoothness: determined by the number of primal or dual vanishing moments. The primal vanishing moments determine the smoothness of the reconstruction. The dual vanishing moments determine the convergence rate of the MRA projections and are necessary to detect singularities.
6. interpolation: it may be required that some function values are exactly interpolated.

7.4 Some well known wavelets

In the next sections we discuss some of the most popular wavelets.

7.4.1 Haar wavelet

This has appeared several times in the text. Their properties are

1. orthogonal
2. compact support
3. the scaling function is symmetric
4. the wavelet function is anti-symmetric
5. it has only one vanishing moment (a minimum)

It is the only one which is compactly supported, orthogonal and has symmetry.

7.4.2 Shannon or sinc wavelet

Its Fourier transform $\mathsf{H}(\omega)$ is

$$\mathsf{H}(\omega) = \begin{cases} \sqrt{2}, & |\omega| < \pi/2 \\ 0, & |\omega| > \pi/2 \end{cases}$$

This is an ideal low pass filter. This corresponds to (see example 5.6.13. We leave out the factor $\sqrt{2\pi}$, which is only needed for normalization. Recall that the solution of the dilation equation is only defined up to a multiplicative factor. See also Section 3.3.)

$$\varphi(t) = \frac{\sin \pi t}{\pi t}.$$

The filter coefficients are

$$h_0 = \frac{1}{\sqrt{2}}, \quad h_{2n} = 0, \ n \neq 0, \quad h_{2n+1} = \frac{(-1)^n \sqrt{2}}{(2n+1)\pi}.$$

The wavelet function is derived from the corresponding high pass filter

$$\mathsf{G}(\omega) = \begin{cases} \sqrt{2}, & |\omega| > \pi/2 \\ 0, & |\omega| < \pi/2 \end{cases}$$

The filter coefficients are now

$$g_0 = \frac{1}{\sqrt{2}}, \quad g_{2n} = 0, \ n \neq 0, \quad g_{2n+1} = \frac{(-1)^{n+1} \sqrt{2}}{(2n+1)\pi}.$$

This leads to (exercise)

$$\psi(t) = 2\varphi(2t) - \varphi(t) = \frac{\sin 2\pi t - \sin \pi t}{\pi t}.$$

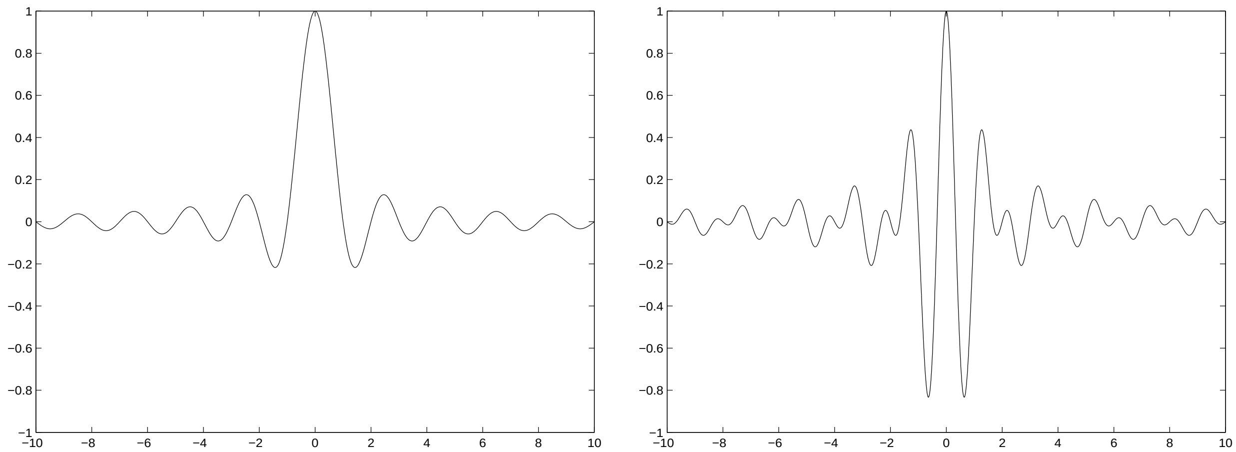
They are

1. orthogonal
2. symmetric scaling and wavelet function
3. infinite number of vanishing moments
4. infinite support and slowly decaying IIR-filters (non-causal)

7.4.3 Mexican hat function

This is a function used in CWT. It has the form $(t^2 - 1)\exp(-\frac{1}{2}t^2)$ and has been discussed in Example 5.3.1. It is the second derivative of the Gaussian. It satisfies the admissibility condition and has two vanishing moments.

Figure 7.1: Shannon scaling function and wavelet



7.4.4 Morlet wavelet

This is another function used in CWT. It is a modulated Gaussian: $\psi(t) = e^{i\alpha t} e^{-t^2/2}$ discussed in Example 5.3.2. It satisfies the admissibility condition only approximately. However if $\alpha > 5.5$, the error can be neglected numerically.

7.4.5 Meyer wavelets

We will not discuss this in detail. Basically they are obtained as follows. For the sinc wavelets, $\mathsf{H}(\omega)$ was a block function. For the Meyer wavelets, this block function is smoothed. Like in Figure 7.3.

They are

1. orthogonal
2. symmetric scaling and wavelet function
3. band limited
4. infinite support but faster decaying than sinc
5. infinitely many times differentiable

7.4.6 Daubechies maxflat wavelets

Here one looks for orthogonal filters with compact support. Let $h_0, \ldots, h_{2p-1}$ be the nonzero coefficients (recall that there has to be an even number of them). By normalization $\sum_k h_k = \sqrt{2}$. They should also satisfy the orthogonality condition (recall that this was realized by

Figure 7.2: Meyer scaling function and wavelet

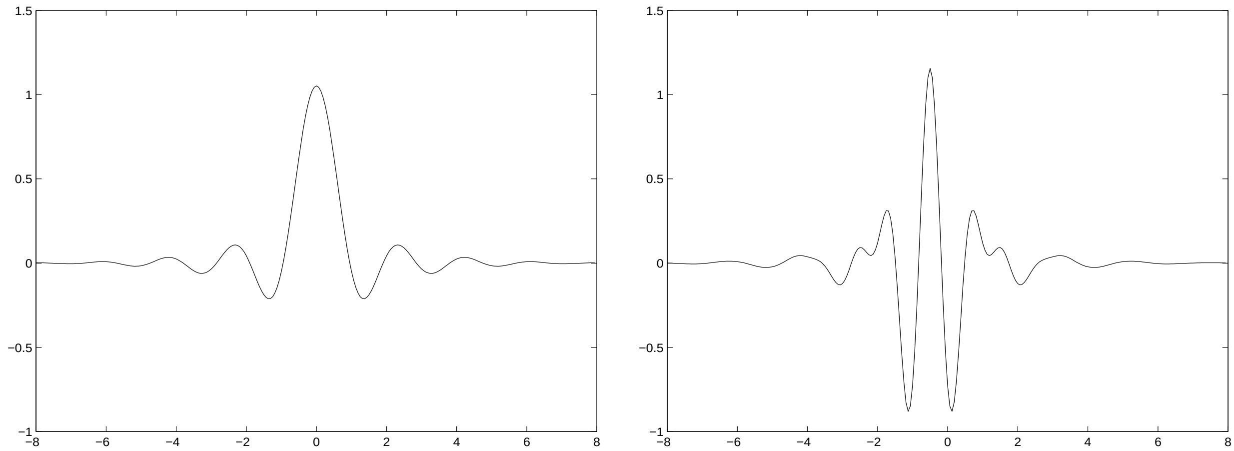
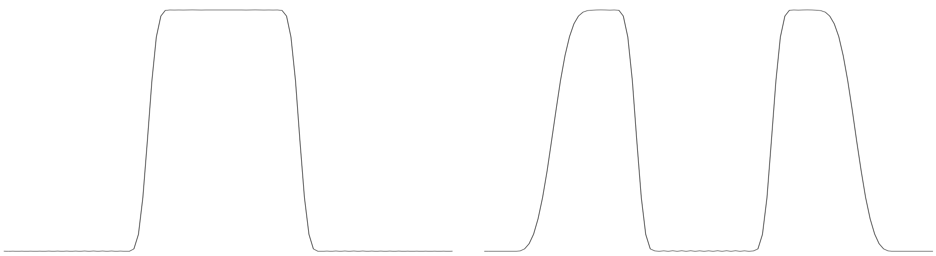
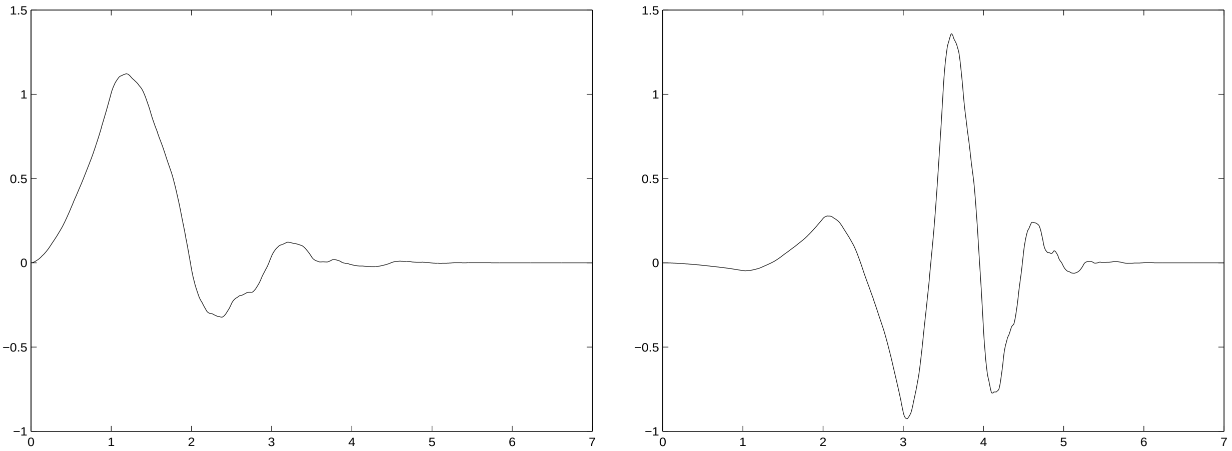


Figure 7.3: Fourier transform of the low pass and high pass filters for the Meyer wavelet



the double shift orthogonality) $\sum_n \bar{h}_n h_{n-2k} = \delta_k$. This gives only nontrivial equations for $k = 0, \ldots, p-1$. Thus we have p conditions. The freedom that remains can be used to generate vanishing moments $\mathsf{H}^{(i)}(\pi) = 0$ for $i = 0, \ldots, p-1$. Recall that $\mathsf{H}(\pi) = 0$ is a consequence of the orthogonality (partition of unity). For $p = 2$ we get the coefficients and figures as in Example 5.6.7. See also Section 6.4 for the derivation and for the coefficients in the case $p = 3$. The case $p = 1$ corresponds to the Haar wavelet. The functions become smoother for higher p.

Figure 7.4: Daubechies maxflat scaling function and wavelet $p = 4$



These wavelets have the following properties

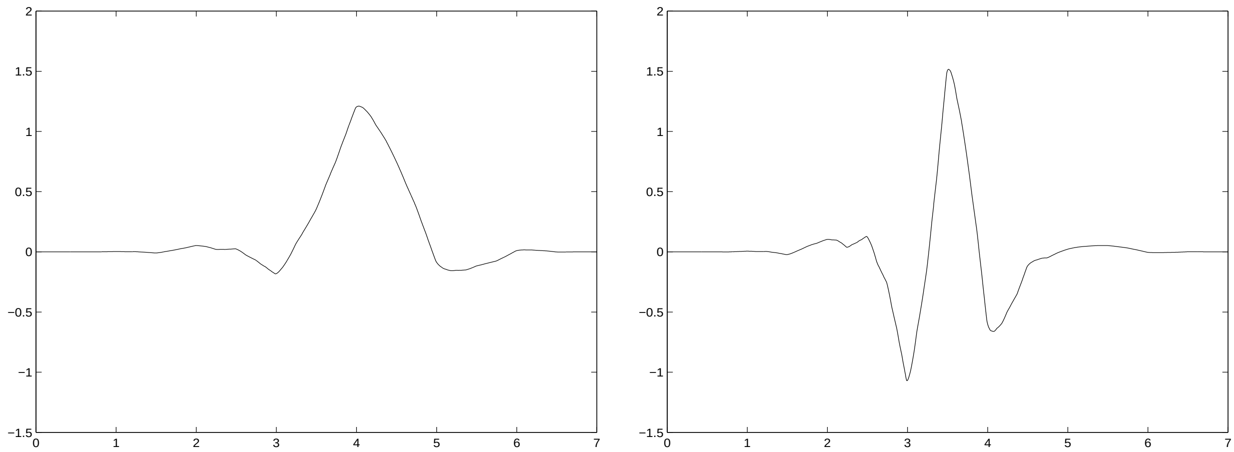
1. orthogonal
2. compact support
3. there is no symmetry for $p > 1$
4. p vanishing moments
5. filter length is $2p$

7.4.7 Symlets

The solution for the maxflat wavelets which was given by Daubechies is not always unique. She gave solutions with minimal phase. This means that all the zeros of $H(z)$ are inside the unit disk. Other choices can lead to more symmetric solutions. They are never completely symmetric though. Symlets have the following properties

1. orthogonal

Figure 7.5: Symlet scaling function and wavelet $p = 4$



2. compact support
3. filter length is $2p$
4. ψ has p vanishing moments
5. φ is nearly linear phase

7.4.8 Coiflets

Consider a wavelet with p vanishing moments:

$$\int t^k \psi(t) dt = 0, \quad k = 0, \dots, p-1.$$

On the other hand we know that $\int \varphi(t) dt \neq 0$, but if we require also

$$\int t^k \varphi(t) dt = 0, \quad k = 1, \dots, p-1,$$

then we have for any polynomial P of degree at most p that

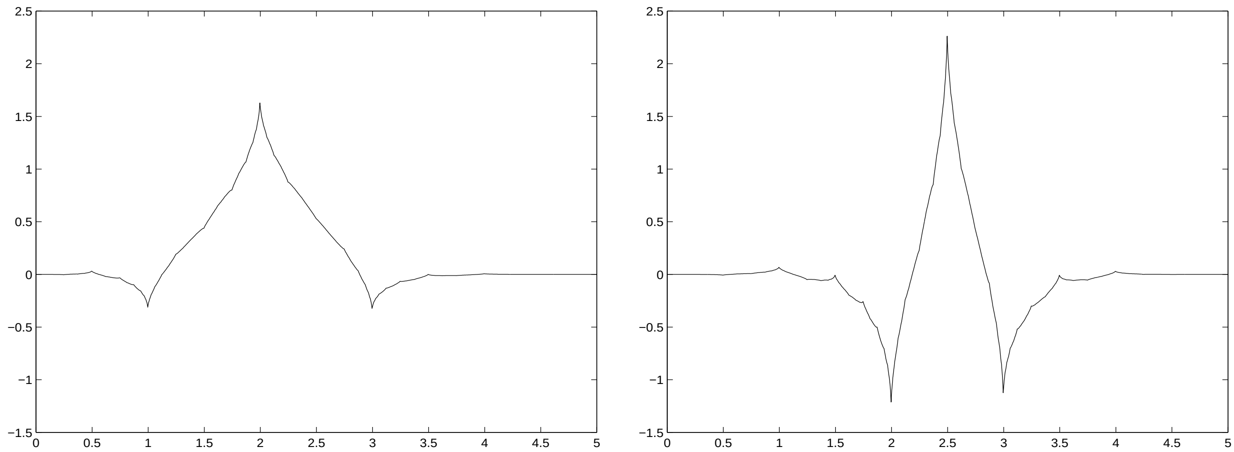
$$a_{0,k} = \int P(t)\varphi(t-k) = \int P(u+k)\varphi(u) du = P(k).$$

Thus, the projection

$$\hat{f}_N(t) = \sum_k f(2^{-N}k)\varphi(2^N t - k)$$

converges to f as fast as $O(2^{-Np})$. Recall that in general, this was only $O(2^{-N})$. These extra conditions require more coefficients, so that we have less compact support. Properties of Coiflets

Figure 7.6: Coiflet scaling function and wavelet $p = 2$



1. orthogonal
2. compact support
3. filter length $6p$
4. almost symmetric
5. ψ has $2p$ vanishing moments and φ has $2p - 1$ vanishing moments

7.4.9 CDF or biorthogonal spline wavelets

The Cohen-Daubechies-Feauveau (CDF) [7] wavelets are biorthogonal wavelets for which a number of moments are made zero:

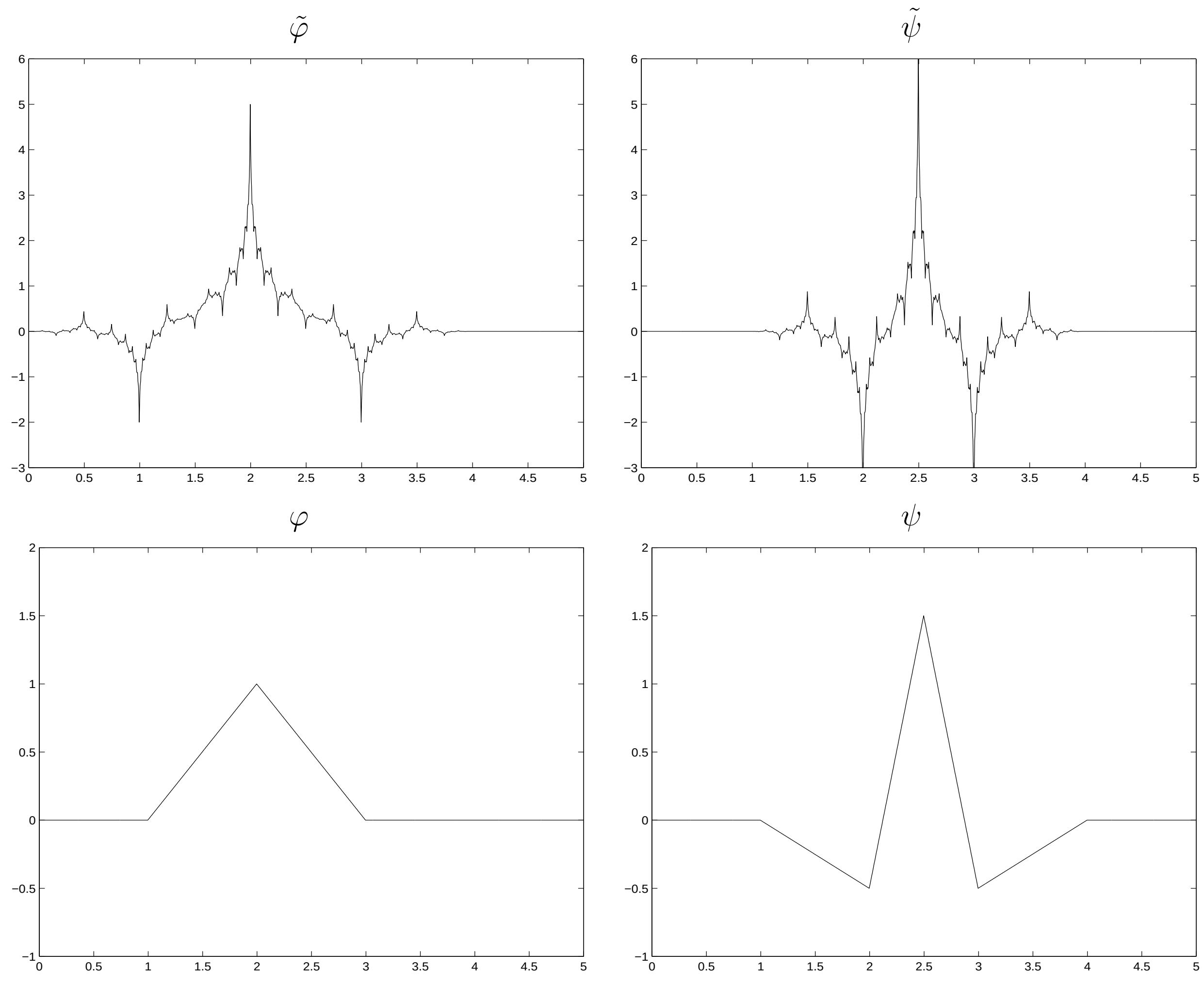
$$\mathsf{H}^{(k)}(\pi) = 0, \quad k = 0, \ldots, p-1 \quad \text{and} \quad \tilde{\mathsf{H}}^{(k)}(\pi) = 0, \quad k = 0, \ldots, q-1$$

The larger p, the smoother $\tilde{\psi}$ and the larger q, the smoother ψ. A larger p implies more filter coefficients h_k, thus more filter coefficients $\tilde{g}_k$, thus a larger support for the wavelet $\tilde{\psi}$, while a larger q needs longer filters $\tilde{H}$ and G and a larger support of the wavelet ψ.

These wavelets are indicated by CDF(p, q). Again one can use linear algebra to find the filter coefficients. Define a matrix $\mathbf{T}$ like in (6.1) and a similar one with tildes, then biorthogonality requires that $\mathbf{T}\tilde{\mathbf{T}}^* = \mathbf{I}$. Like in the orthogonal case, this leads to a number of biorthogonality relations for the filter coefficients, but because the $\tilde{h}_k$ and the h_k are now different, one has more freedom to impose smoothness conditions. The latter determine the flatness of the wavelets at the end points of their support and they are sometimes called maxflat filters.

The next figures give a number of examples. Each figure contains a plot of φ and ψ: The φ-function is postivive, the ψ-function oscillates. Note that wavelet functions of type (p, q) are even functions if p and q are even, while they are odd when p and q are odd. Recall that such a kind of symmetry was not possible for orthogonal wavelets. The functions become increasingly smooth as p increases; the wavelets oscillate more as q increases.

Figure 7.7: CDF(2,2) wavelet



Concerning the support of the functions φ, $\tilde{\varphi}$, ψ, and $\tilde{\psi}$, we refer to Theorem 6.6.2. The MATLAB wavelet toolbox has a routine `wvdtool` which allows to generate figures of primal and dual scaling functions and wavelet functions for all types of biorthogonal wavelets and all kinds of other wavelet functions. An example of the output is given in Figure 7.9. This example shows a general trend: the φ and ψ are in general smoother functions than the $\tilde{\varphi}$ and $\tilde{\psi}$. Thus the smoother filters are used as basis functions in the synthesis phase and the

Figure 7.8: CDF primal φ and ψ functions of type (p, q), $p, q = 1, 3, 5$ and type (p, q), $p, q = 2, 4, 6$

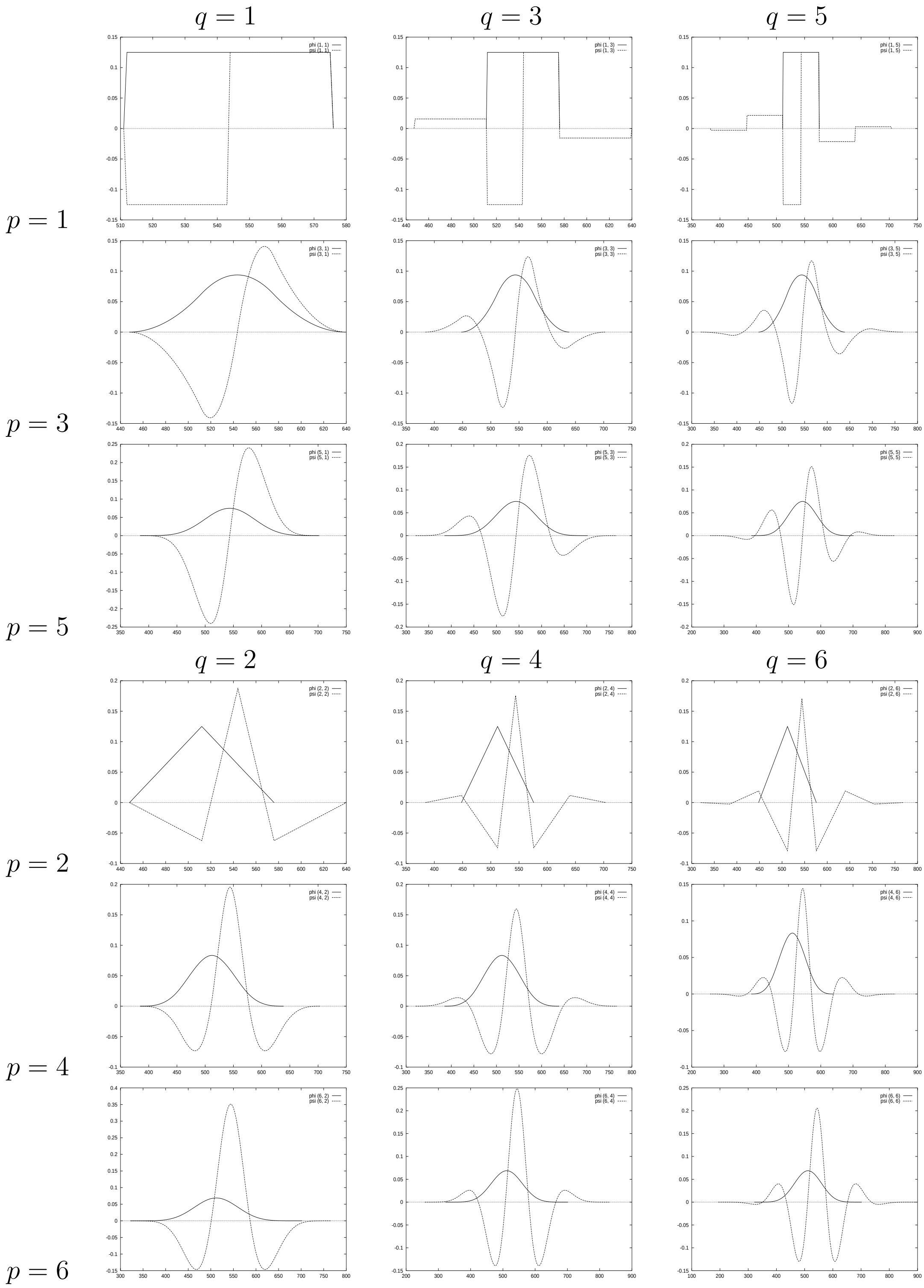
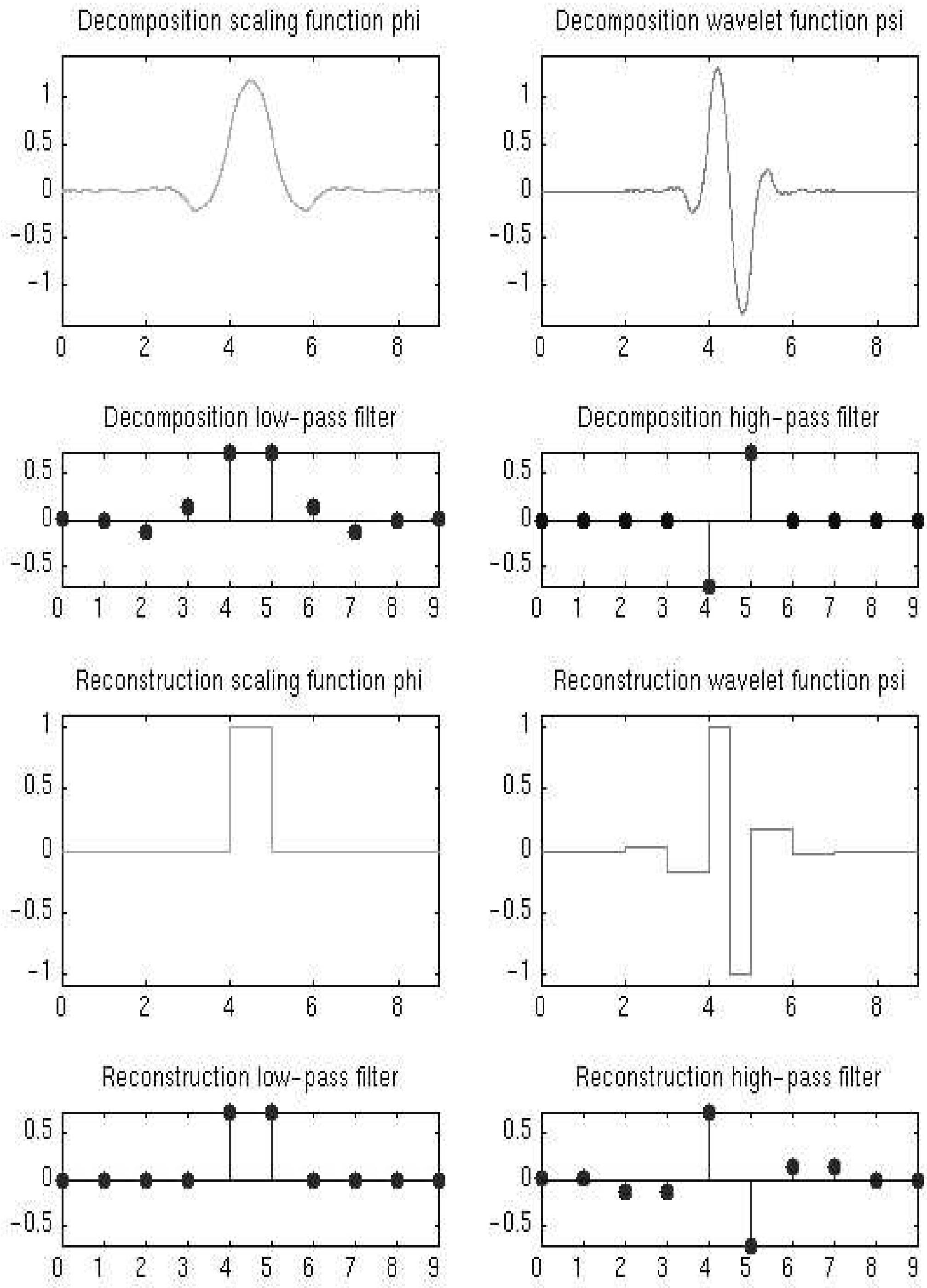


Figure 7.9: Biorthogonal wavelet of type (1,5)



less smooth ones in the analysis phase. This has some advantages in image processing.

7.5 Battle-Lemarié wavelet

We conclude with the description of the *Battle-Lemarié family of wavelets*. These wavelets are based on B-splines. Let $C^N(\mathbb{R})$ be the space of N times continuously differentalble functions, $C^0(\mathbb{R}) = C(\mathbb{R})$, the continuous functions and $C^{-1}(\mathbb{R})$ the piecewise continuous functions; $P_N[x]$ is the space of polynomials of degree at most N. The space

$$\mathcal{C}^N = \left\{ f : f \in C^{N-1}(\mathbb{R}),\, f|_{[k,k+1]} \in P_N[x], k \in \mathbb{Z} \right\}$$

is the space of *cardinal splines of order* N. For example, $\mathcal{C}^0$ is the space of piecewise constant functions and has for basis the indicator functions $\chi_{[k,k+1)}$, $k \in \mathbb{Z}$ (Haar basis). Piecewise linear splines are represented by $\mathcal{C}^1$. Cubic splines correspond to $N = 3$.

B-Spline wavelets have many advantages. The have compact support, the filter coefficients are particularly "easy" and they are as smooth as can be. The main disadvantage is that they are not orthogonal (except for the Haar wavelets).

The most interesting basis to work with is given by B-splines. They have a compact support and can be found by convolutions of the zero-order spline, i.e., the box function. For example $\varphi_0 = \chi_{[0,1)}$, $\varphi_1 = \varphi_0 * \varphi_0$, $\varphi_2 = \varphi_0 * \varphi_0 * \varphi_0$, a cubic spline is the convolution of 4 box functions etc.

The easiest way to deal with these convolutions is to move to the Fourier domain. The Fourier transform of $\varphi_0 = \chi_{[0,1)}$ is

$$\Phi_0(\omega) = \frac{1}{\sqrt{2\pi}} \frac{1 - e^{-i\omega}}{i\omega} \quad \Rightarrow \quad \Phi_{N-1}(\omega) = [\Phi_0(\omega)]^N = \left[\frac{1}{\sqrt{2\pi}} \frac{1 - e^{-i\omega}}{i\omega} \right]^N .$$

The function φ_{N-1} is a piecewise polynomial of degree $N-1$ and the jumps of the $(N-1)$st derivative at the points $k = 0, 1, \ldots, n$ are the scaled alternating binomial coefficients $(2\pi)^{-N/2}(-1)^k \binom{N}{k}$. This is most easily seen by the fact that a derivative in the t-domain corresponds to a multiplication with $i\omega$ in the Fourier domain. Thus for example, the Fourier transform of the 4th derivative of φ_3 is $(2\pi)^{-2}(1-e^{-i\omega})^4$, which means that the 3rd derivative has jumps $(2\pi)^{-2}[1, -4, 6, -4, 1]$.

The filter corresponding to the box function has coefficients $\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}$, and transfer function $H(z) = \frac{1}{\sqrt{2}}(1 + z^{-1})$. Thus the transfer function of the filter corresponding to φ_{N-1} is $[(1 + z^{-1})/\sqrt{2}]^N$. Thus the filter coefficients for the cubic B-spline are for example $\frac{1}{4}(1, 4, 6, 4, 1)$.

From the obvious identity

$$\left[\frac{1}{\sqrt{2\pi}} \frac{1 - e^{-i\omega}}{i\omega} \right]^N = \left[\frac{1 + e^{-i\omega/2}}{2} \right]^N \left[\frac{1}{\sqrt{2\pi}} \frac{1 - e^{-i\omega/2}}{i\omega/2} \right]^N \equiv \Phi_{N-1}(\omega) = \frac{1}{\sqrt{2}} \mathsf{H}(\omega/2)\Phi_{N-1}(\omega/2)$$

we find by inverse Fourier transform that the dilation equation is

$$\varphi_{N-1}(t) = 2^{1-N} \sum_{k=0}^{N} \binom{N}{k} \varphi_{N-1}(2t - k).$$

The space V_0 contains all smooth splines of degree $N-1$ and is generated by the basis of B-splines on unit intervals: $V_0 = \text{span}\{\varphi_{N-1}(t-k) : k \in \mathbb{Z}\}$. This basis is *not orthogonal.* The space V_1 contains piecewise polynomials on half intervals etc. In general $V_n = \text{span}\{2^{n/2}\varphi_{N-1}(2^n t - k) : k \in \mathbb{Z}\}$. It can be shown that these $\{V_n\}$ form a MRA of $L^2(\mathbb{R})$.

These basis functions are very good in approximating. For example, $C(\omega)$ has a zero of order $p = N$ at $\omega = \pi$, and as we have seen in section 7.1, this means that the polynomials of to degree $N-1$ are all in V_0, i.e., they can be represented exactly by the B-splines from V_0. Spline wavelets are very smooth.

There is a major drawback though: B-splines functions are not orthogonal. If one would apply an orthogonalization procedure, then the orthogonal basis functions, although decaying fast, would be supported on the whole real line.

Compactly supported spline wavelets are only possible when the orthogonality condition is relaxed to a biorthogonality condition.

7.6 Discrete versus continuous wavelet transforms revisited

Historically, the continuous wavelet transform (CWT) came first and was used by physicists as an alternative for the short time or windowed Fourier transform. The discrete wavelet transform (DWT) is more popular for applications in numerical analysis and signal or image processing.

Recall that the CWT of a signal $f(t)$ is given by

$$\mathcal{W}_\psi f = F(a,b) = \frac{1}{\sqrt{2\pi C_\psi}} \int_{\mathbb{R}} \overline{\psi_{a,b}(t)} f(t) dt = \frac{1}{\sqrt{C_\psi}} \langle \psi_{a,b}, f \rangle_{L^2(\mathbb{R})}$$

where

$$\psi_{a,b}(t) = \sqrt{|a|}\psi\left(a(t-b)\right) \quad \text{and} \quad C_\psi = \int_{\mathbb{R}} \frac{|\Psi(\omega)|^2}{|\omega|} d\omega.$$

The inverse transform is then given by

$$\mathcal{W}_\psi^{-1} F = f(t) = \frac{1}{\sqrt{2\pi C_\psi}} \iint_{\mathbb{R}^2} F(a,b)\psi_{a,b}(t) da\, db.$$

This requires that $0 < C_\psi < \infty$. In other words, the *admissibility condition*

$$C_\psi = \int_{\mathbb{R}} |\Psi(\omega)|^2 \frac{d\omega}{|\omega|} < \infty$$

should be satisfied. ($\Psi(\omega)$ is the Fourier transform of $\psi(t)$.) This implies that we should have $\Psi(0) = 0$, which means that

$$\int_{\mathbb{R}} \psi(t) dt = 0.$$

So, in continuous wavelet analysis, one usually defines a wavelet to be any function whose integral is zero and that satisfies the admissibility condition. It can be shown that if $\gamma(t)$ is a function which is k times differentiable and $\gamma^{(k)} \in L^2(\mathbb{R})$, then $\psi(t) = \gamma^{(k)}(t)$ is a wavelet according to this definition. However, with such a general definition, one does not have a multiresolution analysis (MRA) to sustain the theory. For example, the Mexican hat (Example 5.3.1) and the Morlet wavelet (Example 5.3.2) do not fit into a MRA. The Morlet wavelet does only satisfy the admissibility condition approximately.

Obviously, the wavelet transform is an overcomplete representation of the signal $f(t)$ (we have a frame here). Indeed, instead of a one-dimensional function, one obtains a two-dimensional representation.

The CWT is often used to characterize the (type of) singularities of the functions $f(t)$ [11]. It can be used for example to study fractals, self-similarity etc.

For implementation on a computer, the CWT should be discretized, but this differs definitely from the FWT where one starts out with a discrete signal. Only if the $\psi(t)$ of the CWT fits into a MRA, a discretization like in the discrete case is possible, but in general, the transform remains redundant.

7.7 Overcomplete wavelet transform

To approximate the CWT, we can compute it in a subset (grid) of the time-scale plane. In the DWT, we have chosen to evaluate the CWT in the points

$$(a, b) \in \Gamma_{\mathrm{DWT}} = \{(a_n, b_{nm}) : a_n = 2^n, b_{nm} = 2^{-n}m\}.$$

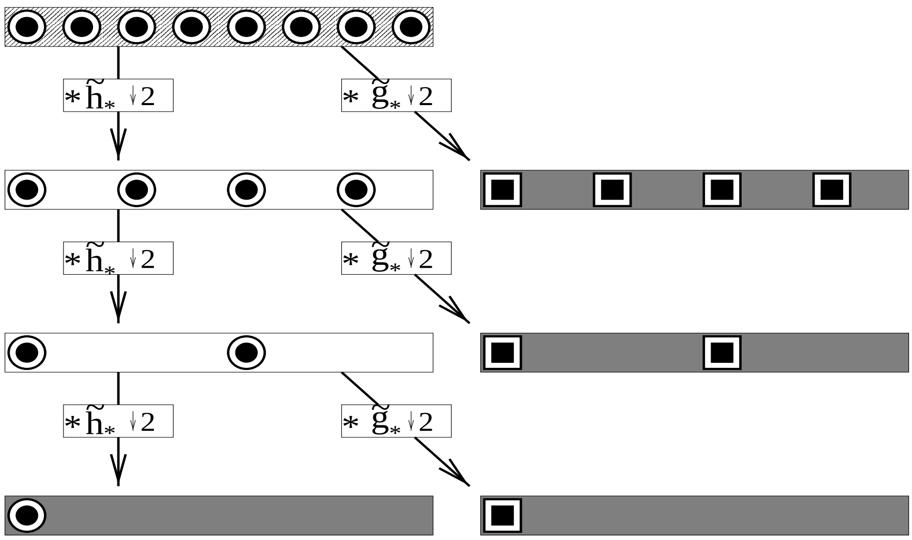
Of course we could choose a more general grid Γ. The overcomplete wavelet transform (OWT) is just the CWT but restricted to the grid Γ. Of most practical interest are of course the cases where Γ has some "regularity". For example the semilog regular sampling is related to the grid Γ_{DWT}. It is defined as $\Gamma(\Delta, a_0) = \{a_0^m\} \times \{n\Delta\}$ where $\Delta > 0$ and $a_0 > 1$. That is linearly along the time axis and exponentially along the scale axis. In general such a system will not form a basis but will be redundant. It is a frame, and thus the general treatment should be made in the context of frames. Because the nonredundancy requirement need not be satisfied, there is again much more freedom in designing the wavelet function to meet whatever condition that would be demanded by the application. The reconstruction in general frames is however not so simple as it is with a Riesz basis and the computation is more expensive. However if the restriction of sufficiently regular grids is accepted, the computations are not that much more expensive. The redundancy also makes the transform much more robust against noise. The redundant discrete wavelet transform (RWT) discussed in the next section takes the semilog regular grid $\Gamma(1, 2)$, i.e., the grid $a_n = 2^n$ and $b_{nm} = m$. It corresponds to the DWT grid without subsampling.

7.8 Redundant discrete wavelet transform

The redundant wavelet transform is the FWT that we have described, but without the subsampling. It is also called the *stationary wavelet transform* or the a trous algorithm

(Mallat). In the usual fast wavelet transform (FWT), we subsample after each filtering

Figure 7.10: The fast wavelet transform



step: both the high pass and the low pass component. This is represented in Figure 7.10. Only the black circles (low pass information) and the black squares (high pass information) is kept.

In the redundant wavelet transform (RWT) however, we omit the subsampling and we keep the black *and* the white circles and squares of Figure 7.11. Note that both the low pass and the high pass transform has as many coefficients as the the part that as transformed. To obtain results that are consistent with the results of the FWT, in the sense that the black circles and squares in the RWT are the same as the ones obtained in the FWT, we have to upsample the filters and use $(\uparrow 2)\tilde{h}_*$ and $(\uparrow 2)\tilde{g}_*$ in the second step, $(\uparrow 2)(\uparrow 2)\tilde{h}_*$ and $(\uparrow 2)(\uparrow 2)\tilde{g}_*$ in the third step etc. In this way, we use filters with "holes", whence "a trous" algorithm. Note that the white circles and squares are not zero though, so that the RWT and the FWT gives different results.

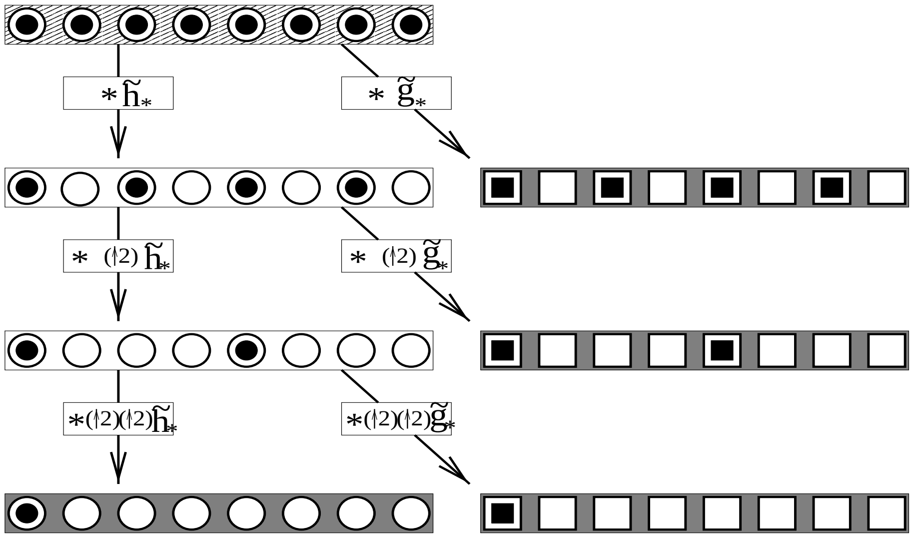
Since, in the RWT, we double the number of coefficients in each step, the memory requirements and the computer time will increase: it is $O(N \log N)$ for the RWT instead of $O(N)$ for the FWT. On the other hand, the RWT has some advantages that are not available for the FWT.

1. The RWT is translation invariant: the RWT of the translated signal is the translation of the RWT. This is not true for the FWT.

2. The RWT is immediately extendable to non dyadic inputs

3. The RWT is redundant (frame) and the reconstruction is not unique. Since the number of wavelet coefficients is doubled in each step, one can compute two independent

Figure 7.11: The redundant wavelet transform



reconstructions. If the wavelet coefficients are exact, then the two reconstructions are the same. However, if the wavelet coefficients are manipulated, then we will not have a RWT of any signal. This may be exploited however by computing the average of several of the possible inverse transforms and this has a smoothing effect, which may be advantageous for noise reduction.

Finally we recall that the wavelet coefficients in the biorthogonal FWT for a function $f \in V_N$ are given by (real case)

$$w_{nk}^{\mathrm{FWT}} = \frac{2^{n/2}}{\sqrt{2\pi}} \int_{\mathbb{R}} f(t)\tilde{\psi}(2^n t - k)dt = \left\langle \tilde{\psi}_{nk}, f \right\rangle_{L^2(\mathbb{R})}$$

so that we have for the RWT

$$w_{nk}^{\mathrm{RWT}} = \frac{2^{n/2}}{\sqrt{2\pi}} \int_{\mathbb{R}} f(t)\tilde{\psi}(2^n t - 2^{n-N}k)dt = \left\langle \tilde{\psi}_{n,2^{n-N}k}, f \right\rangle_{L^2(\mathbb{R})}.$$

This can be seen as a dyadic discretisation of the biorthogonal CWT

$$\sqrt{C_\psi}F(a,b) = \frac{1}{\sqrt{2\pi a}} \int_{\mathbb{R}} f(t)\tilde{\psi}\left(\frac{t-b}{a}\right)dt = \left\langle \tilde{\psi}_{a,b}, f \right\rangle_{L^2(\mathbb{R})}$$

with $a_n = 2^{-n}$ and $b_k = 2^{-N}k$.

7.9 Exercises

1. Compute the Fourier transform of the Haar wavelet. Show that its envelope decays like $1/\omega$. This is very slow and hence the frequency localization with the Haar wavelet

is poor.

The Shannon wavelet is in a sense the dual of the Haar wavelet. Here the wavelet itself decays like $1/t$ (in the time domain) and hence it has a poor localizing power for the t-domain. How about the localization properties in the frequency domain?

2. Prove that for the Shannon wavelet and scaling function $\varphi(t) + \psi(t) = 2\varphi(2t)$. Hint: use the filter coefficients and the dilation equations for φ and ψ.

3. Prove that the Shannon wavelets are orthogonal and that they have an infinite number of vanishing moments.

4. Give the filter coefficients (h_k) for the B-splines φ_{N-1}?

5. By using the Figure 7.11, prove that it holds for the wavelet coefficients: $w_{nk}^{\mathrm{RWT}} = w_{n,2^{n-N}k}^{\mathrm{FWT}}$.

6. (**Coiflets**) Prove that for an orthogonal wavelet with $\int t^k \varphi(t)dt = 0$ for $k = 1, \ldots, p-1$ one has
$$v_{Nk} = (2^{-N/2}/\sqrt[4]{2\pi})f(2^{-N}k) + O(2^{-N(p+1/2)})$$
where $v_{Nk} = \langle f, \varphi_{Nk}\rangle$
Hint: This is an extension of the exercise about the wavelet crime. Write $v_{Nk} = (2^{-N/2}/\sqrt{2\pi}) \int f(2^{-N}(t+k))\varphi(t)dt$ and expand $f(2^{-N}(t+k))$ in Taylor series at $t_{Nk} = 2^{-N}k$ and use partition of unity.

7. (**Coiflets**) Prove that for an orthogonal wavelet with $\int t^k \psi(t)dt = 0$ for $k = 0, 1, \ldots, p-1$ and $\int t^k \varphi(t)dt = 0$ for $k = 1, \ldots, p-1$ (Coiflets), one has, under suitable conditions for f that $\|f - \hat{f}_N\| = O(2^{-Np})$ where
$$\hat{f}_N(t) = \frac{2^{-N/2}}{\sqrt[4]{2\pi}} \sum_k f(2^{-N}k)\varphi_{Nk}(t).$$
Hint: With the previous exercise show that $\|f_N - \hat{f}_N\| = O(2^{-Np})$ where $f_N = P_n f = \sum_k \langle f, \varphi_{Nk}\rangle \varphi_{Nk}$ and use Theorem 7.2.1 to get $\|f - f_N\| = O(2^{-pN})$. Combining these gives the result.

8. Show that the RWT is indeed translation invariant, i.e., the RWT of a translated signal is a translation of the RWT. Does it hold for the CWT?

9. Prove that
$$P_j f(t) = \sum_k \langle \tilde{\varphi}_{jk}, f\rangle \varphi_{jk}(t).$$
is the projection of $f \in L_2(\mathbb{R})$ onto $V_j = \mathrm{span}\{\varphi_{jk} : k \in \mathbb{Z}\}$ parallel to $\tilde{V}_j^\perp$ with $\tilde{V}_j = \mathrm{span}\{\tilde{\varphi}_{jk} : k \in \mathbb{Z}\}$.
Hint: One has to show that $L^2(\mathbb{R}) = V_j \oplus \tilde{V}_j^\perp$, i.e., $V_j \cap \tilde{V}_j^\perp = \{0\}$ and $L^2(\mathbb{R}) = V_j + \tilde{V}_j^\perp$.

Chapter 8

Multidimensional wavelets

An image is a signal that is two-dimensional. The variable is not the time t but the variables are now the x and the y direction. For a gray-scale image, the signal itself gives the value of the "grayness" at position (x, y). One can derive a completely analoguous theory for Fourier transform, filters, wavelet basis, etc in two variables. This leads to a theory of wavelets in two variables which are in general not separable, i.e., $\psi(x, y)$ can not be written as a product $\psi_1(x)\psi_2(y)$. A much easier approach is to construct tensor product wavelets which are separable. Since this is the easiest part, we shall start with this approach.

8.1 Tensor product wavelets

A wavelet transform of a d-dimensional vector is most easily obtained by transforming the array sequentially on its first index (for all values of its other indices), then on the second etc. Each transformation corresponds to a multiplication with an orthogonal matrix. By associativity of the matrix product, the result is independent of the order in which the indices are chosen.

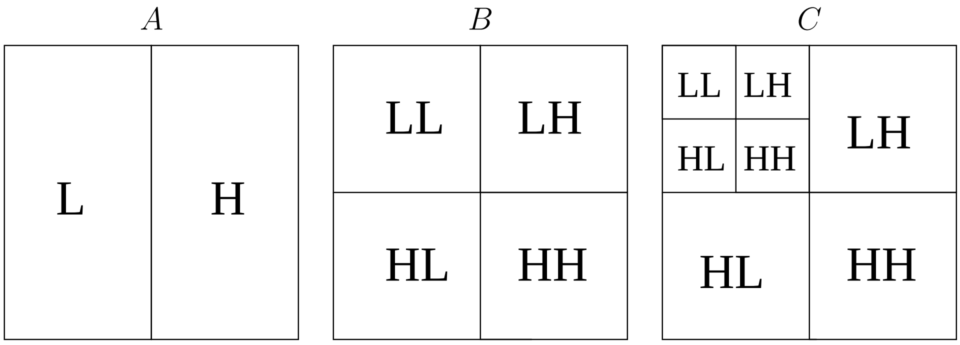
Let us consider a two-dimensional array (a square image say). First one can perform one step of the 1D transform on each of the rows of the (square) image. This results in a low resolution part L and and a high resolution part H (see Figure 8.1.A). Next, one performs one step of the 1D transforms on the columns of this result. This gives four different squares (Figure 8.1.B):

LL: low pass filtering for rows and columns
LH: low pass filtering for columns after high pass for rows
HL: high pass filtering for columns after low pass for rows
HH: high pass filtering for rows and columns

HH gives diagonal features of the image while HL gives horizontal features and LH gives vertical features.

Thus, if $\mathbf{f}$ is the (square) matrix containing the pixels of the image, and if $\tilde{\mathbf{K}} = [\tilde{\mathbf{H}} \;\; \tilde{\mathbf{G}}]$

Figure 8.1: Separable 2D wavelet transform



is the matrix associated with a single step of the 1D wavelet transform, then

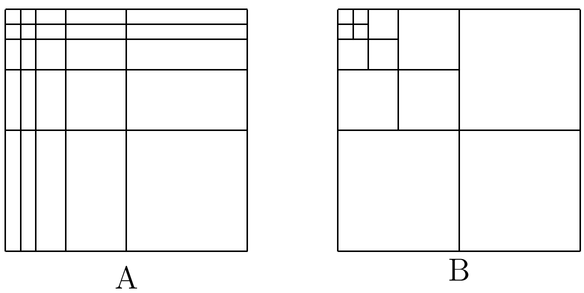
$$\mathbf{f}^1 = \begin{array}{|c|c|} \hline \text{LL} & \text{LH} \\ \hline \text{HL} & \text{HH} \\ \hline \end{array} = [\tilde{\mathbf{K}}^*]\mathbf{f}[\tilde{\mathbf{K}}^*]^t.$$

There are two possibilities to proceed:

1. the *rectangular* transform: further decomposition is performed on everything but the HH part. (Figure 8.2.A)

2. the *square* transform: further decomposition is performed on the LL part only. (Figure 8.2.B)

Figure 8.2: Different subdivisions of square



The rectangular transform corresponds to taking a 1-dimensional wavelet transform in x and y independently. Thus with a matrix $\tilde{\mathbf{K}}_1$ of smaller size, but of the same form as $\tilde{\mathbf{K}}$, we get for the second step

$$\mathbf{f}^2 = \left[\begin{array}{c|c} \tilde{\mathbf{K}}_1^* & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{I} \end{array}\right] \mathbf{f}^1 \left[\begin{array}{c|c} \tilde{\mathbf{K}}_1^* & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{I} \end{array}\right]^t$$

and similarly for all the next steps. Thus if $\mathbf{W}$ is the matrix for the complete wavelet transform of a column, then the rectangular transform for the image $\mathbf{f}$ is $\mathbf{W}\mathbf{f}\mathbf{W}^t$.

The rectangular division corresponds to setting

$$W_n = \mathrm{span}\{\psi_{nk}(x)\psi_{nl}(y) : k,l \in \mathbb{Z}\},$$

while

$$V_n = \mathrm{span}\{\varphi_{nk}(x)\varphi_{nl}(y), \varphi_{nk}(x)\psi_{nl}(y), \psi_{nk}(x)\varphi_{nl}(y) : k,l \in \mathbb{Z}\}.$$

Setting by definition

$$(h \otimes g)(x,y) = h(x)g(y),$$

this gives rise to a wavelet expansion of the form

$$f(x,y) = \sum_{m,l}\sum_{n,k} q_{m,n,k,l}(\psi_{ml} \otimes \psi_{nk})(x,y).$$

Note that the terms in this expansion give a different resolution in x- and y-direction: For the term with $\psi_{ml} \otimes \psi_{nk}$, we get in the x-direction the scaling 2^{-m} while in the y-direction the scaling is 2^{-n}.

In the square transform we get regions like in Figure 8.2 B or 8.1.C. At each stage, only the LL quarter is further subdivided. The second step can *not* be described by row and column operations on the image $\mathbf{f}^1$. We have to take out the LL part $\mathbf{f}^1_{LL}$ explicitly and we subdivide only this part by an operation of the form $[\tilde{\mathbf{K}}^*_1][\mathbf{f}^1_{LL}][\tilde{\mathbf{K}}^*_1]^t$ etc. This case gives subspaces V_n in the MRA which are now spanned by

$$V_n = \mathrm{span}\{\varphi_{nk}(x)\varphi_{nl}(y) : k,l \in \mathbb{Z}\} \quad \text{(LL squares)}$$

but the W_n are spanned by mixtures of basis functions which are now easy to describe:

$$W_n = \mathrm{span}\{\varphi_{nk}(x)\psi_{n,l}(y), \psi_{nk}(x)\varphi_{nl}(y), \psi_{nk}(x)\psi_{nl}(y) : k,l \in \mathbb{Z}\}.$$

The first set is for the HL quarters, the second set for the LH quarters and the last one for the HH quarters.

Note that there is now only one scaling 2^{-n} for both x- and y-direction.

It is the latter approach we shall follow below. We now have

$$\begin{aligned}
V_{n+1} &= V^{(x)}_{n+1} \otimes V^{(y)}_{n+1} \\
&= (V^{(x)}_n \oplus W^{(x)}_n) \otimes (V^{(y)}_n \oplus W^{(y)}_n) \\
&= (V^{(x)}_n \otimes V^{(y)}_n) \oplus (V^{(x)}_n \otimes W^{(y)}_n) \oplus (W^{(x)}_n \otimes V^{(y)}_n) \oplus (W^{(x)}_n \otimes W^{(y)}_n)
\end{aligned}$$

The projectors are

$$P_n f = \sum_{k,l} v_{nkl}\varphi_{nk}(x)\varphi_{nl}(y)$$

and

$$Q_n f = \sum_{k,l}[w^{(x)}_{nkl}\psi_{nk}(x)\varphi_{nl}(y) + w^{(y)}_{nkl}\varphi_{nk}(x)\psi_{nl}(y) + w^{(xy)}_{nkl}\psi_{nk}(x)\psi_{nl}(y)].$$

The coefficients are now arranged in matrices ($\mathbf{v}_n = (v_{nkl})_{k,l}$ etc.) and they are found by the recursions

$$\begin{aligned}
\mathbf{v}_{n-1} &= \tilde{\mathbf{H}}^* \mathbf{v}_n [\tilde{\mathbf{H}}^*]^t \quad \text{(LL part)} \\
\mathbf{w}_{n-1}^{(x)} &= \tilde{\mathbf{H}}^* \mathbf{v}_n [\tilde{\mathbf{G}}^*]^t \quad \text{(LH part)} \\
\mathbf{w}_{n-1}^{(y)} &= \tilde{\mathbf{G}}^* \mathbf{v}_n [\tilde{\mathbf{H}}^*]^t \quad \text{(HL part)} \\
\mathbf{w}_{n-1}^{(xy)} &= \tilde{\mathbf{G}}^* \mathbf{v}_n [\tilde{\mathbf{G}}^*]^t \quad \text{(HH part)}.
\end{aligned}$$

Note that each of these matrices is half the number of rows and half the number of columns of $\mathbf{v}_n$: each contains one fourth of the information.

As for the 1-dimensional case, the $\mathbf{v}_{n-1}$ matrices give the coarser information, while at each level, there are three $\mathbf{w}$-matrices that give the small scale information. For example, a high value in $\mathbf{w}_{n-1}^{(y)}$ indicates horizontal edges, a high value of $\mathbf{w}_{n-1}^{(x)}$ indicates vertical edges, and large $\mathbf{w}_{n-1}^{(xy)}$ indicates corners and diagonals.
The reconstruction algorithm uses

$$\mathbf{v}_{n+1} = \mathbf{H}\mathbf{v}_n\mathbf{H}^t + \mathbf{G}\mathbf{w}_n^{(x)}\mathbf{H}^t + \mathbf{H}\mathbf{w}_n^{(y)}\mathbf{G}^t + \mathbf{G}\mathbf{w}_n^{(xy)}\mathbf{G}^t.$$

Example 8.1.1. In Figure 8.3, we show an image of a house and below it, one can see the square and the rectangular transform respectively[1]. Only two steps of the transform are done. Observe the horizontal, vertical and diagonal features that can be seen in the different parts. We have used the most simple wavelet: the Haar wavelet. ◇

Example 8.1.2. To better illustrate the vertical, diagonal and horizontal aspects of the components, we have transformed the image in Figure 8.4 for 3 levels, using a Coiflet with 2 vanishing moments. ◇

8.2 Nonseparable wavelets

We shall not work out all the details, but only introduce the main ideas.

A 2D filter is still described by a convolution: Let $\mathcal{H}$ be a filter with impulse response $\mathbf{h} = (h_{k_1,k_2})$, $k_1, k_2 \in \mathbb{Z}$, then $\mathbf{g} = \mathcal{H}\mathbf{f} = \mathbf{h} * \mathbf{f}$ which is defined by

$$g_{n_1,n_2} = (\mathcal{H}\mathbf{f})_{n_1,n_2} = \sum_{k_1}\sum_{k_2} h_{k_1,k_2} f_{n_1-k_1,n_2-k_2}$$

In the z-domain, this becomes a multiplication:

$$G(\mathbf{z}) = H(\mathbf{z})F(\mathbf{z}) = \left(\sum_{k_1,k_2} h_{k_1,k_2} z_1^{-k_1} z_2^{-k_2}\right)\left(\sum_{k_1,k_2} f_{k_1,k_2} z_1^{-k_1} z_2^{-k_2}\right).$$

[1]To be precise, we have plotted in the smallest upper left corner the coefficients of $\varphi_{nl}(x)\varphi_{nk}(y)$, but for the other blocks we have plotted the *negative* of the image for better visibility. The coefficients in those blocks are small and since 0 corresponds to black and 255 corresponds to white, plotting the original transform gives (at the resolution of the printer) almost uniformly black blocks.

Figure 8.3: Image and its wavelet transform

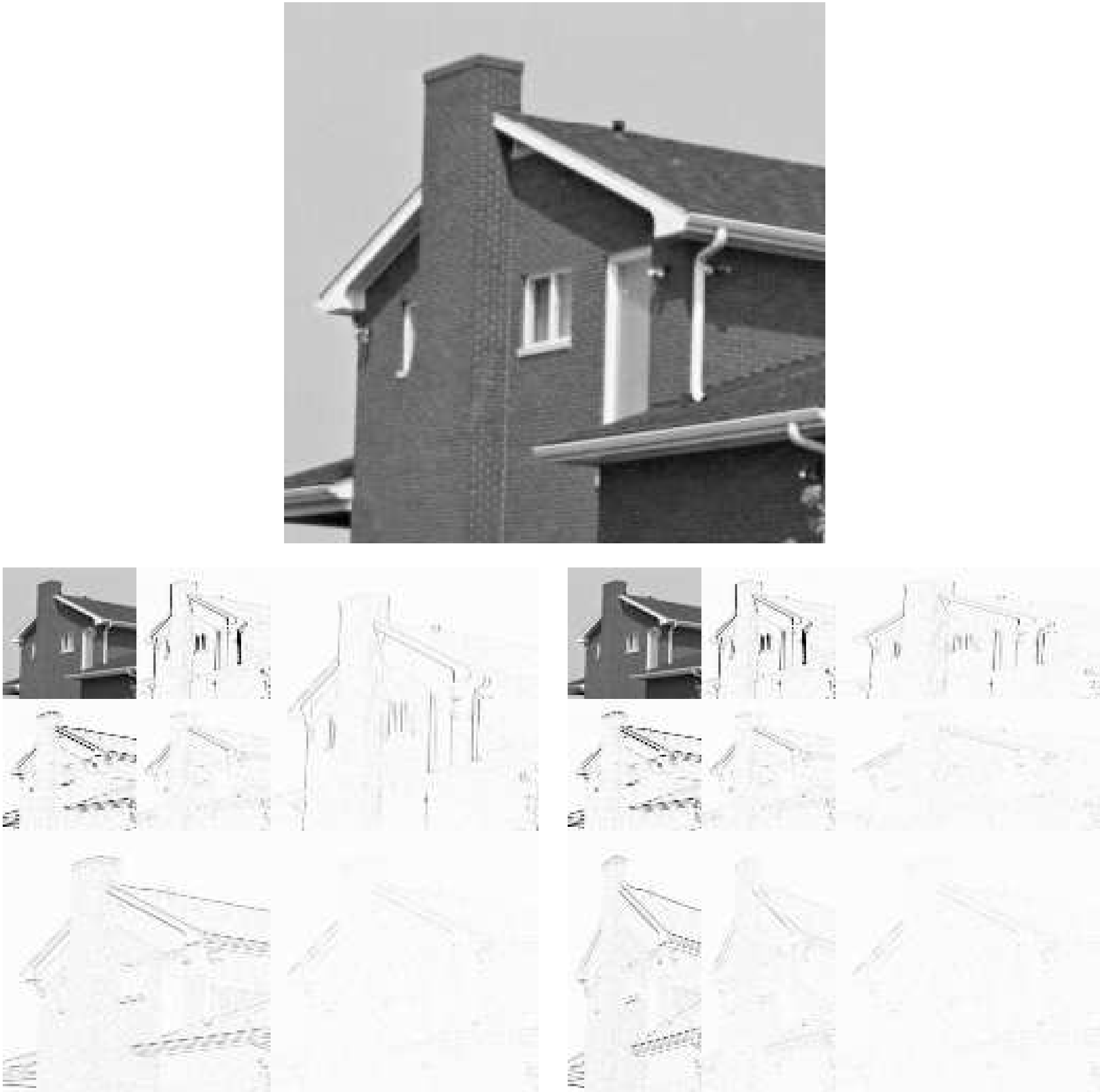
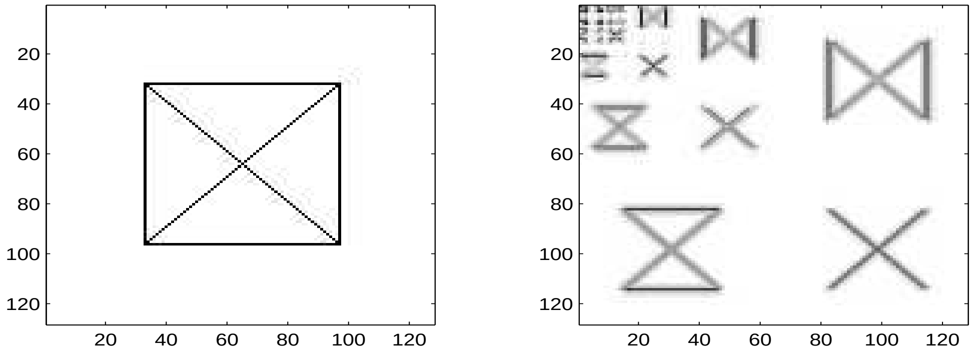


Figure 8.4: Image and its wavelet transform



In 2D the subsampling will be described by a 2×2 matrix $\mathbf{M}$. In the separable case it is diagonal, in general it is not. For example, if $\mathbf{M} = 2I_2$, then

$$(\downarrow \mathbf{M})y(n_1, n_2) = y(\mathbf{Mn}) = y(2n_1, 2n_2)$$

thus, we have an ordinary downsampling $(\downarrow 2)$ in each direction. We keep only 1 sample out of 4, thus we need 4 sublattices to cover the whole original lattice. Note that $\det \mathbf{M} = 4$.

For a quincunx filter bank, the subsampling matrix is

$$\mathbf{M}_q = \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}.$$

The subsampling operation keeps the samples for which $n_1 + n_2$ is even.

$$(\downarrow \mathbf{M}_q)y(n_1, n_2) = y(\mathbf{M}_q\mathbf{n}) = y(n_2 + n_1, n_2 - n_1)$$

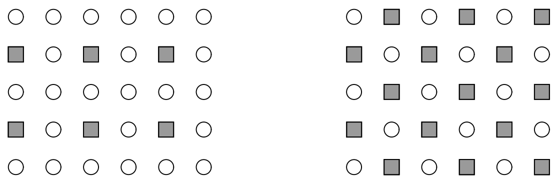
This sampling scheme keeps 1 out of 2 samples, so that we need 2 sublattices. Note that here $2 = \det \mathbf{M}_q$.

For the separable wavelets, one needs 4 channels in the filter bank (as we saw in the previous section), For the quincunx example, one needs only 2 channels. The modulation matrix for the filter bank is thus of the form

$$M(z_1, z_2) = \begin{bmatrix} H(z_1, z_2) & H(-z_1, -z_2) \\ G(z_1, z_2) & G(-z_1, -z_2) \end{bmatrix}$$

where H and G represend the low-pass and the band-pass filters involved. Like in the 1D case, the simplest solution is given by a paraunitary solution i.e., by choosing $MM_* = 2I$, so that $G(z_1, z_2) =$ an odd 2D delay of $H(-z_1^{-1}, -z_2^{-1})$.

Figure 8.5: Sampling lattices for $\mathbf{M}$ and $\mathbf{M}_q$



As for the multiresolution analysis, we can say the following. Recall the definition $H(\omega) = \sum_k h_k e^{-ik\omega}$ from the 1D MRA which represented a low-pass filter with an impulse response given by $h = (h_k)$ with $h_k = \frac{1}{\sqrt{2}} c_k$. The dilation equation was

$$\varphi(t) = \sqrt{2} \sum_k h_k \varphi(2t - k) = \sqrt{2} \langle h_*, \Phi_0(2t) \rangle_{\ell^2(\mathbb{Z})}, \quad h_* = (\bar{h}_{-k}), \quad \Phi_0(t) = (\varphi_{0k}(t)).$$

This is generalized to

$$\varphi(\mathbf{t}) = \sqrt{M} \sum_{\mathbf{k}} h_{\mathbf{k}} \varphi(\mathbf{Mt} - \mathbf{k}) = \sqrt{M} \langle h_*, \Phi_0(\mathbf{Mt}) \rangle_{\ell^2(\mathbb{Z}^2)},$$

$\mathbf{k} = (k_1, k_2)$, $\mathbf{t} = (t_1, t_2)$, $M = \det \mathbf{M}$, $h_* = (\bar{h}_{-\mathbf{k}}) \in \ell^2(\mathbb{Z}^2)$, $\Phi_0(\mathbf{t}) = (\varphi_{0\mathbf{k}}(\mathbf{t})) \in \ell^2(\mathbb{Z}^2)$ and $\varphi_{0,\mathbf{k}}(\mathbf{t}) = \varphi(\mathbf{t} - \mathbf{k})$. Note that M preserves the double integral if we change variables $\mathbf{s} = \mathbf{Mt} - \mathbf{k}$

$$M \iint \varphi(\mathbf{Mt} - \mathbf{k}) dt_1 dt_2 = \iint \varphi(\mathbf{s}) ds_1 ds_2.$$

In general there are $M - 1$ wavelet functions, defined by the equation

$$\psi^{(m)}(\mathbf{t}) = \sqrt{M} \sum_{\mathbf{k}} g_{\mathbf{k}}^{(m)} \varphi(\mathbf{Mt} - \mathbf{k}) = \sqrt{M} \left\langle g_*^{(m)}, \Phi_0(\mathbf{Mt}) \right\rangle_{\ell^2(\mathbb{Z}^2)}, \quad m = 1, \ldots, M - 1$$

where $g_*^{(m)} = (\bar{g}_{-\mathbf{k}}^{(m)})$, Φ_0 as above.

$$V_0 = \operatorname{span}\{\varphi_{0\mathbf{k}}(\mathbf{t}) = \varphi(\mathbf{t} - \mathbf{k}) : \mathbf{k} \in \mathbb{Z}^2\}$$

while the orthogonal complement is given by

$$W_0 = \operatorname{span}\{\psi_{0\mathbf{k}}^{(m)}(\mathbf{t}) = \psi^{(m)}(\mathbf{t} - \mathbf{k}) : m = 1, \ldots, M - 1; \mathbf{k} \in \mathbb{Z}^2\}.$$

Taking all the dilates and translates of the wavelets

$$\psi_{n\mathbf{k}}^{(m)}(\mathbf{t}) = M^{n/2} \psi^{(m)}(\mathbf{M}^n \mathbf{t} - \mathbf{k}),$$

we obtain an orthonormal basis for the whole space $L^2(\mathbb{R}^2)$.

8.3 Examples of 2D CWT wavelets

Some of the wavelet functions used in CWT are easily generalized to the 2D case.

8.3.1 The 2D Mexican hat

The 2D Mexican hat function is defined as

$$\psi(\mathbf{t}) = (\|\mathbf{t}\|^2 - 2) \exp\left(-\frac{1}{2}\|\mathbf{t}\|^2\right).$$

This is an obvious generalization of the 1D case. It is real and rotation invariant. It is in fact obtained by rotating the 1D function around the vertical-axis.

There does exist an anisoropic version when $\mathbf{t}$ is replaced by $A\mathbf{t}$ where $A = \mathrm{diag}(\epsilon^{-1/2}, 1)$ with $\epsilon > 1$.

8.3.2 The 2D Morlet wavelet

The definition is

$$\psi(\mathbf{t}) = \exp(i\mathbf{k}^T\mathbf{t}) \exp\left(-\frac{1}{2}\|A\mathbf{t}\|^2\right).$$

The vector $\mathbf{k}$ is the wave vector and the diagonal A matrix is the anisotropy matrix. It plays the same role as explained above for the Mexican hat function. Like in the 1D case, it does not satisfy the admissibility condition exactly, but the error is negligible for $\|\mathbf{k}\| > 5.6$. The modulus is a Gaussian, while the phase is constant along directions orthogonal to $\mathbf{k}$. It smooths in all directions, but detects sharp transitions in directions perpendicular to $\mathbf{k}$.

Chapter 9

Subdivision, second generation wavelets and the lifting scheme

Although the *lifting scheme*, which is an efficient computational scheme to compute wavelet transforms can be derived directly from the polyphase matrix, it can be applied in much more general situations than the classical wavelet filter banks. Therefore, we introduce it via an alternative approach to wavelets (subdivision schemes) which will give a framework in which it is easier to consider more generale situations. These more general wavelets are often referred to as *second generation wavelets*. This approach to the lifting scheme has the advantage that it places this computational scheme in the context where it was discovered. For simplicity we work with real data and real filters.

The contents of this chapter is extensively described in the papers [23] and [8].

Instead of starting with the most general subdivision schemes, we give first the example of the Haar wavelet transform and illustrate how the lifting scheme operates in this simple case.

9.1 In place Haar transform

We consider the unnormalized scaling functions $\varphi_{nk}(t) = 2^n\varphi(2^n t - k)$ with $\varphi(t) = \chi_{[0,1)}(t)$ the box function and the wavelets $\psi_{nk}(t) = \psi(2^n t - k)$ with $\psi(t) = \varphi(2t) - \varphi(2t-1)$. So the filter coefficients are $(h_0, h_1) = (1/2, 1/2)$ and $(g_0, g_1) = (1, -1)$. Thus the moving average and moving difference filters give the following computation of scaling and wavelet coefficients in the FWT

$$v_{nk} = \frac{v_{n+1,2k+1} + v_{n+1,2k}}{2} \quad \text{and} \quad w_{nk} = v_{n+1,2k+1} - v_{n+1,2k}.$$

Although the v_{nk} and w_{nk} take the same amount of memory to store, we cannot write v_{nk} and w_{nk} immediately in the place of $v_{n+1,k}$ because one needs the $v_{n+1,2k+1}$ and $v_{n+1,2k}$ until both v_{nk} and w_{nk} are computed. However, we can rearrange the computations as follows. First compute $w_{nk} = v_{n+1,2k+1} - v_{n+1,2k}$, then it is easily seen that $v_{nk} = v_{n+1,2k} + w_{nk}/2$. Thus, once w_{nk} has been computed, we do not need to keep $v_{n+1,2k+1}$ and we can replace it with w_{nk}. After v_{nk} has been computed, it can be stored in the place of $v_{n+1,2k}$. Thus, we do not really need the index n and we can simply write

$$v_{2k+1} \leftarrow v_{2k+1} - v_{2k} \quad \text{and} \quad v_{2k} \leftarrow v_{2k} + v_{2k+1}/2.$$

The inverse transform is

$$v_{2k} \leftarrow v_{2k} - v_{2k+1}/2 \quad \text{and} \quad v_{2k+1} \leftarrow v_{2k+1} + v_{2k}.$$

Note that this is just a matter of changing signs.

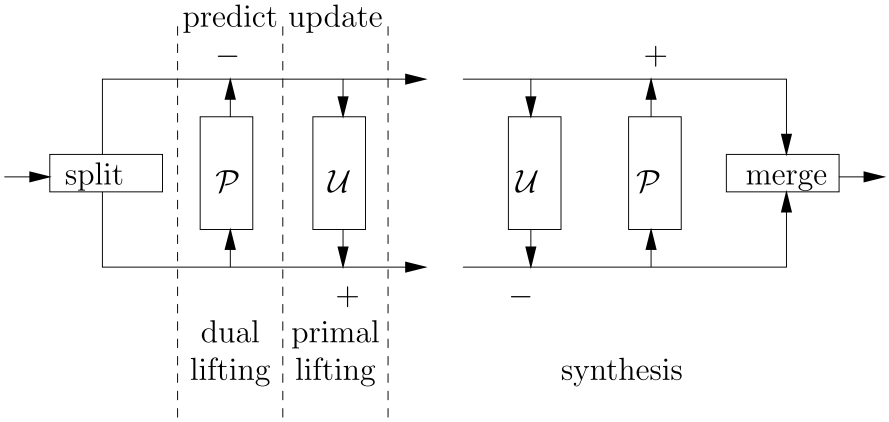
This in-place Haar transform is a very simple case of a lifting scheme. Such a scheme consists of essentially three steps

1. *Splitting*: The data are split into two sets say O and E (in this case the even and the odd samples).

2. *Dual lifting = Prediction*: The data from set O are predicted (by some operator – i.e. filter – $\mathcal{P}$) from the data in the other set and only the difference, i.e., the prediction error is kept: $O \leftarrow \mathcal{P}(E) - O$. In our example the odd samples are predicted by the even samples by the trivial operator which predicts v_{2k+1} by v_{2k}. The difference $v_{2k+1} - v_{2k}$ is stored, instead of the odd samples. The odd samples can always be recovered from this prediction error and the prediction from the even samples.

3. *Primal lifting = Updating*: Now the set E is updated (using some operator – i.e. filter – $\mathcal{U}$) by the (new) set O: $E \leftarrow E + \mathcal{U}(O)$. In our example, this is $v_{2k} \leftarrow v_{2k} + v_{2k+1}/2$, thus the operator $\mathcal{U}$ takes half of the odd samples.

Figure 9.1: Lifting scheme



Notes:

1. The splitting is sometimes called the *lazy wavelet*. The splitting into even and odd samples as we have proposed is the most logical one. Since we use the set E to predict the set O, and because we assume there is a high correlation between neighbouring samples, the even-odd splitting is the obvious one to choose.

2. What we described above is just one dual-primal pair of lifting steps. In practice, there may be several such pairs before a new splitting is done.

3. The prediction error, i.e., the set O, keeps the detail information and so the set O is like the high pass part. The other set E gives the low pass part.

4. The updating in the primal lifting step is the one that really matters. If the set E is some low resolution approximation of the original signal, then we would expect that the average remains the same in the original signal and in the low resolution approximation. In general one may impose more conditions on the moments. Since E contains half as many samples as the original signal, we expect in the example of the Haar transform that
$$\sum_k v_{n+1,k} = 2\sum_k v_{n,2k}.$$
Since $2\sum_k v_{n,2k} = 2\sum_k v_{n+1,2k} + (\sum_k v_{n+1,2k+1} - \sum_k v_{n+1,2k})$, this is precisely what is obtained.

5. There is no problem to invert the lifting scheme: just reverse the order of the lifting steps and replace a plus by a minus and conversely. The splitting operation is undone by a merging operation.

6. Suppose the filters $\mathcal{P}$ and $\mathcal{U}$ are replaced by approximations $\{\mathcal{P}\}$ and $\{\mathcal{U}\}$, for example because the calculations are done in integer arithmetic, and the approximation is caused by rounding (a nonlinear filter!). The previous inversion operation is still valid. On condition that the rounding is performed consistently, then perfect inversion, i.e. PR, remains possible.

7. This scheme is equivalent with a filter bank that realizes a wavelet transform. If we feed the system with the scaling coefficients at level $n+1$, namely $v_{n+1,k}$, then the output after the lifting steps, just before the next splitting, are the wavelet coefficients w_{nk} (in the top channel) and the scaling coefficients v_{nk} (in the bottom channel).

After having illustrated the idea of lifting with this very simple example of the Haar wavelet, we will give a more systematic way of obtaining predictions for the set O from the set E and for designing updating operations. These predictions are based on subdivision schemes which are introduced in the next sections. There are two possibilities: the interpolating subdivision scheme and the averaging subdivision scheme.

To see the relation with the previous wavelet analysis, we give some general considerations first. Recall the cascade algorithm of Section 5.6.4. If we do the synthesis in the previous lifting scheme, starting at level 0 and performing an infinite number of reconstruction steps (so that in theory we know the signal at infinite precision), then, if we start with $v_{0k} = \delta_k$ and $w_{0k} = 0$, it is clear that, just like in the cascade algorithm, the resulting signal by this reconstruction will be the scaling function $\varphi(t)$ (if the algorithm converges). More generally, starting with $v_{nl} = \delta_{k-l}$ and $w_{nl} = 0$, then the resulting signal will be $\varphi_{nk}(t)$. Note that in principle the samples need not be taken in equidistant points (but for simplicity we almost always assume that the sampling points at level n are $t_{nk} = 2^{-n}k$). Assume that we start with

$v_{0k} = \delta_k$ and $w_{0k} = 0$ and that after one step of reconstruction, we get coefficients $v_{1k} = c_k$, then starting at level 1 with the data c_k results in the same $\varphi(t)$ as the one obtained by the original data $v_{0k} = \delta_k$ and $w_{0k} = 0$ at level 0. This means that we have the relation $\varphi(t) = \sum_k c_k \varphi_{1k}(t)$, thus for a set of dyadic points $t_{nk} = 2^{-n}k$, this is $\varphi(t) = \sum_k c_k \varphi(2t-k)$, so that one step of the synthesis scheme gives the coefficients of the dilation equation.

9.2 Interpolating subdivision

The following subdivision interpolation scheme is due to Deslauriers and Dubuc. Suppose we know a continuous signal $y(t)$, $t \in \mathbb{R}$ by its samples at integer points $t_{0k} = k$, $k \in \mathbb{Z}$: $y(t_{0k}) = y_{0k}$, $k \in \mathbb{Z}$. We can try to find values in between these points by interpolation. For example, defining the finer mesh $t_{1,2k} = t_{0k}$ and $t_{0,2k+1} = (t_{0,k} + t_{0,k+1})/2$, we compute the values $y_{0,2k+1}$ in the latter points by linear interpolation;

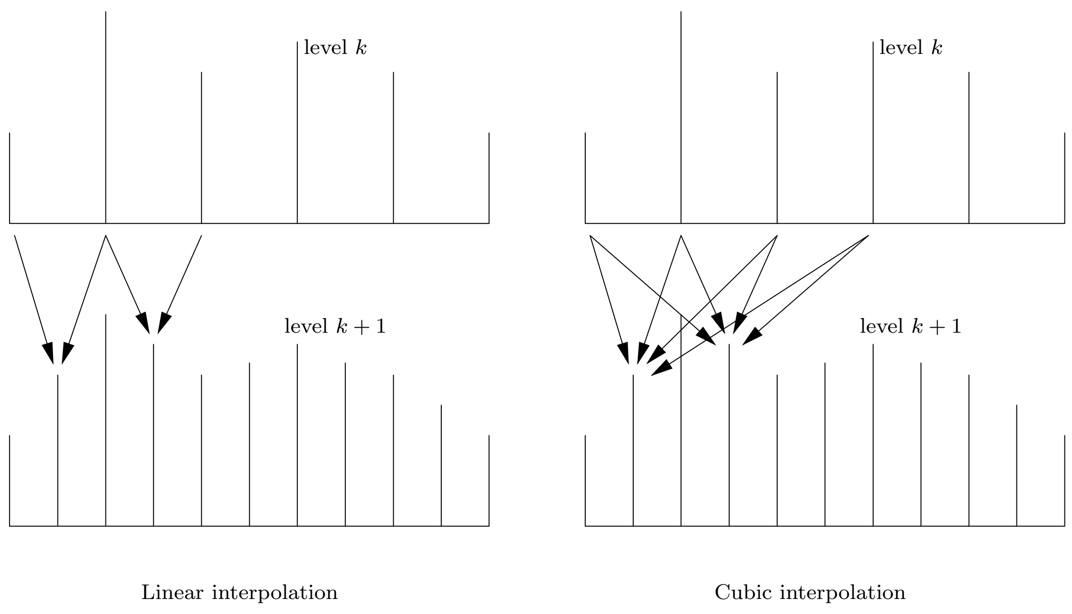
$$y_{1,2k} = y_{0,k} \quad \text{and} \quad y_{1,2k+1} = \frac{1}{2}(y_{0,k} + y_{0,k+1}), \quad k \in \mathbb{Z}.$$

This procedure can be repeated over and over to define

$$y_{n+1,2k} = y_{n,k} \quad \text{and} \quad y_{n+1,2k+1} = \frac{1}{2}(y_{n,k} + y_{n,k+1}), \quad k \in \mathbb{Z}.$$

So we get a representation at different resolution levels. As $n \to \infty$, we obtain a continuous piecewise linear function which connects the originally given points by straight lines. See Figure 9.2.

Figure 9.2: Interpolating subdivision



Of course, this is but the simplest possible case of an interpolation scheme. More generally, we take polynomials of a higher degree to interpolate. For example of odd degree $N' = N - 1$ with $N = 2D$. This N is called the order of the subdivision interpolation scheme. To define the value of y at a midpoint, we take D values to the left and D values to the right, and find the interpolating polynomial of degree N' which interpolates these points and evaluate it at the midpoint. Thus at level n, we define polynomials p_k of degree $N-1$ such that

$$p_k(t_{n,k+i}) = y_{n,k+i}, \quad i = -(D-1), \ldots, (D-1), D$$

and define the value at level $n+1$ by

$$y_{n+1,2k} = p_k(t_{n+1,2k}) = p_k(t_{n,k}) = y_{n,k}, \quad k \in \mathbb{Z}$$

while

$$y_{n+1,2k+1} = p_k(t_{n+1,2k+1}), \quad k \in \mathbb{Z}.$$

Of course, if there are only a finite number of data, we can not take symmetric points near the boundary. One can then take $2D$ points which are "as symmetric as possible". For simplicity we shall initially assume that we have infinitely many points, so that there are no boundary effects.

If we start this interpolating subdivision scheme with the impuse δ (i.e. $y_{0,0} = 1$ and $y_{0,k} = 0$ for all $k \neq 0$), then this scheme will give a continuous function, which we shall call the *scaling function* of the process. For the linear interpolating subdivision scheme, this is a hat function. The following properties are easily verified.

Theorem 9.2.1. *If $\varphi(t)$ is a scaling function of an interpolation scheme with polynomials of degree $N' = N - 1 = 2D - 1$, then*

1. *It has compact support (it is zero outside $[-N', N']$) and it is symmetric around $t = 0$.*
2. *It satisfies $\varphi(k) = \delta_k$, $k \in \mathbb{Z}$*
3. *φ and its integer translates reproduce all polynomials of degree $< N$: $\sum_k k^p \varphi(t-k) = t^p$, $0 \leq p < N$, $t \in \mathbb{R}$.*
4. *$\varphi \in C^\alpha(\mathbb{R})$ with $\alpha = \alpha(N)$ nondecreasing with N*
5. *φ satisfies a refinement equation: there are h_k such that $\varphi(t) = \sum_{j=-N}^{N} h_j \varphi(2t - j)$.*

Proof. The points (1)–(3) follow immediately by construction. For (4) we refer to [23]. Point (5) is seen as follows. When starting from level 0 with the delta function, then at level 1 this will generate values in the points $t_{1,j}$, $j = -N, \ldots, N$. Call these values h_j. It is obvious that the result after infinitely many refinement steps starting from level 0 with the delta function or starting from level 1 with the coefficients h_j will be the same. This implies the dilation equation. Note that $h_{2j} = \delta_{0,j}$, so that in particular $h_{-N} = h_N = 0$. □

Example 9.2.1. For linear interpolation the coefficients are given by $(h_{-1}, h_0, h_1) = (1/2, 1, 1/2)$. The scaling function is the hat function connecting $(-1, 0)$, (0,1) and (1,0). ◇

Define the scaled translates $\varphi_{n,k}(t) = \varphi(2^n t - k)$. Note that $\varphi_{n,k}(t)$ is obtained by starting the subdivision scheme at level n with the data δ_{i-k}, $i \in \mathbb{Z}$.

Corollary 9.2.2. *The functions $\varphi_{n,k}(t)$ for a subdivision scheme with filter coefficients (h_j) satisfy $\varphi_{n,k} = \sum_j h_{j-2k}\varphi_{n+1,j}$.*

Proof. This follows from the refinement equation, setting $t \leftarrow 2^n t - k$ and $j \leftarrow j - 2k$. $\square$

If we define $V_n = \operatorname{span}\{\varphi_{n,k} : k \in \mathbb{Z}\}$, $n = 0, 1, \ldots$, then obviously $V_0 \subset V_1 \subset V_2 \subset \cdots$.

If $f \in V_n$, then it can be written as $f = \sum_k v_{nk}\varphi_{nk}$. Thus f is the result of a subdivision scheme starting at level n with the data $v_{n,k}$. Since $f \in V_n \subset V_{n+1}$, it can also be written as $f = \sum_i v_{n+1,i}\varphi_{n+1,i}$. We have

Theorem 9.2.3. *If the subdivision scheme has coefficients (h_j), and if*

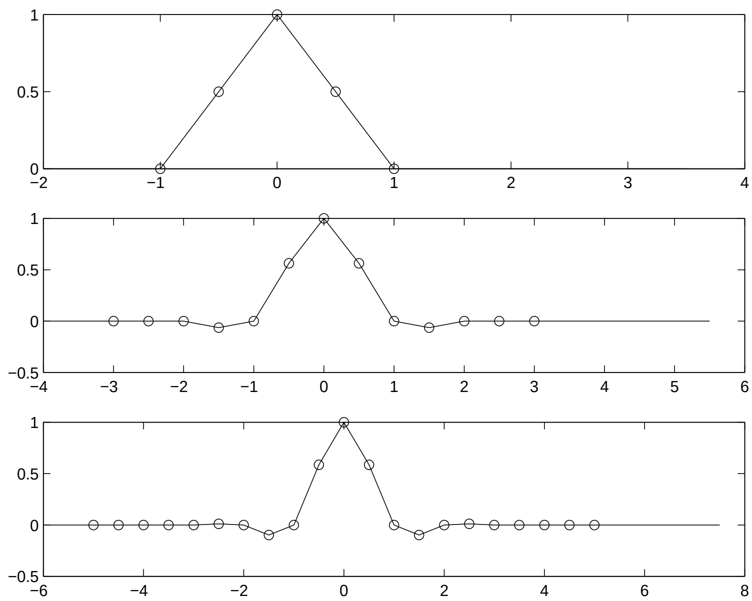
$$f = \sum_k v_{n,k}\varphi_{n,k} = \sum_i v_{n+1,i}\varphi_{n+1,i}$$

then $v_{n+1,i} = \sum_k h_{i-2k} v_{n,k}$.

Proof. This follows from the refinement equation and Corollary 9.2.2. $\square$

In Figure 9.3 we have shown the scaling functions for interpolating subdivision for degree $N' = 1, 3, 5$. The relevant filter coefficients are indicated by circles.

Figure 9.3: Interpolating subdivision scaling functions



9.3 Averaging subdivision

The following subdivision scheme is introduced by Donoho. Consider as before the points $t_{nk} = k/2^n$ for $k \in \mathbb{Z}$ which correspond to the subdivision at resolution level n. For ease of notation, we define the averaging operator

$$\mathcal{I}_{nk}\{f\} = \frac{\int_{t_{nk}}^{t_{n,k+1}} w(t)f(t)dt}{\int_{t_{nk}}^{t_{n,k+1}} w(t)dt}.$$

For simplicity of notation, we shall assume most of the time that $w \equiv 1$. Suppose we are given the averages of a function over intervals $I_{0k} = [k, k+1]$. Thus with $t_{0k} = k$, $k \in \mathbb{Z}$, we know $v_{0k} = \mathcal{I}_{0k}\{f\}$. With these data we can represent the function at this resolution level as a piecewise constant function taking the value v_{0k} in the interval I_{0k}. To compute averages on the finer mesh $t_{1,k}$, we can define a polynomial of degree 2 which satisfies $\mathcal{I}_{0,k+i}\{p\} = v_{0,k+i}$, $i = -1, 0, 1$, at level 0 and then define $v_{1,2k} = \mathcal{I}_{1,k}\{p\}$ and $v_{1,2k+1} = \mathcal{I}_{1,k+1}\{p\}$. Thus the middle interval of level n is split in half and each half gets a new average constant function value (see Figure 9.4). This process is continued so that at level n, one computes a polynomial of degree 2 such that $\mathcal{I}_{n,k+i}\{p\} = v_{n,k+i}$, $i = -1, 0, 1$, and then define $v_{n+1,2k} = \mathcal{I}_{n+1,k}\{p\}$ and $v_{n+1,2k+1} = \mathcal{I}_{n+1,k+1}\{p\}$. It is clear that if the original function was quadratic, then this scheme will regenerate the original function.

In general, one takes a polynomial of degree $N-1$ with $N = 2D+1$ odd and starts from the averages over N intervals at level n to compute the average over the middle intervals at level $n+1$. Thus in general, given $\{v_{n,k-D}, \ldots, v_{n,k+D}\}$, one constructs a polynomial p of degree $N-1$ satisfying

$$\mathcal{I}_{n,k+l}\{p\} = v_{n,k+l}, \quad -D \le l \le D$$

and computes the averages for the two central intervals at the finer level $n+1$ as

$$v_{n+1,2k} = \mathcal{I}_{n+1,2k}\{p\} \quad \text{and} \quad v_{n+1,2k+1} = \mathcal{I}_{n+1,2k+1}\{p\}.$$

See Figure 9.4.

Again, as in the previous section, we can start with the averages given by the impulse δ and apply the previous averaging subdivision scheme and the result is called the scaling function of this scheme. It has the following properties

Theorem 9.3.1. *If $\varphi(t)$ is a scaling function of an averaging scheme with polynomials of degree $N' = N-1 = 2D$, then*

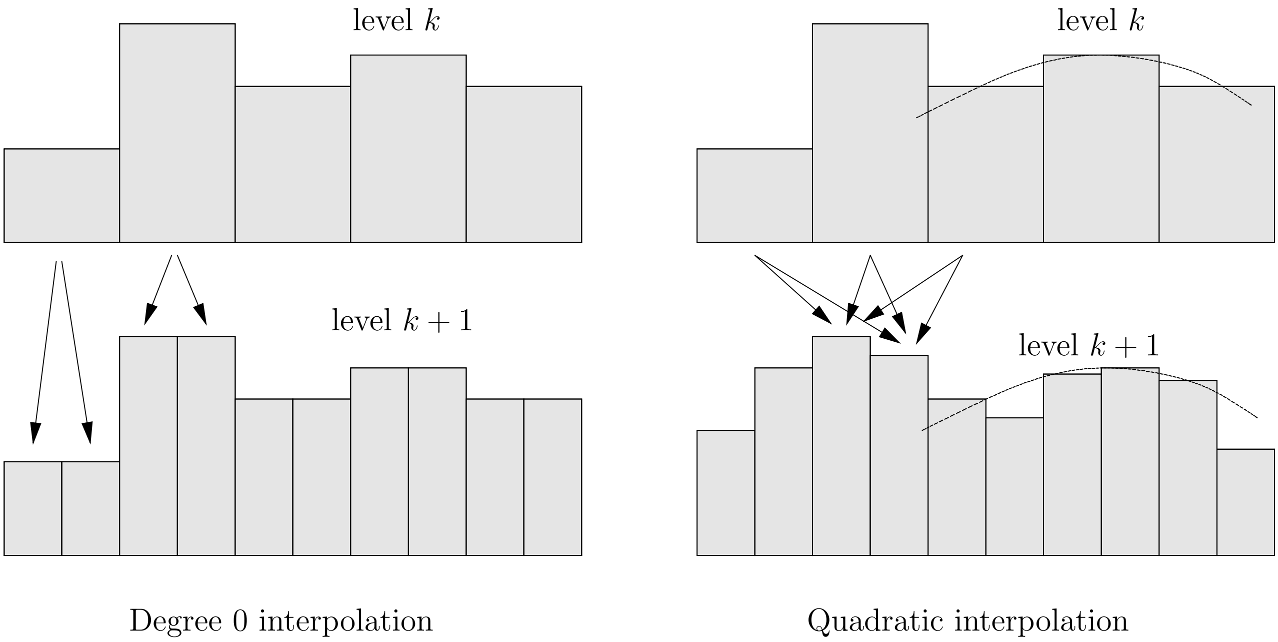
1. *It has compact support (it is zero outside $[-N', N]$) and it is symmetric around $t = 1/2$.*

2. *It satisfies $\mathcal{I}_{0k}\{\varphi\} = \delta_k$, $k \in \mathbb{Z}$*

3. *φ and its integer translates reproduce all polynomials of degree $< N$. For example, if $w \equiv 1$ then*
   $\frac{1}{p+1}\sum_k[(k+1)^{p+1} - k^{p+1}]\varphi(t-k) = t^p$, $0 \le p < N$, $t \in \mathbb{R}$.

4. *$\varphi \in C^\alpha(\mathbb{R})$ with $\alpha = \alpha(N)$ nondecreasing with N*

Figure 9.4: Averaging subdivision



5. *φ satisfies a refinement equation: there are h_k such that $\varphi(t) = \sum_{j=-N+1}^{N} h_j \varphi(2t - j)$.*

Proof. The proof is like in the case of interpolating subdivision. For example in (3), we have for $w = 1$ that $\mathcal{I}_{0k}\{p\} = \int_k^{k+1} f(t)dt$ for any polynomial f of degree p. For $f(t) = t^p$, we get the data $v_{0k} = \frac{1}{k+1}[(k+1)^{p+1} - k^{p+1}]$. Thus starting at level 0 with the data $\sum_k v_{0k}\delta(t-k)$, the subdivision scheme will end up with $\sum_k v_{0k}\varphi(t-k)$. And since every step of the subdivision reproduces polynomials of degree p, this has got to be t^p itself. The proof of (5) is along the same lines. $\square$

The construction also implies that $h_0 = h_1$ and $h_{2j} = -h_{2j+1}$ for $j \neq 0$. If we define as before $\varphi_{nk}(t) = \varphi(2^n t - k)$, then we can prove as in the previous section that $f = \sum_k v_{n,k}\varphi_{n,k} = \sum_j v_{n+1,j}\varphi_{n+1,j}$ implies $v_{n+1,j} = \sum_k h_{j-2k} v_{n,k}$.

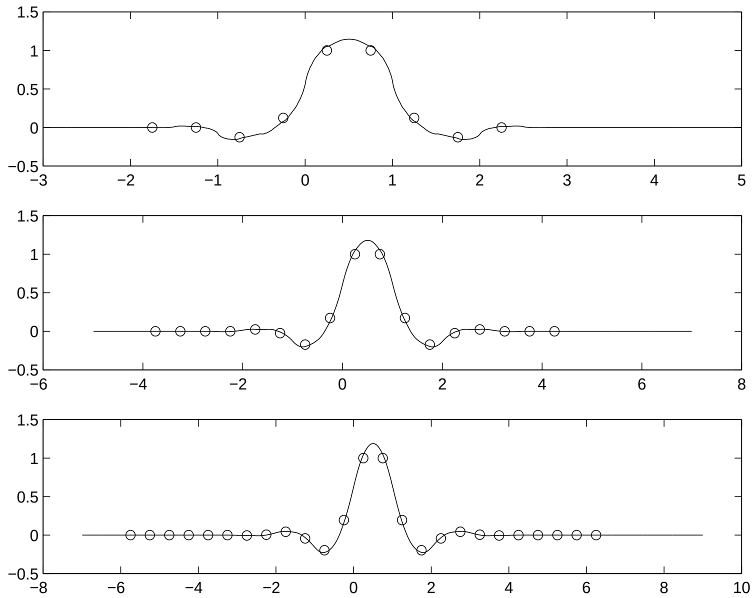
Also here some precautions should be taken near the boundary when only a finite number of data are available.

In Figure 9.5 we have shown the scaling functions for averaging subdivision for degree $N' = 2, 4, 6$. The relevant filter coefficients are indicated by circles.

9.4 Second generation wavelets

Second generation wavelets refer to situations which are generalizations of the classical case. For example, in the classical case the MRA is for the whole real line. If however there is a finite number of data, then the functions are defined in a finite interval. An image contains

Figure 9.5: Averaging subdivision scaling functions



only a finite number of pixels. At the boundary, where one can run out of data, one can use several strategies, like symmetric or asymmetric or cyclic extension of the data or more drastically: zero padding (extend it with zeros). It would be more natural if there were special basis functions near the boundary (like for spline functions for example) such that all basis functions are *only defined in the interval.* This still needs some adaptations near the boundary, but they can be obtained by the same principle in these subdivision schemes as we have seen above.

Also *irregularly spaced data* form a problem in classical analysis. For the subdivision schemes, there is again no problem to generalize the idea.

For the orthogonality, one can make use of a *weighted inner product*, like $\langle f, g\rangle = \int_0^1 f(t)g(t)w(t)dt$ in $L^2([0,1], w)$ where w is some positive weight function. Also this can be dealt with, without causing real difficulties by the subdivision schemes.

As an example, we shall treat in the next sections the MRA of real functions on the interval $[0,1]$ which are square integrable with respect to a weighted inner product as the one above.

9.5 Multiresolution

We assume now that we work with a limited number of data and we shall consider the *real* functions in $L^2 = L^2([0,1], w)$. Since the functions are real, we have $\langle f, g\rangle = \langle g, f\rangle = \int_0^1 w(t)f(t)g(t)dt$. The coarsest level is $n = 0$ and we will consider levels of inreasing resolution $n = 1, 2, \ldots$. The subdivision points at level n are denoted as $t_{n,k}$ and these could be irregularly spaced. In the interpolating subdivision scheme, these points are numbered $k = 0, 1, \ldots, k_n = 2^n$, while for the averaging subdivision scheme they are numbered $k = 0, 1, \ldots, k_n = 2^n - 1$.

A MRA for L^2 can then be defined as the set of nested spaces

$$V_n = \text{span}\{\varphi_{nk} : k = 0, 1, \ldots, k_n\}, \quad n = 0, 1, \ldots$$

where φ is the (primal) scaling function of the subdivision scheme and $\varphi_{nk}(t) = \varphi(2^n t - k)$. Clearly $V_0 \subset V_1 \subset \cdots$ and $\bigcup_{n>0} V_n$ is dense in L^2.

Let P_n denote the orthogonal projection operator onto V_n. If the basis functions $\{\varphi_{nk}\}_{k=0}^{k_n}$ are orthogonal, then of course $P_n f = \sum_k \langle \varphi_{nk}, f \rangle \varphi_{nk}$, but we shall immediately work with a biorthogonal basis of scaling functions $\tilde{\varphi}_{nk}(t) = \tilde{\varphi}(2^n t - k)$ with $\tilde{\varphi}$ some dual biorthogonal scaling function. So we assume the biorthogonality condition $\langle \varphi_{nk}, \tilde{\varphi}_{nk'} \rangle = \delta_{k-k'}$ and the normalizing condition $\int_0^1 \tilde{\varphi}_{nk}(t) w(t) dt = 1$. Note that this does not define the $\tilde{\varphi}$ uniquely. There are more solutions possible. In any case, for a given biorthogonal basis, we obtain the the oblique projection on V_n by

$$P_n f = \sum_k \langle \tilde{\varphi}_{nk}, f \rangle \varphi_{n,k}.$$

Example 9.5.1. In the subdivision schemes $P_n f(t) = \sum_k v_{nk} \varphi_{nk}(t)$.

In the interpolatory subdivision, let us take the biorthogonal basis such that $P_n f(t)$ is the function in V_n, (thus the linear combination of the φ_{nk}) that takes the values $f(t_{nk})$ in the points t_{nk}. We also know that in this scheme the φ_{nk} are functions satisfying $\varphi_{nk}(t_{nl}) = \delta_{k-l}$, and so $P_n f(t_{nk}) = f(t_{nk}) = \sum_i v_{ni} \delta_{k-i} = v_{nk}$. Thus $v_{nk} = f(t_{nk})$. On the other hand $v_{nk} = \langle \tilde{\varphi}_{nk}, f \rangle$. Hence, because this is true for all f, the dual functions $\tilde{\varphi}_{nk}$ are delta functions: $\tilde{\varphi}_{nk}(t) = \delta(t - t_{nk})$, so that $v_{nk} = \langle \tilde{\varphi}_{nk}, f \rangle = f(t_{nk})$.

In the averaging subdivision scheme, a similar derivation can be made. We assume that $v_{nk} = \langle \tilde{\varphi}_{nk}, f \rangle$ should be an average over the interval $I_{nk} = [t_{nk}, t_{n,k+1}]$, and so, $\tilde{\varphi}_{nk} = \chi_{nk}/|I_{nk}|$ with χ_{nk} the indicator function of interval I_{nk} and $|I| = \int_I w(t) dt$. Indeed, we then have

$$v_{nk} = \langle \tilde{\varphi}_{nk}, f \rangle = \int_0^1 f(t) \tilde{\varphi}_{nk}(t) w(t) dt = \mathcal{I}_{nk}\{f\}.$$

$\diamond$

We shall say that the MRA has order N if the order of the subdivision scheme is N, thus this means that if the data v_{nk} at level n correspond to a polynomial of degree $< N$, then the subdivision scheme shall synthesize this polynomial exactly. In other words $P_n t^p = t^p$ for $0 \leq p < N$.

Defining $\tilde{V}_n = \text{span}\{\tilde{\varphi}_{nk} : k = 0, \ldots, k_n\}$ and $\tilde{P}_n = \sum_k \langle \varphi_{nk}, \cdot \rangle \tilde{\varphi}_{nk}$, it should be clear that the dual scaling function $\tilde{\varphi}$ defines (at least formally) a dual MRA of L^2. However, the order $\tilde{N}$ of the dual MRA can be different from the order N of the primal MRA.

Example 9.5.2. In the linear interpolating subdivision scheme, the primal functions are hat functions and the dual functions are Dirac impulses. Therefore $\tilde{N} = 0$, while for the primal MRA, we had $N = 2$. $\diamond$

Generalizing the filter relations we derived before, we could say that if $\tilde{\varphi}_{n,k} = \sum_j \tilde{h}_{n,k,j}\tilde{\varphi}_{n+1,j}$, then because $v_{nk} = \langle\tilde{\varphi}_{nk}, f\rangle$ and $v_{n+1,k} = \langle\tilde{\varphi}_{n+1,k}, f\rangle$,

$$v_{n,k} = \sum_j \tilde{h}_{n,k,j} v_{n+1,j}.$$

On the other hand if $f \in V_n$ for all $n \geq n_0$, then $f = \sum_k v_{nk}\varphi_{nk} = \sum_j v_{n+1,j}\varphi_{n+1,j}$, $n \geq n_0$, so that using $\varphi_{nk} = \sum_j h_{n,k,j}\varphi_{n+1,j}$, we get that the coefficients at successive levels are related by filtering relations

$$v_{n+1,k} = \sum_j h_{n,j,k} v_{n,j}.$$

Note also that $\lim_{n\to\infty} v_{n,k2^{n-n_0}} = f(k2^{-n_0})$ for sufficiently smooth functions, since the average over an interval converges to the function value.

Example 9.5.3. In the interpolating subdivision scheme we had $v_{nk} = v_{n+1,2k}$. Thus $\tilde{h}_{n,k,j} = \delta_{j-2k}$. Moving to a coarser level (analysis) is obtained by subsampling the even samples.

In the averaging subdivision scheme, the dual scaling functions are normalized box functions. Thus

$$\tilde{\varphi}_{nk} = \frac{1}{|I_{nk}|}\left[|I_{n+1,2k}|\tilde{\varphi}_{n+1,2k} + |I_{n+1,2k+1}|\tilde{\varphi}_{n+1,2k+1}\right]$$

so that in this case $v_{nk} = h_{n,k,2k}v_{n+1,2k} + h_{n,k,2k+1}v_{n+1,2k+1}$ with

$$h_{n,k,2k} = \frac{|I_{n+1,2k}|}{|I_{nk}|}, \quad \text{and} \quad h_{n,k,2k+1} = \frac{|I_{n+1,2k+1}|}{|I_{nk}|}.$$

◇

9.6 The (pre-)wavelets

When a function is represented at different levels n and $n+1$, then the coarser representation $P_n f$ will loose some fine detail which was present in the finer representation $P_{n+1}f$. Assume we capture this fine detail as $(P_{n+1} - P_n)f$ which is in V_{n+1} but not in V_n. It is in a complementary space W_n which is such that we have the direct sum relation $V_{n+1} = V_n \oplus W_n$, but these spaces need not be orthogonal.

Since $\dim V_n = 2^n$ (averaging subdivision) or 2^n+1 (interpolating subdivision), it is clear that $\dim W_n = 2^n$. Thus there should be a set of wavelet functions ψ_{nk} such that they form a basis for W_n:

$$W_n = \operatorname{span}\{\psi_{nk} : k = 0, \ldots, 2^n - 1\}.$$

Of course, since $W_n \subset V_{n+1}$, there must exist coefficients $g_{n,k,j}$ such that

$$\psi_{nk}(t) = \sum_j g_{n,k,j}\varphi_{n+1,j}(t).$$

This W_n is however not completely arbitrary. Indeed, since $P_nV_{n+1} = V_n$, it follows that $P_nW_n = \{0\}$, and thus it should hold that $\langle\psi_{nk}, \tilde{\varphi}_{nl}\rangle = 0$ for all relevant k and l. Thus $W_n \perp \tilde{V}_n$.

By duality we also have a space

$$\tilde{W}_n = \text{span}\{\tilde{\psi}_{nk} : k = 0, \ldots, 2^n - 1\}$$

with $\left\langle \tilde{\psi}_{nk}, \varphi_{nl} \right\rangle = 0$, i.e., $\tilde{W}_n \perp V_n$. Moreover, we assume that the primal and dual wavelet basis is biorthogonal: $\left\langle \psi_{nk}, \tilde{\psi}_{nl} \right\rangle = \delta_{k-l}$. Let

$$\tilde{\psi}_{nk}(t) = \sum_j \tilde{g}_{n,k,j} \tilde{\varphi}_{n+1,j}(t)$$

and assume that $(P_{n+1} - P_n)f = \sum_k w_{nk}\psi_{nk}$, then the wavelet coefficients are given by $w_{nk} = \left\langle f, \tilde{\psi}_{nk} \right\rangle$ and they satisfy

$$w_{nk} = \sum_j \tilde{g}_{n,k,j} v_{n+1,j}.$$

The wavelet transform of $f \in V_n$ corresponds to the representation

$$f(t) = P_0 f(t) + \sum_{m=0}^{n-1} \sum_{k=0}^{2^m-1} w_{mk}\psi_{mk}(t),$$

which is based on $V_n = V_0 \oplus W_0 \oplus W_1 \oplus \cdots \oplus W_{n-1}$.

Example 9.6.1. Consider again the interpolating subdivision scheme. Since

$$\delta_{k-i} = \varphi_{nk}(t_{n,i}) = \sum_j h_{n,k,j}\varphi_{n+1,j}(t_{n+1,2i}) = h_{n,k,2i},$$

the refinement equation becomes

$$\varphi_{nk} = \sum_j h_{n,k,j}\varphi_{n+1,j} = \varphi_{n+1,2k} + \sum_j h_{n,k,2j+1}\varphi_{n+1,2j+1}.$$

Thus after subsampling, i.e., setting $v_{nk} = v_{n+1,2k}$, we get

$$P_n f = \sum_k v_{nk}\varphi_{nk} = \sum_k v_{n+1,2k}\varphi_{n+1,2k} + \sum_k \sum_j v_{nk} h_{n,k,2j+1}\varphi_{n+1,2j+1}.$$

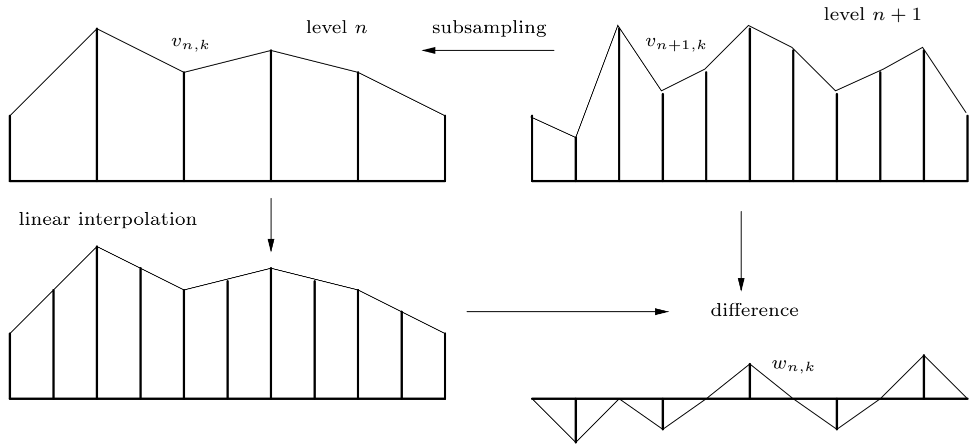
Thus the difference $P_{n+1}f - P_n f$ depends only on the odd $\varphi_{n+1,2k+1}$. Thus we may use $\psi_{nk} = \varphi_{n+1,2k+1}$. Identification of coefficients gives

$$w_{nk} = v_{n+1,2k+1} - \sum_j h_{n,j,2k+1} v_{nj}.$$

Take for example the linear interpolating subdivision scheme (i.e., $N = 2$),

$$P_n f(t) = \sum_k v_{nk}\varphi_{nk}(t), \quad v_{nk} = f(k2^{-n}) = f(t_{nk}).$$

Figure 9.6: Linear interpolation



Moving to a coarser level is obtained by subsampling, i.e., by keeping only the even samples: $v_{n,k} = v_{n+1,2k}$. The difference at the odd points is

$$w_{nk} = (P_{n+1} - P_n)f(t_{n+1,2k+1}) = v_{n+1,2k+1} - \frac{1}{2}(v_{nk} + v_{n,k+1}). \tag{9.1}$$

Note $\psi_{nk}(t) = \psi(2^n t - k)$ with $\psi(t) = \varphi(2t-1)$. See Figure 9.6. $\diamond$

Computing values at the intermediate points $v_{n+1,2k+1}$ by interpolation or averaging using the data $P_n f$ can be seen as a prediction and $Q_n f = (P_{n+1} - P_n)f$ then gives the prediction error.

Theorem 9.6.1. *If the orders of the primal and dual MRA are given by N and $\tilde{N}$ respectively, then the primal and dual wavelet functions will have $\tilde{N}$ and N vanishing moments respectively:*

$$\left\langle t^p, \tilde{\psi}_{nk} \right\rangle = 0, \quad 0 \le p < N \quad \textit{and} \quad \langle t^p, \psi_{nk} \rangle = 0, \quad 0 \le p < \tilde{N}.$$

Proof. By definition $P_n t^p = t^p$ for $0 \le p < N$ and thus $\sum_k \langle \tilde{\varphi}_{nk}, t^p \rangle \varphi_{nk} = t^p$. Because $\left\langle \varphi_{nk}, \tilde{\psi}_{nl} \right\rangle = 0$, we get after taking the inner product with $\tilde{\psi}_{nl}$ that $\left\langle t^p, \tilde{\psi}_{nk} \right\rangle = 0$ for $0 \le p < N$. The other statement is by duality. $\square$

9.7 The lifting scheme

The subdivision schemes and the corresponding "wavelets" do not really give what you would expect from a wavelet analysis.

Example 9.7.1. For example, take the interpolating subdivision scheme and assume that we have at level $n+1$ the data $1, 0, 1, 0, 1, 0, \ldots$. Taking the even samples to move to the coarser level n, would result in the constant data 1, 1, 1, ... This is of course not what we want. We would expect some averages to be maintained like

$$\int_0^1 P_{n+1}f(t)w(t)dt = \int_0^1 P_n f(t)dt.$$

This means that the zeroth moment should be maintained, thus $\tilde{N} = 1$. By Theorem 9.6.1, this means that it should hold that $\int_0^1 \psi_{nk}(t)w(t)dt = 0$. $\diamond$

This is remedied by modifying the wavelets:

$$\psi_{nk} = \psi_{nk}^o + \sum_j s_{n,j,k}\varphi_{nj}$$

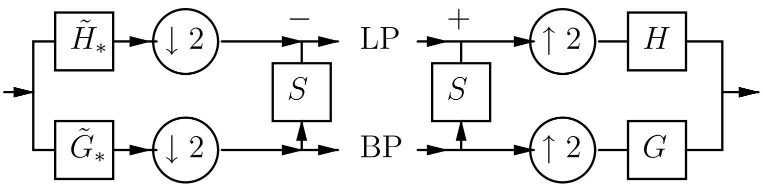
with ψ_{nk}^o the old wavelets. To keep the biorthogonality, this requires a modification of the dual scaling functions as well:

$$\tilde{\varphi}_{nk} = \tilde{\varphi}_{n,k}^o - \sum_i s_{n,k,i}\tilde{\psi}_{ni},$$

where $\tilde{\varphi}_{n,k}^o$ are the old scaling functions.

A general (primal) lifting step is represented in Figure 9.7. This is to be interpreted as

Figure 9.7: Primal lifting step



follows: First the signal is split into a low pass and a band pass part by the (old) filters $\tilde{H}_*$ and $\tilde{G}_*$ respectively. This corresponds to computing the V_n and the W_n part of a signal in V_{n+1}. The results are then subsampled, to remove redundancy. Next a (primal) lifting step is executed with the filter S (filter coefficients $s_{n,k,i}$). At the synthesis side, the same operations are undone in opposite order to obtain PR.

Example 9.7.2. Consider again the interpolating subdivision scheme. As we know, this gives $\tilde{N} = 0$ because the dual $\tilde{\varphi}_{nk}$ are delta functions. Now we modify the pre-wavelet $\psi_{nk} = \varphi_{n+1,2k+1}$ as

$$\psi_{nk} = \varphi_{n+1,2k+1} + A_{nk}\varphi_{nk} + B_{nk}\varphi_{n,k+1}$$

with $A_{nk} = s_{n,k,k}$ and $B_{nk} = s_{n,k+1,k}$ such that we have 2 vanishing moments:

$$\int_0^1 w(t)\psi_{nk}(t)dt = 0 = \int_0^1 tw(t)\psi_{nk}(t)dt.$$

Then $\tilde{N} = 2$. Because $V_{n+1} = V_n \oplus W_n$,

$$\sum_{k=0}^{2^{n+1}} v_{n+1,k}\varphi_{n+1,k} = \sum_{k=0}^{2^n} v_{nk}\varphi_{nk} + \sum_{k=0}^{2^n-1} w_{nk}\psi_{nk},$$

we get after substituting the new definition of the wavelets that

$$v_{nk} = v_{n+1,2k} - A_{nk}w_{nk} - B_{nk}w_{n,k-1}.$$

If for example we consider linear interpolation, then $A_{nk} = B_{nk} = -1/4$. The result is a wavelet with $N = \tilde{N} = 2$ and it is identical to the biorthogonal CDF(2,2) wavelet. It is one of the most popular wavelets used in applications.

In terms of filter coefficients we have: H has filter coefficients δ_k, since $v_{nk} = v_{n+1,2k}$. Thus also $\tilde{H}_*(z) = 1$ (alternating flip does not do anything). $\tilde{G}_*$ has filter coefficients $(-1/2, 1, -1/2)$, as can be seen from (9.1) which corresponds to $\tilde{\psi}_{nk} = \tilde{\varphi}_{n+1,2k+1} - \frac{1}{2}\tilde{\varphi}_{n+1,2k} - \frac{1}{2}\tilde{\varphi}_{n+1,2k+2}$. Thus $\tilde{H}_*(z) = 1$, $\tilde{G}_*(z) = -\frac{1}{2z} + 1 - \frac{z}{2}$ and S has filter coefficients $(-1/4, -1/4)$ so that $S(z) = \frac{1}{4} + \frac{1}{4z}$. The computations are thus reduced to the two formulas

$$w_{nk} = v_{n+1,2k+1} - \frac{1}{2}(v_{n+1,2k} + v_{n+1,2k+2}) \tag{9.2}$$

$$v_{nk} = v_{n+1,2k} + \frac{1}{4}(w_{nk} + w_{n,k-1}). \tag{9.3}$$

The first one is the filtering with $\tilde{G}_*$ and the second one is the lifting step.

The analysis algorithm for the interpolating subdivision is thus given by

for $n = m-1(-1)0$
  for $k = 0(1)2^n$: $v_{nk} = v_{n+1,2k}$
  for $k = 0(1)2^n - 1$: $w_{nk} = v_{n+1,2k+1} - \sum_i h_{n,i,2k+1}v_{ni}$
  for $k = 0(1)2^n$: $v_{nk} = v_{nk} + A_{nk}w_{nk} + B_{n,k-1}w_{n,k-1}$
endfor

and the synthesis is obtained by simple inversion:

for $n = 0(1)m-1$
  for $k = 0(1)2^n$: $v_{nk} = v_{nk} - A_{nk}w_{nk} - B_{n,k-1}w_{n,k-1}$
  for $k = 0(1)2^n - 1$: $v_{n+1,2k+1} = w_{n,k} + \sum_i h_{n,2k+1,i}v_{ni}$
  for $k = 0(1)2^n$: $v_{n+1,k} = v_{n,k}$
endfor

◇

A similar derivation can be given for the averaging subdivision scheme (see [23]).

What we have described here is in fact a *primal lifting step* because it modifies the *primal* functions ψ_{nk}. We could also modify the dual wavelets and hence also the primal scaling functions φ_{nk}. This would correspond to a *dual lifting step*. Such a dual lifting is described by the following change of basis

$$\tilde{\psi}_{nk} = \tilde{\psi}^o_{nk} - \sum_j t_{n,j,k} \tilde{\varphi}_{nj}$$

with $\tilde{\psi}^o_{nk}$ the old dual wavelets, and correspondingly

$$\varphi_{nk} = \varphi^o_{n,k} + \sum_i t_{n,k,i} \psi_{ni},$$

where $\varphi^o_{n,k}$ are the old scaling functions. Hence this corresponds to the additional transformation step

$$w_{nk} = w^o_{nk} - \sum_j t_{n,j,k} v_{nj}.$$

The diagram of a dual lifting is like in Figure 9.7 with S replaced by T and the arrow of that T-branch pointing in the other direction.

The advantage of the lifting steps is that they are conceptually very simple and extremely easy to invert, but they are also more efficient in general. For example the formulas (9.2)-(9.3) require 4 additions and some binary shifting. If these formulas were written explicitly by substituting (9.2) in (9.3), this would give

$$v_{nk} = -\frac{1}{8} v_{n+1,2k-2} + \frac{1}{4} v_{n+1,2k-1} + \frac{3}{4} v_{n+1,2k} + \frac{1}{4} v_{n+1,2k+1} - \frac{1}{8} v_{n+1,2k+2},$$

and this, together with (9.2), would require 6 additions and a multiplication with 3.

This observation makes it worth thinking about the idea of writing any filter bank as a succession of (elementary) primal and dual lifting steps. If we start from the polyphase representation (see Figure 4.5), then this question has a solution if we can write the polyphase matrix $P(z)$ as a product of matrices of the form

$$P(z) = \begin{bmatrix} H_e(z) & G_e(z) \\ H_o(z) & G_o(z) \end{bmatrix} = \begin{bmatrix} 1 & S_1(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ T_1(z) & 1 \end{bmatrix} \cdots \begin{bmatrix} 1 & S_m(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ T_m(z) & 1 \end{bmatrix}.$$

If all the filters involved are FIR, then all the functions mentioned should be Laurent polynomials in z.

It turns out that such a factorization is exactly what is obtained by the Euclidean algorithm, and thus such a factorization in elementary lifting steps is always possible.

The part of the scheme of Figure 4.5 which comes in front of the $\tilde{P}_*$-part, just takes the even (top branch) and the odd (bottom branch) samples of the signal. This transform does not do much and it is therefore often referred to as the *lazy wavelet transform*. Note that this corresponds to the filters $\tilde{H}_*(z) = 1$ and $\tilde{G}_*(z) = z$ and thus to a polyphase matrix $P = I$.

9.8 The Euclidean algorithm

For polynomials, the Euclidean algorithm computes a continued fraction expansion for the ratio of two polynomials a_0 and b_0 by recursive division:

$$\begin{aligned}
\frac{a_0}{b_0} &= q_0 + \frac{r_0}{b_0}, \quad \text{set } r_0 = b_1 \text{ and } b_0 = a_1 \\
&= q_0 + \frac{1}{\frac{a_1}{b_1}} \\
&= q_0 + \frac{1}{q_1 + \frac{r_1}{b_1}} \quad \text{set } r_1 = b_2 \text{ and } b_1 = a_2 \\
&= q_0 + \frac{1}{q_1 + \frac{1}{\frac{a_2}{b_2}}} \\
&= \cdots \\
&= q_0 + \frac{1 \rfloor}{\lceil q_1} + \frac{1 \rfloor}{\lceil q_2} + \cdots
\end{aligned}$$

where q_k is the quotient and r_k the remainder for the division of a_k by b_k:

$$\begin{cases} a_k - b_k q_k = r_k = b_{k+1} \\ b_k = a_{k+1} \end{cases}, \quad k = 0, 1, \ldots$$

or in short hand $a_{k+1} = b_k$ and $b_{k+1} = a_k \div b_k$ where $\div$ denotes the polynomial part of the ratio. Thus

$$\begin{bmatrix} a_{k+1} & b_{k+1} \end{bmatrix} = \begin{bmatrix} a_k & b_k \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & -q_k \end{bmatrix}, \quad k = 0, 1, \ldots$$

and this can be continued until $b_n = 0$. In that case a_n will be a greatest common divisor of a_0 and b_0.

This algorithm works in any Euclidean domain: for example, it can be used to compute a GCD of two integers in $\mathbb{Z}$ etc. The GCD is not uniquely defined. It is only fixed up to a unit, that is an invertible element. In the set of integers, the only units are $+1$ and -1 because the inverse of these numbers is again an integer. For the polynomials, the set of units are the constant plynomials.

The algorithm will always lead to a GCD because it will always end. This can be seen as follows: Define $|a|$ to be the absolute value if a is an integer or let it be the degree if a is a polynomial. Then the division is so defined that in the relation of quotient-remainder $a = bq + r$, the quotient q and remainder r satisfy $|r| < |q|$, where we define $|0| = -\infty$.

We can make a Euclidean domain out of the set of Laurent polynomials as well. The only thing we need is the definition of a quotient-remainder relation. We define $|a| = u - l$ if $a(z) = \sum_{k=l}^{u} p_k z^k$ with $p_u p_l \neq 0$. We shall define a quotient $q = a \div b$ and a remainder $r = a - qb$ if they satisfy $a = bq + r$ with $|r| < |b|$. Note however that quotient and remainder are not uniquely defined. For example, in $\mathbb{Z}$ one can write $5 = 2 \cdot 2 + 1$ thus $5 \div 2 = 2$, remainder 1, but one could as well have written $5 = 2 \cdot 3 + (-1)$, thus $5 \div 2 = 3$, remainder -1. For Laurent polynomials, there is even much more freedom. We illustrate this with an example.

Example 9.8.1. Consider the Laurent polynomials

$$a(z) = z^{-2} + 2z + 3z^2 \quad \text{and} \quad b(z) = z^{-1} + z.$$

Since $|b| = 2$, we have to find a Laurent polynomial $q(z)$ such that in $|a - bq| < 2$. Thus there may only remain at most two successive nonzero coefficients in the result. Setting

$$q(z) = q_{-2}z^{-2} + q_{-1}z^{-1} + q_0 + q_1 z + q_2 z^2$$

(other possibilities do not lead to a solution), we see that the remainder is in general

$$r(z) = (a - bq)(z) = r_{-3}z^{-3} + r_{-2}z^{-2} + r_{-1}z^{-1} + r_0 + r_1 z + r_2 z^2 + r_3 z^3$$

with

$$\begin{aligned}
r_{-3} &= q_{-2} \\
r_{-2} &= 1 - q_{-1} \\
r_{-1} &= -q_{-2} - q_0 \\
r_0 &= -q_{-1} - q_1 \\
r_1 &= 2 - q_0 - q_2 \\
r_2 &= 3 - q_1 \\
r_3 &= -q_2
\end{aligned}$$

Now one can choose to keep the successive coefficients r_k and r_{k+1}, for some $k \in \{-3, \ldots, 2\}$ and make all the others equal to zero. This corresponds to a system of 5 linear equations in 5 unknowns. Possible solutions are therefore

$$\begin{array}{ll}
q(z) = -2z^{-2} - 3z^{-1} + 2 + 3z & r(z) = -2z^{-3} + 4z^{-2} \\
q(z) = -3z^{-1} + 2 + 3z & r(z) = 4z^{-2} - 2z^{-1} \\
q(z) = z^{-1} + 2 + 3z & r(z) = -2z^{-1} - 4 \\
q(z) = z^{-1} + 3z & r(z) = -4 + 2z \\
q(z) = z^{-1} - z & r(z) = 2z + 4z^2 \\
q(z) = z^{-1} - z + 2z^2 & r(z) = 4z - 2z^2.
\end{array}$$

◇

In general, if

$$a(z) = \sum_{k=l_a}^{u_a} a_k z^k \quad \text{and} \quad b(z) = \sum_{k=l_b}^{u_b} b_k z^k$$

then we have

$$q(z) = \sum_{k=l_q}^{u_q} q_k z^k$$

with $u_q = u_a - l_b$ and $l_q = l_a - u_b$ so that $|q| = |a| + |b|$. The quotient has $|a| + |b| + 1$ coefficients to be defined. For the product bq we have $|qb| = |a| + 2|b|$, thus it has $|a| + 2|b| + 1$

coefficients. Thus also $a - bq$ has that many coefficients. Since at most $|b|$ subsequent of these coefficients may be arbitrary and all the others have to be zero, it follows that there are always $|a| + |b| + 1$ equations for the $|a| + |b| + 1$ unknowns. When these coefficients are made zero in $a - bq$ then there remain at most $|b|$ successive coefficients which give the remainder r.

We can conclude that the quotient and remainder always exists and thus we do have a Euclidean domain. The units are the monomials, i.e., Laurent polynomials of the form cz^k. Therefore we can apply the Euclidean algorithm and obtain a greatest common divisor which will be unique up to a unit factor. It is remarkable that, with all the freedom we have at every stage of the Euclidean algorithm, we will always find the same greatest common divisor up to a monomial factor.

Assume that P and $\tilde{P}$ are the polyphase and the dual polyphase matrix of a PR filter bank with FIR filters. Thus $P(z)\tilde{P}_*(z) = I$. This implies that $\det P(z)$ should be a monomial. Assume without loss of generality that it is normalized such that $\det P(z) = 1$. (Hence also $\det \tilde{P}_*(z) = 1$ and $\det \tilde{P}(z) = 1$.) With this normalization we say that the underlying filters G and H are *complementary*.

Primal and dual lifting are now caught in the following theorem, which says that lifting steps transforms a pair of complementary filters into another pair of complementary filters.

Theorem 9.8.1 (lifting). *Let (G, H) be a couple of complementary filters, then*

1. *(G', H) will be another couple of complementary filters iff G' is of the form $G'(z) = G(z) + H(z)S(z^2)$ with $S(z)$ a Laurent polynomial.*

2. *(G, H') will be another couple of complementary filters iff H' is of the form $H'(z) = G(z) + H(z)T(z^2)$ with $T(z)$ a Laurent polynomial.*

Proof. If P is a normalized polyphase matrix for complementary FIR filters, then

$$P'(z) = P(z) \begin{bmatrix} 1 & S(z) \\ 0 & 1 \end{bmatrix} \quad \text{and} \quad P'(z) = P(z) \begin{bmatrix} 1 & 0 \\ T(z) & 1 \end{bmatrix}$$

will also be normalized polyphase matrices for complementary FIR filters for any choice of the Laurent polynomials S and T. In the first case for example, the even and the odd part of $H(z)S(z^2)$ is indeed $H_e(z)S(z)$ and $H_o(z)S(z)$. □

Note that the PR condition requires that if a dual lifting is applied on the synthesis side, i.e.,

$$P'(z) = P(z) \begin{bmatrix} 1 & S(z) \\ 0 & 1 \end{bmatrix},$$

then the analysis side should be given by

$$\tilde{P}'_*(z) = \begin{bmatrix} 1 & -S(z) \\ 0 & 1 \end{bmatrix} \tilde{P}_*(z).$$

Now we can describe the factorization algorithm. Suppose we start with a filter $H(z) = H_e(z^2) + zH_o(z^2)$ and some other complementary filter G. The Laurent polynomials H_e and

H_o are coprime. Indeed, if they were not, then they would have a nontrivial common divisor which would divide the first column in $P(z)$, thus also divide $\det P(z)$, but we assumed that $\det P(z) = 1$, so this is impossible. The Euclidean algorithm will thus compute a greatest common divisor which we can always assume to be a constant, say K. This leads to

$$[H_e(z) \quad H_o(z)]V_1(z)\cdots V_n(z) = [K \quad 0]$$

with the V_k matrices of the form

$$V_k(z) = \begin{bmatrix} 0 & 1 \\ 1 & -q_k(z) \end{bmatrix}$$

where $q_k(z)$ are Laurent polynomials. After inverting and transposing, this reads

$$\begin{bmatrix} H_e(z) \\ H_o(z) \end{bmatrix} = W_1(z)\cdots W_n(z) \begin{bmatrix} K \\ 0 \end{bmatrix}$$

where the matrices $W_k(z)$ are given by

$$W_k(z) = [V_k(z)]^{-T} = \begin{bmatrix} q_k(z) & 1 \\ 1 & 0 \end{bmatrix}.$$

We can always assume that n is even. Indeed, if it were odd, we can multiply the filter H with z and the filter G with z^{-1}. They would still be complementary since the determinant of $P(z)$ does not change. This would interchange the role of H_e and H_o which would introduce some "dummy" V_0 which does only interchange these two Laurent polynomials.

Let $G^c(z)$ be a filter which is complementary to $H(z)$ for which G_e^c and G_o^c are defined by

$$P^c(z) = \begin{bmatrix} H_e(z) & G_e^c(z) \\ H_o(z) & G_o^c(z) \end{bmatrix} = W_1(z)\cdots W_n(z) \begin{bmatrix} K & 0 \\ 0 & K^{-1} \end{bmatrix}.$$

Because

$$\begin{bmatrix} q_k(z) & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & q_k(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ q_k(z) & 1 \end{bmatrix},$$

we can set

$$P^c(z) = \prod_{k=1}^{n/2} \begin{bmatrix} 1 & q_{2k-1}(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ q_{2k}(z) & 1 \end{bmatrix} \begin{bmatrix} K & 0 \\ 0 & K^{-1} \end{bmatrix}.$$

In case our choice of G^c does not correspond to the given complementary filter $\tilde{G}$, then by an application of Theorem 9.8.1, we can find a Laurent polynomial $s(z)$ such that

$$P(z) = P^c(z) \begin{bmatrix} 1 & s(z) \\ 0 & 1 \end{bmatrix}.$$

As a conclusion we can formulate the following theorem.

Theorem 9.8.2. *Given two complementary finite impulse response filters $(H(z), G(z))$, then there exist Laurent polynomials $s_k(z)$ and $t_k(z)$, $k = 1, \ldots, m$ and some nonzero constant K such that the polyphase matrix can be factored as*

$$P(z) = \prod_{k=1}^{m} \begin{bmatrix} 1 & s_k(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ t_k(z) & 1 \end{bmatrix} \begin{bmatrix} K & 0 \\ 0 & K^{-1} \end{bmatrix}.$$

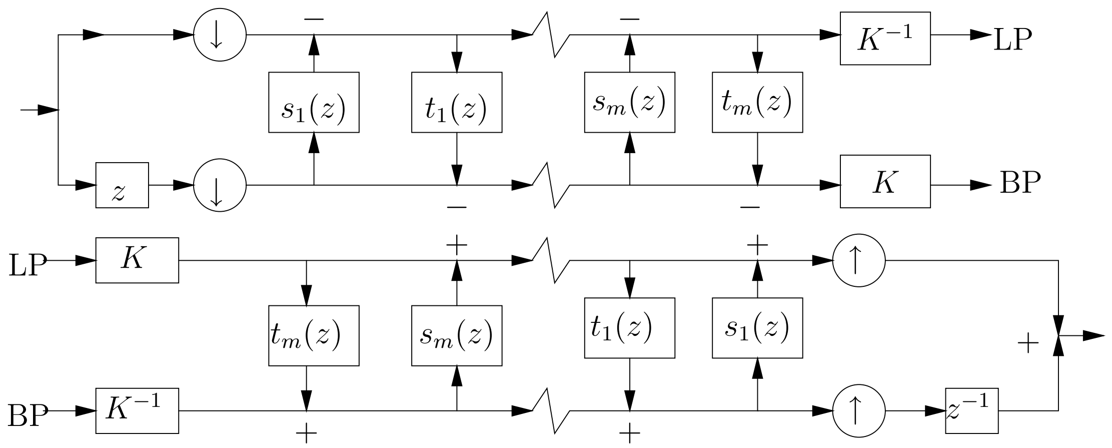
The interpretation of this theorem is obvious. It says that any couple of complementary filters which does (one step of) an inverse wavelet transform can be implemented as a sequence of primal and dual lifting steps and some scaling (by the constants K and K^{-1}). For the forward transform in the corresponding analysis step of a perfectly reconstructing scheme, the factorization is accordingly given by (recall $\tilde{P}_* = P^{-1}$)

$$\tilde{P}_*(z) = \begin{bmatrix} K^{-1} & 0 \\ 0 & K \end{bmatrix} \prod_{k=1}^{m} \begin{bmatrix} 1 & 0 \\ -t_k(z) & 1 \end{bmatrix} \begin{bmatrix} 1 & -s_k(z) \\ 0 & 1 \end{bmatrix}$$

or equivalently

$$\tilde{P}(z) = \prod_{k=1}^{m} \begin{bmatrix} 1 & 0 \\ -s_{k*}(z) & 1 \end{bmatrix} \begin{bmatrix} 1 & -t_{k*}(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} K^{-1} & 0 \\ 0 & K \end{bmatrix}.$$

Figure 9.8: Analysis and synthesis phase decomposed in a sequence of lifting steps



Example 9.8.2. The simplest of the classical wavelets one can choose are the (unnormalized) Haar wavelets. They are described by the filters

$$H(z) = 1 + z^{-1} \quad \text{and} \quad G(z) = \frac{1}{2}(-1 + z^{-1}).$$

The dual filters are

$$\tilde{H}(z) = \frac{1}{2}(1 + z^{-1}) \quad \text{and} \quad \tilde{G}(z) = -1 + z^{-1}.$$

It is clear how these compute a wavelet transform: the low pass filter $\tilde{H}$ takes the average and the high pass filter $\tilde{G}$ takes the difference of two successive samples. Note that we apply the filters $\tilde{G}_*$ and $\tilde{H}_*$, which corresponds here simply to a time-reversal. Thus

$$v_{l,k} = \frac{1}{2}(v_{l+1,2k} + v_{l+1,2k+1}) \quad \text{and} \quad w_{l,k} = v_{l+1,2k+1} - v_{l+1,2k}.$$

The polyphase matrix is trivially factored by the Euclidean algorithm as

$$P(z) = \begin{bmatrix} 1 & -1/2 \\ 1 & 1/2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & -1/2 \\ 0 & 1 \end{bmatrix}.$$

The dual polyphase matrix is factored as

$$\tilde{P}(z) = \begin{bmatrix} 1/2 & -1 \\ 1/2 & 1 \end{bmatrix} = \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1/2 & 1 \end{bmatrix}.$$

Thus, to compute the forward wavelet transform, we have to apply the lazy wavelet, i.e., take the even and the odd samples separately. Then, applying $\tilde{P}^t$ corresponds to a first primal lifting step leaving the even samples untouched and computes the difference $w_{l,k} = v_{l+1,2k+1} - v_{l+1,2k}$. In the next dual lifting step, this result is left untouched, but the even samples are modified by computing $v_{l,k} = v_{l+1,2k} + 1/2 w_{l,k}$.

For the inverse transform, first one computes $v_{l+1,2k} = v_{l,k} - 1/2 w_{l,k}$, and then $v_{l+1,2k+1} = v_{l+1,2k} + w_{l,k}$. This is just a matter of interchanging addition and subtraction.

Note that in this simple example, there is not really a gain in computational effort, but as our earlier examples showed, in general there is. $\diamond$

Many more examples of this idea can be found in the paper [8].

Chapter 10

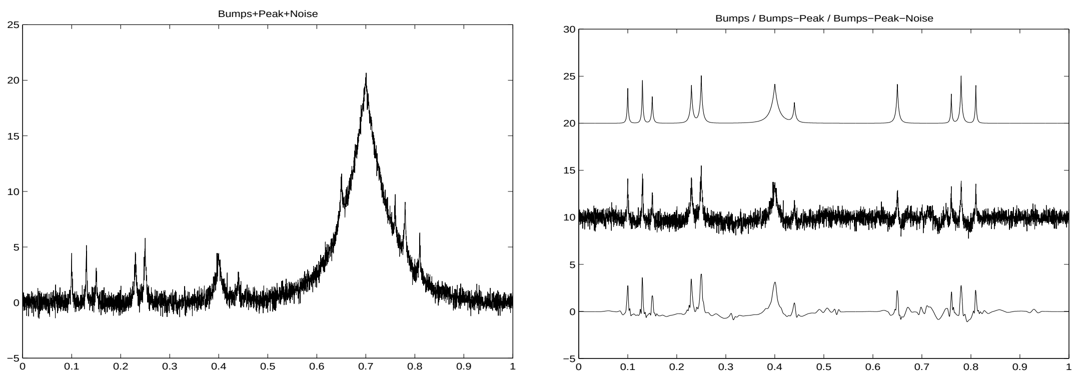
Applications

10.1 Signal processing

10.1.1 NMR Spectroscopy

Here is a simulated Nuclear Magnetic Resonance (NMR) spectroscopy experiment. A typical experiment can be described as follows. A sample is irradiated by a magnetic field, which is then switched off. The sample protons allign in this field, and subsequently relax to their equilibrium state. The frequencies are characteristic for their chemical environment. The Fourier spectrum is the signal to analyse. It consists of several sharp peaks (the spectral lines). Some of the peaks, coming from the protons of the solvent are quite large and should be eliminated. We have simulated such an experiment containing a large parasite peak, which has to be subtracted. Figure 10.1 gives the signal as it is observed, corrupted by the parasite peak and white noise. The upper plot on the right-hand-side is the "clean" signal that we want to recover. Obviously, the parasite peak is recognized to have a maximum

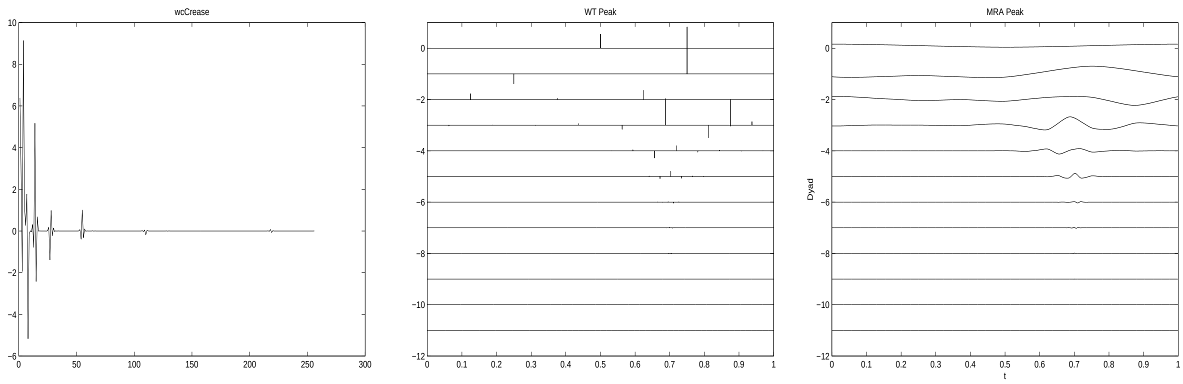
Figure 10.1: The observed signal and its analysis



near 0.7. The smaller peaks are the ones that have to be isolated. We have to subtract from

the observed signal this parasite peak and the noise. First, we simulate a peak with the function $\exp(-20 * |t - .70|)$ and compute its wavelet transform (Coiflet, 3 zero moments). This is represented in Figure 10.2 where we plotted the first 256 wavelet coefficients, the wavelet coefficients at the different scales and the MRA of the peak. We can see that the peak lives at low resolution, and that the "large" wavelet coefficients are located in the neighborhood of $t = 0.7$. Thus we compute the wavelet transform of the signal and make all the coefficients of the wavelet transform zero which are at level 0, 1, 2 (low resolution): (1:7). For the subsequent levels, we make the coefficients zero that are at the boundary (6:9), (14:17), (29:33), (62:65), to eliminate the boundary effects and finally, we set to zero all the coefficients which are related to basis functions in the neighborhood of $t = 0.7$, viz. (12:13), (25:28), (53:55), (108:113). The inverse WT is then computed which gives the result that is plotted in the middle of the right-hand-side Figure 10.1. Finally the remaining noise is removed using the SURE soft thresholding (see below). The final result is plotted at the bottom. The remaining "interesting" peaks are now clearly visible.

Figure 10.2: Peak signal



10.1.2 Music and audio signals

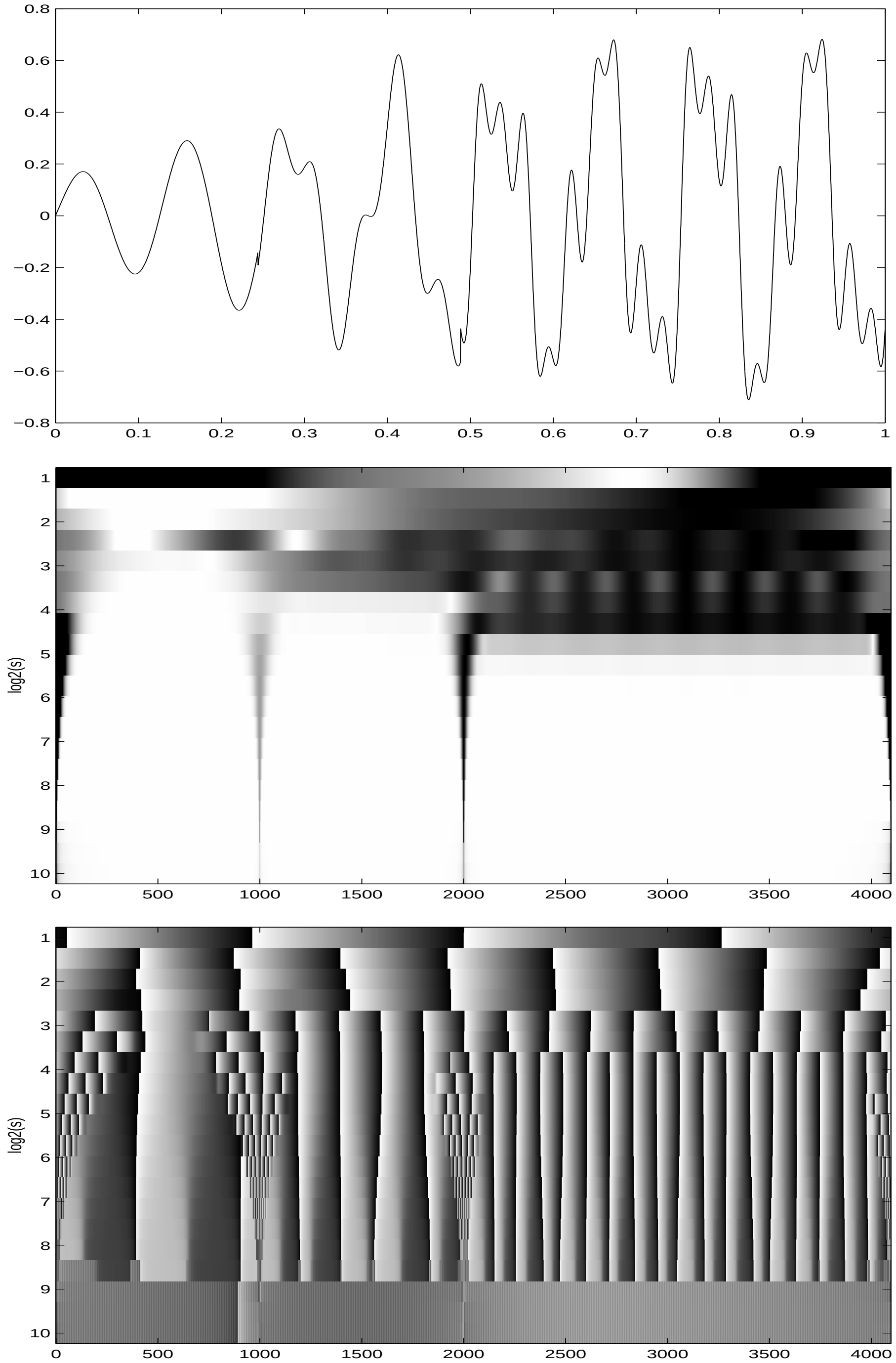
A music score is in fact a time-frequency representation of a audio signal. Suppose we have recorded on tape some signal which is the representation of a piece of music. If we play the tape at lower speed, the piece of music will have a lower pitch and will take more time; playing the tape at a higher speed will raise the pitch and it will take less time. The problem of time-stretching (without changing the pitch) or pitch-shifting (without changing the duration) seems to be simple operations when we have a time-frequency (e.g. a wavelet) representation of a signal. However, for such applications, CWT seems to be more appropriate than the DWT. So the computations take more time. Suppose we have a CWT $S(a, b)$ of a signal $s(t)$. It would then seem as simple as re-scaling the a-axis to change the pitch without changing the duration. However, the CWT is a redundant representation and changing $S(a, b)$, may transform this into something which is not the CWT of any function any more. Indeed, if every time-frequency representation would correspond to a CWT of a signal, then it would be easy to construct an example contradicting the Heisenberg uncertainty principle. It can be shown that if the inverse CWT of a function $S(a,b) \in L^2(\mathbb{R}^2)$ is formally applied and $S(a,b)$ is not the CWT of a signal, then the computed result (whenever it exists) is a least squares solution in the sense that its CWT is as close as possible to $S(a,b)$ in the Hilbert space $L^2(\mathbb{R}^2)$.

Thus, in conclusion, we can not just shrink for example the a-axis and compute the inverse CWT, because that would violate the Heisenberg uncertainty principle. Somehow this has to be compensated for by stretching the b-axis, but that is exactly what we want to avoid. In [9] the following solution is proposed. Suppose the signal $s(t)$ consists of N samples. First the *complex* Morlet wavelet $\psi(t) = e^{i\omega_0 t}e^{-t^2/2}$ is used to compute a complex transform $S(a,b)$. The result is a matrix where b is the column index ($b = 1 : N$) and a is the row index ($a = [a_1, a_2, \ldots, a_s]$ where s is the number of scales computed; usually a_k changes on a logarithmic scale: $a_k = 2^{k/m}$). Here m is the number of voices per octave. For example, if $k = sm + t$, then $a_k = 2^s 2^{t/m}$, which is voice t in octave s. Suppose $S(a,b) = M(a,b)e^{i\Theta(a,b)}$ where M is the modulus and Θ is the argument. Simply replacing a by a/c does not give a CWT anymore. Therefore, to obtain a pitch shift with a factor c, it is proposed to compute the inverse CWT of $S'(a',b) = M'(a',b)e^{i\Theta'(a',b)}$ with the dilation of the a-axis $a' = a/c$, and setting $\Theta'(a',b) = \Theta(a,b)$, but this is compensated by a phase shift $M'(a',b) = M(a,b)e^{ic\Theta(a,b)}$. Figure 10.3 represents a signal consisting of 3 sines and the modulus and phase its complex Morlet CWT (time axis (b) is horizontal and scale axis (a) is vertical; the value $S(a,b)$ is represented by the gray scales of the rectangles).

Another classical applications is compression of audio signals. The techniques used are the same as in image compression (see below).

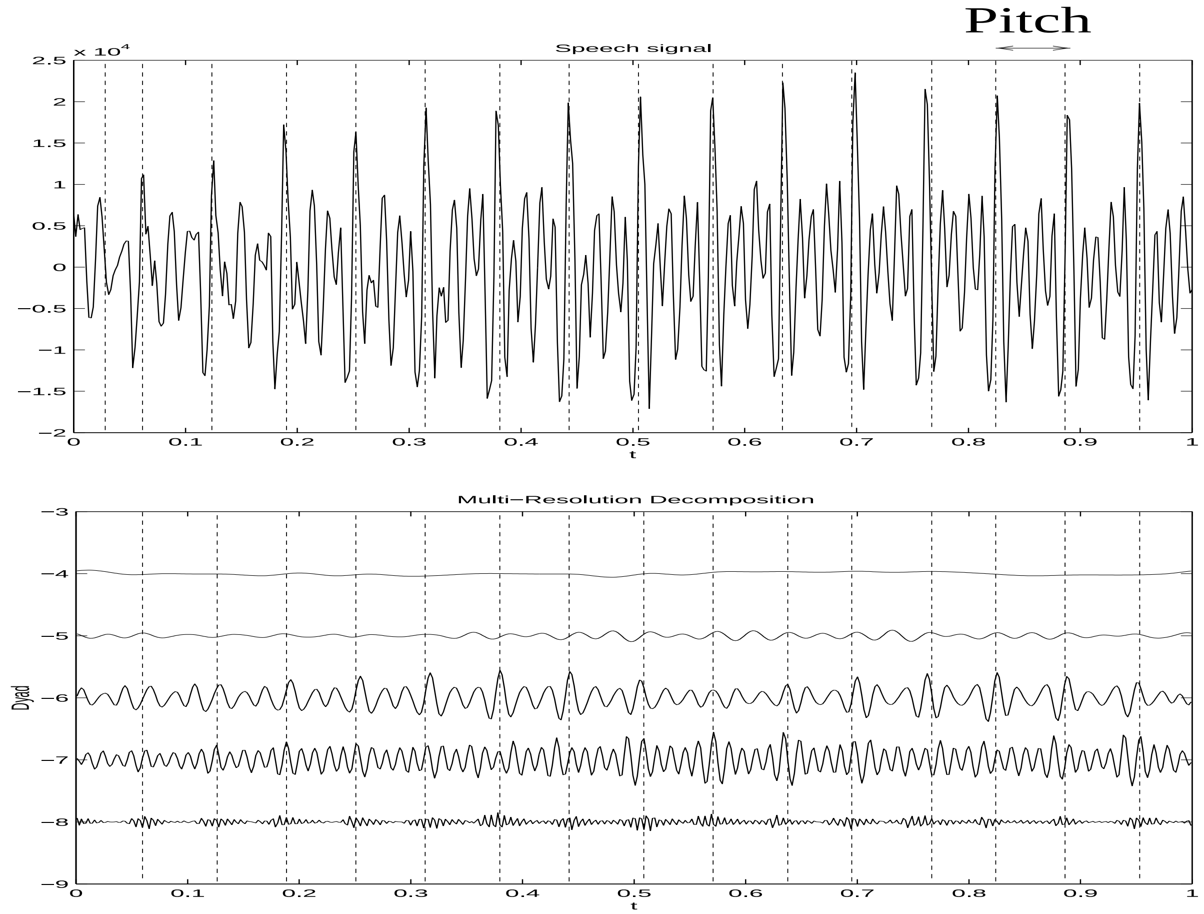
One of the important problems to be solved in almost any audio signal processing problem is *pitch tracking*. Pitch in human speech is defined as the frequency of the vocal chord vibration, or equivalently as the inverse of the in between *glottal closure instances* (GCI) that is the moments when the vocal chords close during speech. The pitch is needed in several applications like speech communication (synchronisation, transmission (see below), synthesis), speech and speaker recognition, phonetics and linguistics (study of prosodic and phonetic features such as tone, word stress, emotions) education (teaching intonation to the deaf), medicine (diagnosis of diseases), musicology etc. The detection of the pitch is in fact the detection of the successive periods in the speech signal, that is the successive “relevant” peaks and then measure the distance between them which is usually slowly varying. This is not so simple because there are many local maxima in the speech waveform. The relevant maxima are however made prominent when several levels of the wavelet transform are compared. The relevant maxima persist also in the low resolution levels (smooth approximations). About the computation of maxima see also the next section. An example of a pitch detector proposed by Kadambe and Boudreaux-Bartels uses 3 levels of the dyadic wavelet transform with the cubic spline wavelet. Other wavelets having about the same form as the cubic spline wavelet give similar results. The maxima are detected by setting a certain rough threshold: the peaks will come above the threshold at the successive levels. In Figure 10.4, we took a small piece (512 samples) from the ‘Caruso’ data-set from the WaveLab packet.

Figure 10.3: Music signal



We used a Coiflet-5 and plotted the speech signal as well as the multiresolution analysis. It

Figure 10.4: Pitch in speech signal



is a very noisy signal. Nevertheless, the pitch is clearly recognized.

Other techniques to recognize the pitch consist computing a (Morlet) CWT. The pitch will then be recognized as a distinguished horizontal line in the time-scale representation that corresponds to a fundamental frequency (at low resolution level) and some harmonics. Such techniques are used in recognition problems. For example voiced sounds that correspond to a vibration of the vocal chords (when uttering vowels) will have certain so called "formants" which are characteristic frequencies. They correspond to horizontal bands in the CWT plane. Because of the (limited) localization of the wavelets in the frequency domain, these frequencies come forward as blurred bands. The problem is then to recognize the relevant frequencies from that image.

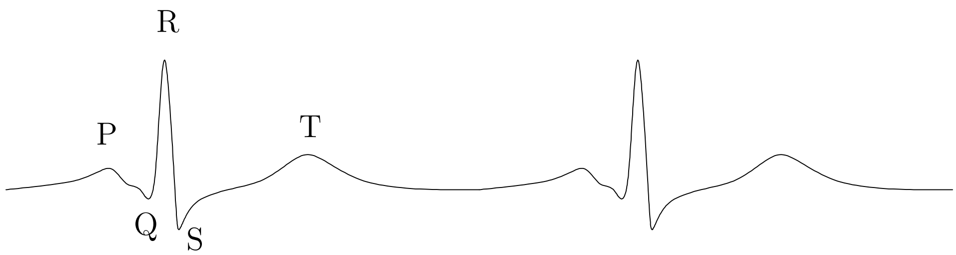
Most of the speech processing methods rely on cepstrum properties. The cepstrum is given by $\mathcal{F}^{-1}(\log|\mathcal{F}f|)$, i.e., the inverse Fourier transform of the log of the absolute value of the spectrum. This is because the cepstrum is claimed to decorrelate and catch the fundamental parameters to characterize the signal.

In fact the basic problem can be described as an identification problem where the model of the speech signal is of the form $f(t) = \sum_k A_k(t) \cos \varphi_k(t) + \eta(t)$. The $\eta(t)$ is a noise signal (or approximation error considered as noise), the amplitudes $A_k(t)$ and the phase $\varphi_k(t)$ are the parametes to catch. In fact it is crucial to find the $\varphi_k(t)$ because the amplitudes can then be easily computed. All kind of techniques using wavelets were proposed in the literature for solving this problem. However, till the present, wavelet methods may give better results for some of the audio processing problems, there is not an algorithm that is fast enough to be of commercial interest.

10.1.3 ECG signals

Like in the NMR signals, also in ElectroCardioGram (ECG) the problem is often to find peaks in the signal or more precisely to detect the elementary waveforms that compose the signal. In Figure 10.5 we have plotted a typical behaviour of an ECG signal. (In practice, this is also corrupted by noise.) A first step to the analysis is to identify the P-wave, the QRS and

Figure 10.5: An artificial ECG



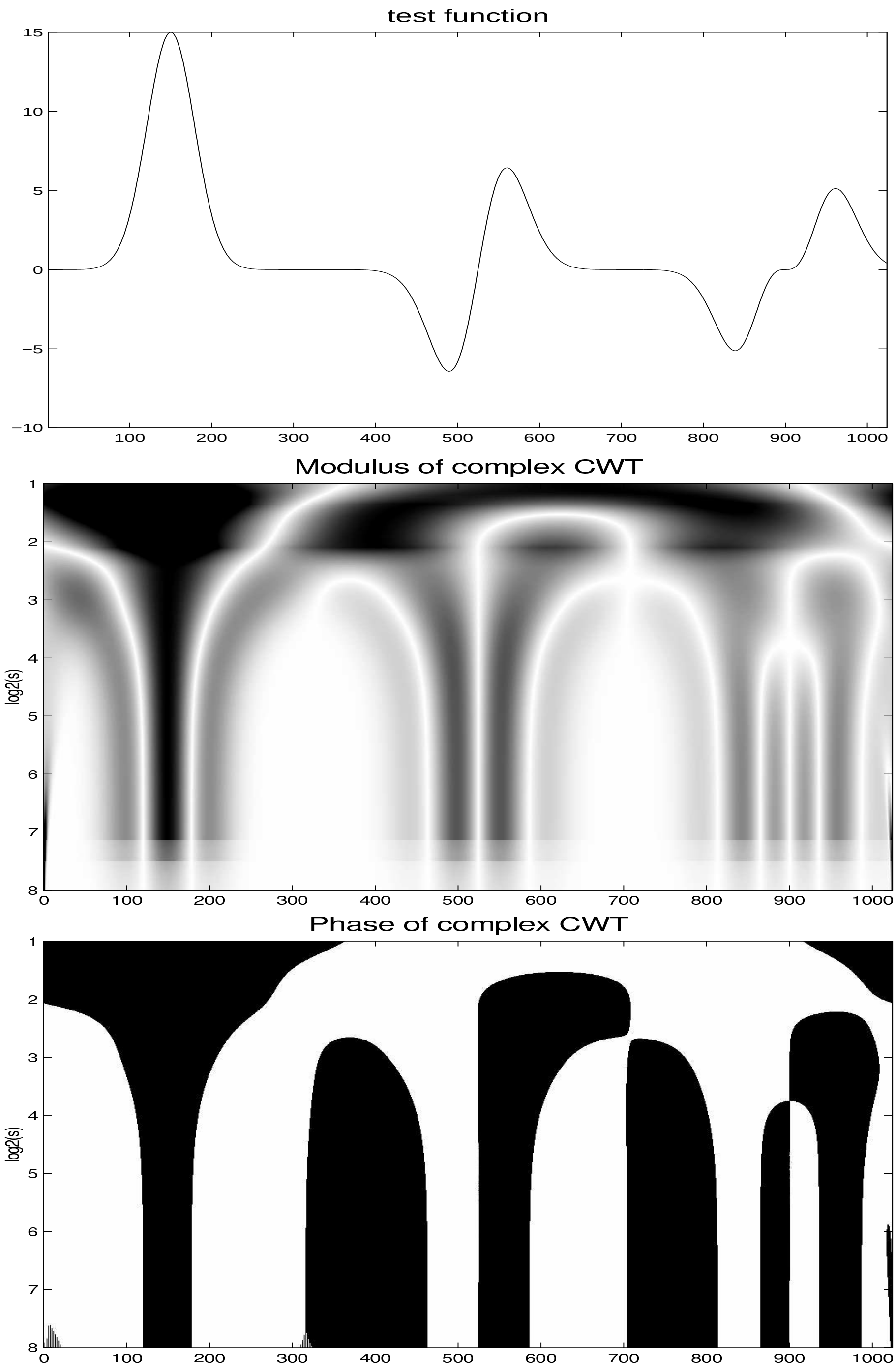
T-waves which are caused by atrial and ventricular polarization and depolarization. Each of these is characterized by symmetry properties, maxima, minima, and inflection points. From their location and values, certain clinical conclusions can be drawn.

Here again a complex CWT can be used. Suppose $S(a, b) = M(a, b)e^{i\Theta(a,b)}$ is the complex CWT, with M the modulus and Θ the phase. It can be shown that (under certain conditions) the b-values where a maximum of the modulus M (and for a pronounced peak this maximum persists through all scales) is reached, there is a point in the signal which is an extremum (first derivative zero). For inflection points with a second derivative equal zero, the modulus is maximal. For a maximum of M, there can be a maximum, a minimum, or an inflection point with an horizontal tangent for $s(t)$. In the case of a maximum, the phase is π, for a minimum, the phase is $\pm\pi/2$ (or jumps from π to 0). This helps classifying the nature of the point.

In Figure 10.6 we have plotted a simulated signal

$$s(t) = A_1 \exp(-(t - m_1)^2/b_1) + A_2(t - m_2) \exp(-(t - m_2)^2/b_2) + A_3(t - m_3)^3 \exp(t - m_3)^2/b_3$$

Figure 10.6: Maxima and minima by CWT



with $A_i = (15, 0.3, 0.0001)$; $b_i = (1700, 2500, 2500)$; $m_i = (150, 525, 900)$. below it, you see the modulus squared and the phase of the complex CWT, using the Mexican hat wavelet, (2 octaves with 96 voices/octave). For the modulus, we use the convention that black is large and white is small, For the phase, we have plotted the absolute value of the phase with black is π and white is 0.

10.2 Image processing

Wavelet transforms have been most successful in image processing. We shall discuss briefly image compression and image denoising.

10.2.1 Image compression

Consider an $n \times n$ image whose pixels can have 256 different gray scales. Then the storage of the image requires $N_{\text{orig}} = n^2$ bytes. If we succeed in representing a reasonable approximation of the image with only N_{comp} bytes, then we have a compression factor $N_{\text{orig}}/N_{\text{comp}}$. Of course with decreasing N_{comp}, we shall have poorer approximations of the image, but visually, there will be almost no difference for relatively high compression rates, provided an appropriate representation of the image can be found. For modern applications like transmission of images over the internet, compression is extremely important. Also in very large images from Geographic Information Systems (GIS) which may contain pictures taken from an airplane that require several Gibabytes, compression is highly important. Also in High Definition Television (HDTV), digital images are transmitted and again compression can make this feasable.

For colored images the representation is given by 3 images for the R(ed), G(reen), and B(lue) component. However, the RGB representation is usually transformed into a YUV representation where Y represents brightness and U and V represent colors. The human eye is much more sensitive for the Y component so that the UV components can be compressed much more without visual loss of quality.

The compression of an image is performed in 3 stages:

$$\boxed{\begin{matrix}\text{digital}\\ \text{image}\end{matrix}} \xrightarrow{\text{transform}} \boxed{\begin{matrix}\text{real}\\ \text{matrix}\end{matrix}} \xrightarrow{\text{quantizer}} \boxed{\begin{matrix}\text{integer}\\ \text{matrix}\end{matrix}} \xrightarrow{\text{coder}} \boxed{\begin{matrix}\text{compressed}\\ \text{image}\end{matrix}}$$

An in depth discussion of the quantization and of the coding procedure is beyond the scope of these notes and we shall only discuss this very briefly.

- **Transform:** If we replace the matrix of pixel values by its wavelet transform, then we obtain a strongly decorrelated representation, more than with a DFT or other transforms such as DCT (discrete cosine transform – used in JPEG) or Karhunen-Loeve transform (basically singular value decomposition). For image compression Daubechies wavelets or biorthogonal CDF wavelets are most commonly used. The wavelet transform has many coefficients with small absolute values, while only few have a significant value. Typically, the histogram of the absolute value of the coefficients have a sharp exponential decay. Replacing the small coefficients by zero will only have a minor visual effect on the image.

- **Quantization:** In this step, the matrix of real wavelet coefficients is replaced by a matrix of integers. The precision used may differ for different coefficients. For example low frequency coefficients are represented with more precision than coefficients related to high frequencies. The range of values for the wavelet coefficients is partitioned in intervals and a wavelet coefficient is replaced by the index of the interval to which it belongs. If we take 256 intervals, then every coefficient is replaced by 1 byte. The number of intervals depends on the contrast information. For example a text image, or in general a black-and-white image has only black or white pixels and in principle only 2 intervals are necessary. In case of the wavelet transform, a quantization per resolution level can improve the performance considerably. If a DCT is used, then setting to zero some of the coefficients will have a global effect on the image because the basis functions are not compactly supported. Therefore the JPEG standard will subdivide the image in smaller blocks of size 8×8 or 16×16 and the DCT is done for each of these blocks. High compression ratios will result in the typical blocking effect of JPEG.

- **Encoding:** The coding procedure is a method to represent the integer matrix as a sequence of bits in the most efficient way. For example the integers that appear most frequently are represented by only few bits, while the less frequent numbers are represented by more bits. Again coding is a science in its own and we refer to the literature for further details.

Note: Usually a digital image is given with integer values for the pixels. With the lifting scheme, it is possible to compute a 2D FWT of that image working only with integers [4, 3]. In that case the wavelet transform matrix will contain integers, but still a quantization, reorganizing the coefficients in a non-equispaced partitioning is usually necessary.

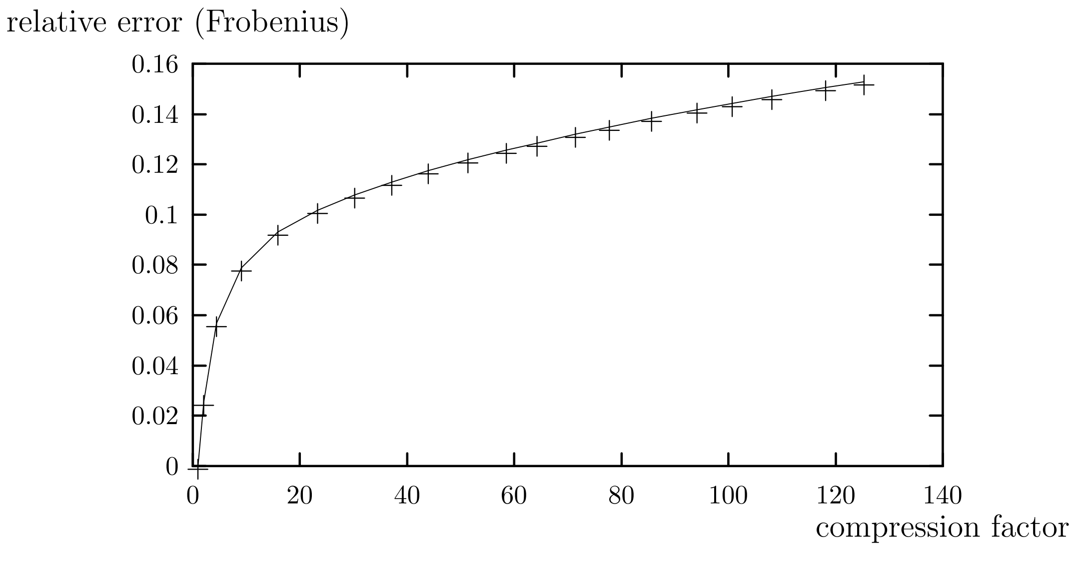
Example 10.2.1. Let us make abstraction of the quantization and coding steps and see how much can be gained from using a wavelet transform. We consider a 256×256 image P shown in Figure 10.8.A, with gray scales between 0 and 255 (1 byte per pixel). We compute the square 2D orthogonal FWT using the Daubechies filter D_2. Suppose we set all wavelet coefficients to zero that are smaller than $p\%$ of the largest coefficient. If this results in $r\%$ nonzero coefficients, then we have obtained a compression factor $C = 100/r$. When the image is reconstructed, we will get an approximation A of the original image P. Suppose we measure the relative approximation error with respect to the Frobenius norm:

$E = \|P - A\|_F / \|P\|_F$, where $\|A\|_F = (\sum_{ij} a_{ij}^2)^{1/2}$. We get the following result

p	C	r	E
0.0	1.00	100.00	0.00
0.1	4.42	22.62	0.06
0.2	15.93	6.27	0.09
0.3	30.18	3.31	0.11
0.4	43.98	2.27	0.12
0.5	58.57	1.71	0.13
0.6	71.39	1.40	0.13
0.7	85.67	1.17	0.14
0.8	100.67	0.99	0.14
0.9	118.08	0.85	0.15
1.0	135.40	0.74	0.16

The relative error versus the compression factor is plotted in Figure 10.7. In Figures 10.8 B

Figure 10.7: Relative error versus compression factor



and C, we show the reconstructed images for $p = 0.3$ and $p = 1.0$. $\diamond$

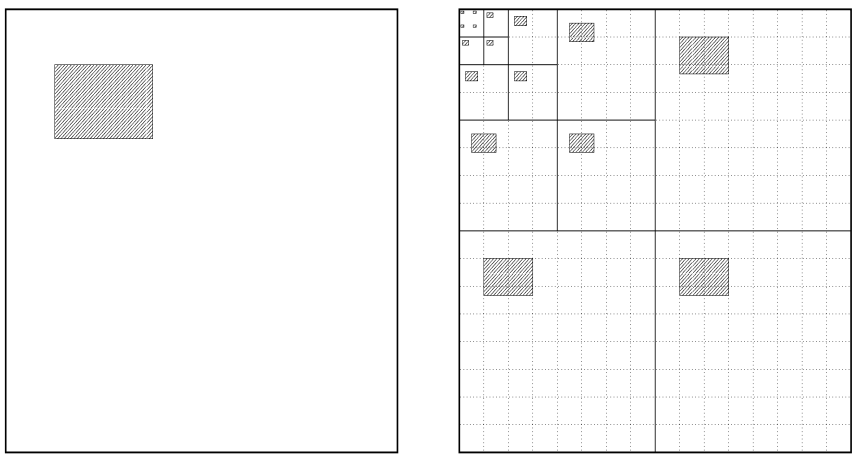
Large images

For very large images where the data can not be handled by the computer as one block, the image has to be tiled. The tiles could then transformed separately, but that would cause some blocking effects, just as the DCT which divides into 8×8 blocks. The same problem

Figure 10.8: Original image and 2 reconstructions from compressed forms

arises in audio signals where the signal is subdivided in frames and each frame could be transformed separately. Again here we have boundary effects that is unacceptable for hifi quality. These internal boundaries of the subblocks, i.e., of the tiles of the image or the frames of the audio signal, can be overcome by computing the transform at the boundary by borrowing some data from the neighbouring block. The number of neighbouring data that one needs depends on the length of the filter that is used. In the case of an image this may require some extra housekeeping to manage the blocks in the spatial as well as in the wavelet domain. The wavelet transform corresponding to one tile will be distributed over the whole wavelet transform domain and will be distributed over several blocks. See Figure 10.9.

Figure 10.9: Distributed transform of one tile



Such a management system is for example implemented in a transparant way for the user in the package WAILI.

10.2.2 Image denoising

Another important application of wavelets in image processing is denoising. The methods vary from very naive to very complex. We give some examples.

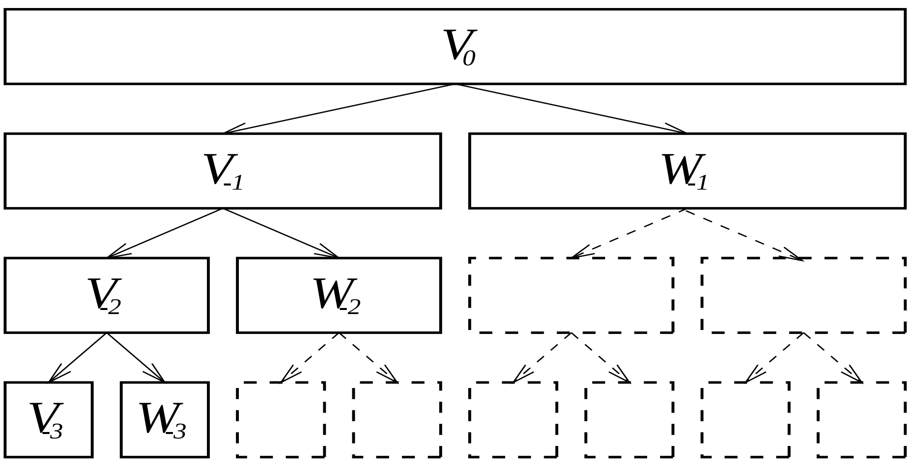
Libraries of waveforms

The idea is to represent a signal or an image as a linear combination of waveforms ("atoms") which are chosen from a catalog or "dictionary". The atoms in this dictionary may or may not be wavelet bases. In any case the dictionary is overcomplete so that a choice has to be made among the waveforms and of the linear combination to be taken. The part of the signal/image that can not be represented is assumed to be noise. Examples of this type are

- Matching pursuit (Mallat and Zhang [17])
- Basis pursuit (Donoho and Chen [5])
- Best orthogonal basis (Coifman and Wickenhauser [28])

As an illustration of the "Best orthogonal basis" idea, we take this opportunity to introduce the notion of *wavelet packet*. In Figure 10.10, we have shown in full lines the filter bank

Figure 10.10: Wavelet packet



algorithm as we have explained. At each step of the algorithm, only the low resolution part is split into a low-pass and a band-pass set of coefficients. As we have discussed it before, this corresponds to a change of basis. V_0 is transformed into $[V_{-3}|W_{-3}|W_{-2}|W_{-1}]$, which corresponds to a certain choice of the basis. However, it could be decided at every stage whether to split only the low resolution or the high resolution part. Splitting the high resolution part would result in a representation with respect to another basis. So we have a binary tree of possible basis functions that one could choose. The best orthogonal basis algorithm selects the most appropriate one.

The algorithm computes the whole tree (also the dotted lines) of Figure 10.10. This costs $O(n \log n)$ operations instead of the usual $O(n)$. Then the selection is made "bottom up". This means that according to some cost function it is computed what is the cheapest representation: the pair of low/high resolution basis or the global basis that is one level higher. For example in Figure 10.10, 4 such decisions have to be made: either a pair of blocks in the bottom row or the block immediately above it. This is to be recursively repeated all the way up the tree. The result is a basis that is the cheapest one among all the possible bases that can be constructed in this tree.

An image is a 2D signal and there we have not a binary tree but a quadtree but of course the same principle can be applied.

There are many possible choices to measure the "cost" of a vector (or a matrix if it concerns an image). If $\sum_k v_k \varphi_k$ is a representation of a signal with respect to the basis φ_k, then the cost of this representation is the "magnitude" of the coefficient vector v. The latter can be measured as

1. 1-norm: $\sum_k |v_k|$
2. Shannon entropy: $-\sum_k v_k^2 \log |v_k|$
3. Threshold based cost: $\sum_k \delta_T(v_k)$ where $\delta_T(x) = 1$ if $|x| > T$ and zero otherwise.
4. log energy: $\sum_k \log(v_k^2)$ which is equivalent with $\sum_k \log |v_k|$.

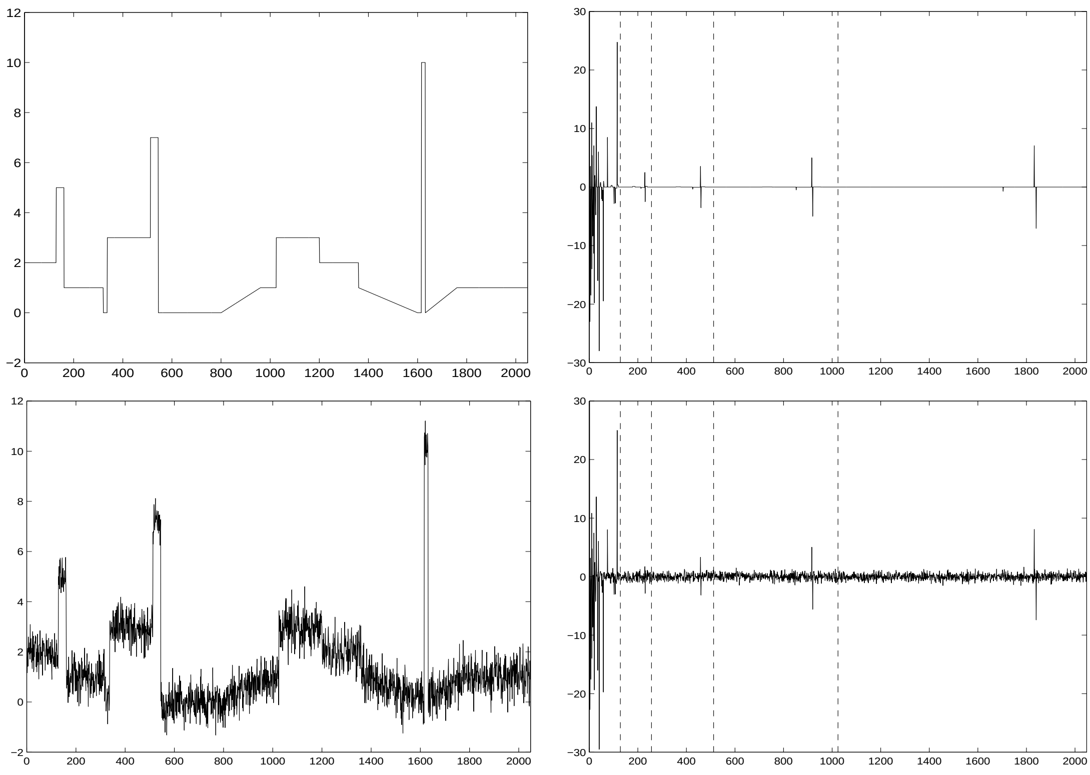
See also [27].

Wavelet shrinking

First of all, by the very nature of the wavelet transform, we know that the finer resolution levels will only contain detail information and that the main features of the signal/image are captured in the low resolution levels. Therefore, taking the wavelet transform, deleting the finest resolution levels, and backtransforming will have a smoothing effect and acts as a very elementary linear filter, which may reduce some of the noise, but it may also blur the original image. So we are looking for more sophisticated methods.

As we already said for image compression, many of the wavelet coefficients are small. To fix the ideas let us start with the 1D case. An example is shown in Figure 10.11 where a clean and a noisy signal are shown together with their (orthogonal) wavelet transforms. It is clear that there are only a few large coefficients and that the (stationary white) noise is transformed into (stationary white) noise on the wavelet coefficients, uniformly at all scales. Thus, it might be a good idea to set to zero all the small wavelet coefficients which is exactly

Figure 10.11: Clean and noisy signal and wavelet transforms



what we did in our compression method. The scheme is quite similar:

$$\boxed{\begin{array}{c}\text{noisy}\\ \text{data}\end{array}} \xrightarrow{\text{FWT}} \boxed{\begin{array}{c}\text{wavelet}\\ \text{coefficients}\end{array}} \xrightarrow{\text{manipulate}} \boxed{\begin{array}{c}\text{modified}\\ \text{coefficients}\end{array}} \xrightarrow{\text{FWT}^{-1}} \boxed{\begin{array}{c}\text{denoised}\\ \text{data}\end{array}}$$

Let us have a closer look at the mathematics. Typically, noise is considered to be stochastic. We observe the signal

$$h_i = f_i + n_i, \quad i = 1, \ldots N, \quad \text{or} \quad \mathbf{h} = \mathbf{f} + \mathbf{n}$$

where $\mathbf{h} = (h_i)$ is the observed signal, $\mathbf{f} = (f_i)$ is the clean signal and $\mathbf{n} = (n_i)$ is the noise. Suppose that the stochastic variables n_i have mean zero: $\mathcal{E}[n_i] = 0$ and a covariance matrix

$$C_{ij} = \mathcal{E}\left[(n_i - \mathcal{E}n_i)(n_j - \mathcal{E}n_j)\right] = C_{ij} = \mathcal{E}\left[n_i n_j\right].$$

The variance is $\mathcal{E}[n_i^2] = \sigma_j^2$. If n_i is *white noise*, then it is uncorrelated, i.e., the covariance matrix is a diagonal: $C_{ij} = \sigma_i^2 \delta_{i-j}$. The noise is called *stationary* if the variance σ_i is not depending on i: $\mathcal{E}[n_i^2] = \sigma^2$. Thus for stationary white noise is $\mathbf{C} = \sigma^2 \mathbf{I}$. Thus if the noise is i.i.d. (independent and identically distributed) then it is stationary and white.

Now if we take the wavelet transform of $\mathbf{h} = \mathbf{f} + \mathbf{n}$ by multiplying with the wavelet matrix $\mathbf{W}$, we obtain

$$\mathbf{Wh} = \mathbf{Wf} + \mathbf{Wn} \quad \text{or} \quad \mathbf{y} = \mathbf{x} + \mathbf{v}.$$

Thus $\mathbf{y}$ contains the wavelet transform of the observed signal, $\mathbf{x}$ of the clean signal and $\mathbf{v}$ of the noise. One may consider $\mathbf{v}$ as noise on the clean coefficients $\mathbf{x}$. Since the covariance matrix of $\mathbf{v}$ is $\mathbf{D} = \mathbf{WCW}^T$, it is clear that if the wavelet transform is orthogonal (i.e., if $\mathbf{W}$ is an orthogonal matrix), and if $\mathbf{n}$ is white and stationary, then also $\mathbf{v}$ is white and stationary. More generally, one can prove that the wavelet transform of stationary noise is stationary within each resolution level. This is illustrated in Figure 10.12 where the biorthogonal FWT of some colored stationary noise is shown. Thus, the denoising method as explained below should be made level-dependent for practical applicability, but for simplicity, we shall assume that we have stationary white noise, so that the reasoning holds for all scales.

Before we explain the denoising, we first define some measures for the amont of noise in a signal. Suppose we have a clean signal/image $\mathbf{f}$ and a noisy version $\hat{\mathbf{f}}$, then the *Signal to Noise Ratio* (SNR) is defined as

$$\text{SNR} = 10 \log_{10} \frac{\|\mathbf{f}\|_2^2}{\|\mathbf{f} - \hat{\mathbf{f}}\|_2^2} = 10 \log_{10} \frac{\sum_k f_k^2}{\sum_k (f_k - \hat{f}_k)^2}.$$

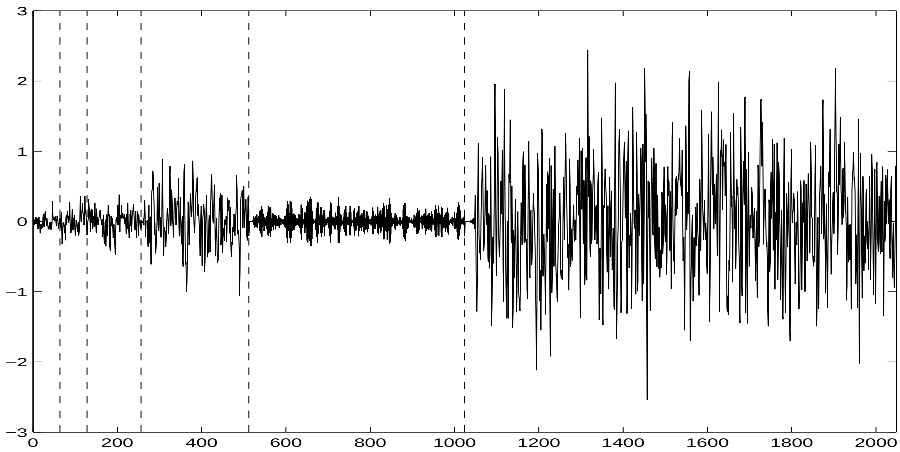
The Peak SNR (PSNR) is

$$\text{PSNR} = 10 \log_{10} \frac{\max_k f_k^2}{\sum_k (f_k - \hat{f}_k)^2}.$$

Both are expressed in decibel (dB). Maximizing the SNR is equivalent to minimizing the least squares or Mean Square Error (MSE)

$$\text{MSE} = \frac{1}{N} \sum_{k=1}^{N} (f_k - \hat{f}_k)^2.$$

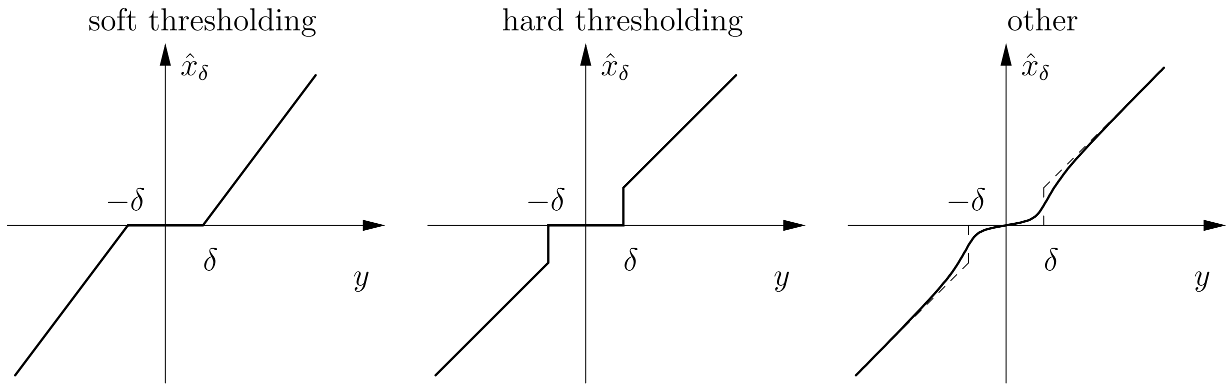
Figure 10.12: Biorthogonal wavelet transform of stationary noise



Observe that for an orthogonal wavelet transform, minimizing the MSE for the pixel values is equivalent to minimizing the MSE for the wavelet coefficients.

We now try to estimate the clean coefficients x_i from the noisy coefficients y_i, for example by shrinking the coefficients: $\hat{x}_i = s_i y_i$ with $0 \leq s_i \leq 1$, where there is usually a preference to leave the large coefficients untouched ($s_i \approx 1$) and reduce the small coefficients to almost zero ($s_i \approx 0$). Some typical examples are given by *hard thresholding* and *soft thresholding* as illustrated in Figure 10.13. In hard thresholding, the wavelet coefficients which are in

Figure 10.13: Thresholding



absolute value smaller than δ are replaced by zero, while the other coefficients are left untouched. This is the most obvious choice since indeed, the noise will be represented by small coefficients, while small coefficients which represent something of the clean image can be set to zero without much visual damage. However, hard thresholding is a rather discontinuous operation and is mathematically less tractable. Therefore, one often replaces this by soft thresholding. Figure 10.13 is self explanatory. Of course, an even smoother strategy, like in the third part of Figure 10.13 could also be chosen.

Once the wavelet coefficients $\mathbf{x}$ are estimated by $\mathbf{x}_\delta$, then an inverse wavelet transform can be computed and we get an approximate signal/image $\mathbf{f}_\delta$.

The hardest part of thresholding techniques is of course the determination of an optimal or suboptimal threshold δ (possibly level dependent). In the ideal case, we should maximize the SNR, or equivalently, minimize the MSE

$$R(\delta) = \frac{1}{N}\sum_{k=1}^{N}(\hat{f}_{\delta k} - f_k)^2.$$

For an orthogonal wavelet transform, this is equivalent with minimization of the MSE in the wavelet domain

$$S(\delta) = \frac{1}{N}\sum_{k=1}^{N}(\hat{x}_{\delta k} - x_k)^2.$$

However, neither the clean image $\mathbf{f}$, nor the clean wavelet coefficients $\mathbf{x}$ are known.

There are several possibilities:

- **universal threshold** (Donoho, Johnstone) This threshold is

$$\delta = \sqrt{2 \ln N}\sigma$$

  where N is the number of data points and σ is the standard deviation of the noise. Of course, in practice σ is not really given and should be estimated by statistical techniques. It can be shown that, under certain restrictive conditions, it has some optimality properties, but in general it has an over-smoothing effect and not always appropriate for image denoising.

- **SURE threshold** (Donoho, Johnstone) This follows from an approximation of the MSE function, namely, instead of minimizing the MSE function $S(\delta)$, one minimizes

$$\mathrm{SURE}(\delta) = \frac{1}{N}\sum_{k=1}^{N}(\hat{x}_{\delta k} - y_k)^2 + 2\sigma^2\frac{N_1}{N} - \sigma^2$$

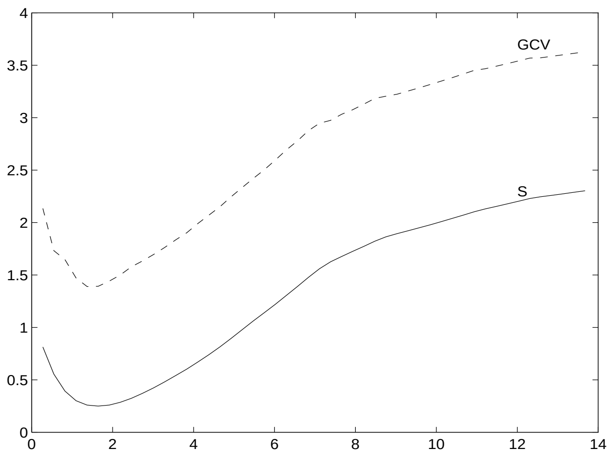
  where σ is again the standard deviation of the noise and N_1 is the number of coefficients with magnitude above the threshold. One can prove that for Gaussian noise, the expected value of SURE(δ) and of $S(\delta)$ are the same. The drawback is that again σ has to be estimated.

- **GCV threshold** This is based on another "approximation" of the MSE. The Generalized Cross Validation (GCV) is a notion from approximation theory which is defined as

$$\mathrm{GCV}(\delta) = \frac{\frac{1}{N}\sum_{k=1}^{N}(\hat{x}_{\delta k} - y_k)^2}{(N_0/N)^2}$$

where N_0 is the number of wavelet coefficients that is replaced by zero. In general, the function GCV(δ) and $S(\delta)$ have similar behavior, in the sense that the minimum of GCV(δ) and of $S(\delta)$ are obtained for approximately the same value of δ. A typical example is shown in Figure 10.14. The GCV has the advantage that it is computable

Figure 10.14: GCV and S



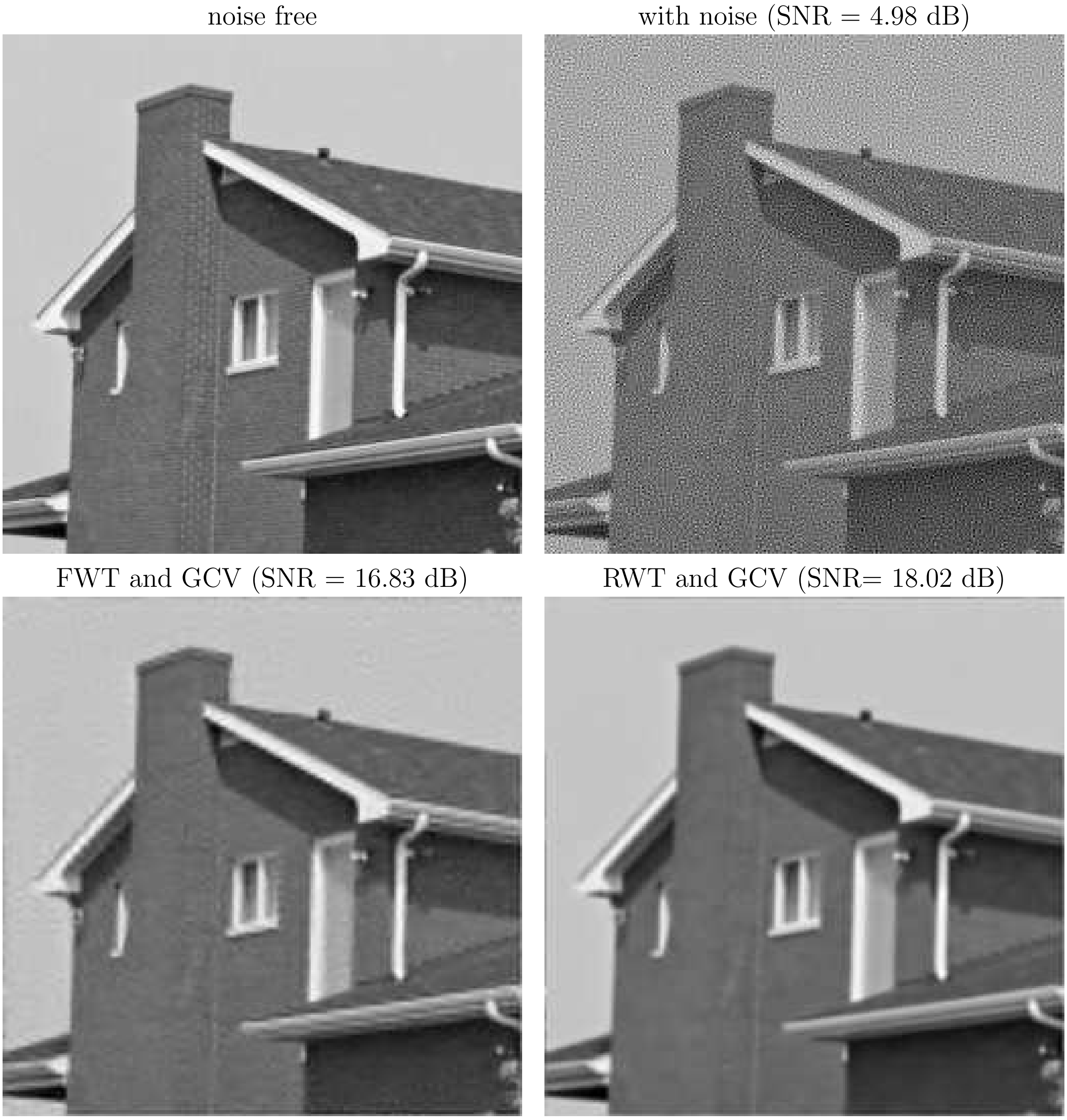
in terms of known numbers (we do not need σ), it has linear complexity, and it is asymptotically optimal as $N \to \infty$ [12].

Example 10.2.2. The image given in Figure 8.3 is corrupted by artificial additive correlated noise. The result is shown in Figure 10.15. One can see the result from a level dependent thresholding (3 levels) using the GCV estimate, soft thresholding and using the FWT. Note the drastic increase in SNR from 4.98 dB to 16.38 dB. Because the GCV estimate is only performing well if N is large, it is better to use the RWT here. The averaging that can be done in the inverse transform has an additional smoothing effect. The result is shown in the last image, which has an even larger SNR of 18.02 dB. ◇

10.3 Wavelet modulation in communication channels

To transmit digital information over some transmission channel, resources are scarce. One is limited in bandwidth and time. If the bandwidth were infinite, then one can send the messages of many users in a fraction of time, but since the bandwidth is usually rather limited, and since there are many users who want to send messages over the same channel, it is important to design methods to let as many messages as possible go through (*maximal throughput*). Moreover, the system should be robust for noise, so that there is a low *bit error rate* (BER). Finally, the system should be *secure*, i.e., not allow for an easy decoding by "the enemy".

Figure 10.15: Denoising with GCV



The channel resources can be considered as a rectangular window in the time frequency plane. There is of course an upper bound for what can be achieved, which is called the *channel capacity*.

Suppose there are N users who want to send one bit each, or equivalently, one user wants to send one number of N bits: $b = (b_1, b_2, \dots, b_N)$. In the first case this is called *multiple access communication* since more users have access to the same channel. In the second case, where the message of one user is "parallellized" by a serial-to-parallel converter, this is called *multitone communication*. The principle is exactly the same. Now this number b is modulated by a set of atoms, i.e., a set of functions $\Phi = \{\varphi_k\}$, to form a signal $f(t) = \sum_{k=1}^{N} b_k \varphi_k(t)$. Thus user k is characterized by an atom φ_k. Classically, one takes these atoms orthogonal to each other, which will of course simplify the decoding. Also, in such a case there will be no *multiple access interference*, because the users are completely decorrelated. Typical examples are φ_k whose support does not overlap in time. This is called *time division multiple access* (TDMA). Thus each user gets the whole bandwidth for a certain slice of time to send his message. Another possibility is that the frequency content of the φ_k do not overlap. Thus here the N users get their own limited frequency band all of the time. This is called *frequency division multiple access* (FDMA). Thus the channel, represented by the time-frequency rectangle is subdivided into horizontal or vertical slices. One could however use some wavelet-like functions φ_k, so that the time-frequency rectangle is divided into subrectangles in a typical wavelet-like manner as in the third picture of Figure 2.6. There may be several reasons why one wants to distribute the message over the time-frequency window of the channel. For example in the case of a mobile sender (telephone) the signal might be temporarily weaker, or there might be interference because the receiver gets a superposition of the message reflected on several physical objects. Or there can be interference from an accidental nearby alian sender, or by a deliberate enemy scrambler. Such an interference can be frequency dependent. It will also be more difficult to decode if the information is spread out. What kind of wavelet functions should we use? We could use an orthogonal wavelet basis, but assume that we use a more general frame, provided by a wavelet packet, classically with orthogonal atoms. This is called *wavelet packet multiple access* (WPMA). The fact that the transform is redundant allows some freedom for error recovery.

One could also use a non-orthogonal frame Φ. Note that in this case the users are not totally decorrelated and there will be some multiple access interference. The frame operator L is defined as

$$L : L^2 \to \ell^2 : f \mapsto Lf = \{\langle f, \varphi_k \rangle\}, \quad \langle f, \varphi_k \rangle = \int f(t)\overline{\varphi_k(t)}dt.$$

Its adjoint is

$$L^* : \ell^2 \to L^2 : b = (b_k) \mapsto L^* b = \sum b_k \varphi_k.$$

Thus, if the received signal f is corrupted by additive noise so that we actually receive $f_r = f + \sigma \cdot w$ (σ is the noise level and w is a normalized noise), then a least squares solution for the decoding can be found by solving $b \approx R^{-1} L f_r$ where $R = LL^*$ is invertible if Φ is a frame. Of course, since we know that the solution should be a vector of bits, we should find the closest binary solution.

Another, simpler, and general solution is proposed by Teolis [24]. It may be described as follows. The supports of the φ_k generate a wavelet-like tiling of the time-fequency plane. Thus if we plot the signal f (or f_r) in the time-frequency plane, then we recognise the blobs at the tile for atom φ_k if the bit b_k is 1. To be more concrete, consider an overcomplete wavelet transform in the sense that we compute a continuous wavelet transform, but sample it at a discrete sample set $(a,b) \in \{(t_{m,n}, s_m) : n,m \in \mathbb{Z}\}$. Suppose we denote $\psi_{a,b}$ with $(a,b) = (t_{m,n}, s_m)$ as ψ_{nm}. To reduce the possible BER, we should make the system ψ_{nm} as dense as possible in the time-frequency rectangle $\mathcal{R}$ characterizing the channel limits. For simplicity, we shall also divide the N bits into groups and give them a double index: b_{nm}. Then $f = L^* b = \sum_{n,m} b_{nm}\psi_{nm}$. The decoding depends on several factors:

- r_s: the support factor ($0 \le r_s \le 1$)
- δ_d: detection threshold
- δ_n: noise rejection threshold.

First we define time-frequency masks

$$M_{nm}(\omega,t) = \begin{cases} 1, & |\psi_{nm}(t)| > r_s\|\psi\|_\infty \quad \& \quad |\Psi_{nm}(\omega)| > r_s\|\Psi\|_\infty \\ 0, & \text{else} \end{cases}$$

where Ψ refers to the Fourier transform of ψ. These masks define rectangles in the rectangle $\mathcal{R}$ by

$$R_{nm} = \{(\omega,t) : |\psi_{nm}(t)| > r_s\|\psi\|_\infty\} \bigcap \{(\omega,t) : |\Psi_{nm}(\omega)| > r_s\|\Psi\|_\infty\}.$$

Their meaning is that ψ_{nm} lives essentially in the rectangle R_{nm}. If the design of the ψ_{nm} is good then these rectangles should form approximately a tiling of the rectangle $\mathcal{R}$. Thus, if we plot the transmitted signal f or f_r in the time-frequency plane, then it will show a blob in rectangle R_{nm} if the bit b_{nm} is 1. That is essentially how we shall read of the original bits b_{nm}. How shall we detect that rectangle R_{nm} has a blob or not, because there may be a lot of noise on the signal that fades out the blobs? First, we compute a thresholded wavelet transform and set

$$(\mathcal{W}_{\psi,\delta_n} f)(\omega,t) = \begin{cases} (\mathcal{W}_\psi f)(\omega,t), & |(\mathcal{W}_\psi f)(\omega,t)| > \delta_n\|\mathcal{W}_\psi f\|_\infty \\ 0, & \text{else} \end{cases}$$

This has a denoising effect. Then we compute the energy in rectangle R_{nm}. If it is large enough, we accept a bit $b_{nm} = 1$, otherwise, it is assumed that the bit b_{nm} was 0. Thus, we compute

$$b_{nm} \approx d_{nm} = \begin{cases} 1, & \int_{R_{nm}} |(\mathcal{W}_{\psi,\delta_n} f)(\omega,t)|^2 dt d\omega > \delta_d \\ 0, & \text{else.} \end{cases}$$

10.4 Other applications

There are many more applications to be found in the literature.

10.4.1 Edge detection

For example, since edges of a signal/image are recognized by large wavelet coefficients that persist over several resolution levels, the wavelet transform can be used to find (sharp) edges in a signal/image.

10.4.2 Contrast enhancement

If one has detected the edges, then one can modify the wavelet coefficients at these places, which will give a larger edge gradient, that is enlarge the difference between the pixel values at both sides of teh edge, thus give sharper edges, and hence will give a more pronounced contrast in the image. In principle the increase in contrast should be perpendicular to the edges.

We describe here a method based on the multiscale edge representation of images as described by Mallat and Zhong [18]. Consider a separable spline scaling function $\varphi(x,y)$, which will play the role of a smoothing filter. Corresponding directional wavelets are defined by partial derivatives:

$$\psi^1(x,y) = \frac{\partial}{\partial x}\varphi(x,y), \quad \psi^2(x,y) = \frac{\partial}{\partial y}\varphi(x,y).$$

If φ is smooth enough and decays fast enough, then both ψ^1 and ψ^2 will be admissible wavelet functions. The 2D dyadic wavelet transform of $f \in L^2(\mathbb{R}^2)$ at scale 2^n, position (x,y) and orientation r is $\mathcal{W}^r_{2^n} f(x,y) = f * \psi^r_{2^n}$, $r = 1,2$ and $\psi^r_{2^n}(x,y) = 4^{-n}\psi^r(2^{-n}x, 2^{-n}y)$. The result is a vector field, called the "multiscale gradient" which we denote as

$$\nabla_{2^n} f(x,y) = (\mathcal{W}^1_{2^n} f(x,y), \mathcal{W}^2_{2^n} f(x,y)) = \frac{1}{4^n}\nabla f * \varphi_{2^n}(x,y).$$

Mallat and Zhong showed that it is possible to recover the image from the data $(\nabla_{2^n} f(x,y))_{n\in\mathbb{Z}}$. Since edges correspond to sharp variations in $f(x,y)$, one should find the maxima of the magnitudes

$$\mu_{2^n} f(x,y) = \|\nabla_{2^n} f(x,y)\|$$

of the multiscale gradient. We say that (x,y) is a multiscale edge point in the direction

$$\theta_{2^n} f(x,y) = \arctan\left[\frac{\mathcal{W}^1_{2^n} f(x,y)}{\mathcal{W}^2_{2^n} f(x,y)}\right]$$

if $\mu_{2^n} f(x,y)$ attains there a local maximum. Suppose these local maxima appear in the points (x_i, y_i), then define

$$A_{2^n}(f) = \{[(x_i,y_i), \nabla_{2^n} f(x_i,y_i)]\}$$

and let for the coarsest level J, $F_J(x,y)$ be the the 2D wavelet transform, then

$$\{F_J(x,y), A_{2^n}(f)_{1\le n\le J}\}$$

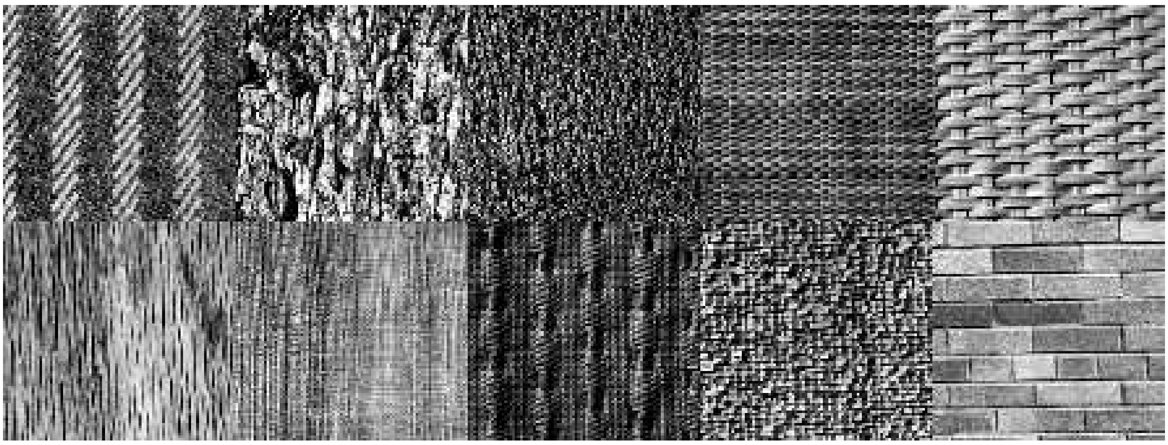
is called the multiscale edge representation of $f(x,y)$. Then the image can be reconstructed from this representation. Thus all the data in between the edges can be reconstructed from the information given in these edges. That is if $f \in V_J$, otherwise a close approximation is found. The algorithm is iterative and quite time consuming, but at least in principle it will work. The multiscale edge representation is very sparse and has therefore high potentials for image compression and has also been used for image denoising.

As for contrast enhancement, this edge information can also be exploited to obtain the desired result. Indeed, we just have to stretch the gradient and replace in the multiscale edge representation $\nabla_{2^n} f(x_i, y_i)$ by $k\nabla_{2^n} f(x_i, y_i)$ where $k > 1$ may or may not depend on the level n.

10.4.3 Texture analysis

Texture is recognized by several (statistical) parameters which are characteristic for that kind of texture. There are two possible problems for classification. Either the texture has to be recognized. Then it should be compared with a dictionary of textures and the computed parameters should be close enough to the parameters of the texture-class to which it belongs. In segmentation problems however, the image has to be subdivided into several segments which are defined by "having the same texture". Using some classification method, it is possible to recognize certain textures or to define the segments automatically. Since texture has certain multiscale characteristics, wavelets may help solving these problems.

Figure 10.16: Examples of textures



Parameters that are often used are first and second order statistics. Denote the 4 subimages at level n of the 2D transform by $L_n(x,y)$, for the low resolution part and $W_{ni}(x,y)$, $i = 1,2,3$ for the high resolution parts. For reasons of translation invariance and to keep enough data, the redundant wavelet transform is used. Then one can compute for example first order statistics like the energies E_{ni} or the mean deviations MD_{ni} as

$$E_{ni} = \frac{1}{N}\sum_{j,k}(W_{ni}(x_j, y_k))^2, \quad MD_{ni} = \frac{1}{N}\sum_{j,k}|W_{ni}(x_j, y_k)|.$$

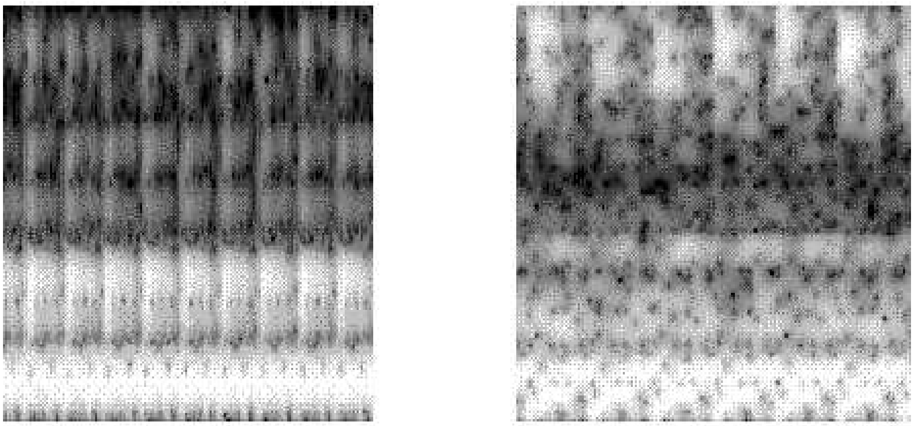
Second order statistics are computed from cooccurence matrices. That is a matrix defined as follows. First D_{ni} is quantized so that it takes (a finite number of different) integer values say. Then element (j, k) in the cooccurrence matrix $C_{ni}^{\delta\theta}$ is the joint probability of $D_{ni}(x, y) = j$ and $D_{ni}(x', y') = k$ occurring at the same time where (x', y') is at a distance δ in the direction θ away from (x, y).

Anyway, in this way one computes a number of parameters that should be characteristic for the texture. In the high-dimensional parameter space one then has to look for to what cluster of parameters characterizing a certain texture that the computed parameters do belong. This is a problem of classification that goes beyond the scope of these lecture notes since it belongs to the domain of artificial intelligence.

In many cases the texture has a directional flavour: for example a brick wall or the bark of a tree or a fabric. In that case the method should recognize textures even if they are rotated with respect to each other. Therefore directional wavelets are used to catch this directional information, or one computes averages to obtain rotation invariant parameters.

Such texture analysis methods do have applications in the medical sector. For example to recognize cancer cells from healty ones or to diagnose. Also one dimensional speech signals may be transformed into an image which is called a spectrogram. Certain defects in the speech system of the patient give spectrograms that differ in texture from a normal spectrogram and thus texture analysis of that spectrogram image can be used to classify the defect.

Figure 10.17: Spectrogram for a vowel `/a/`, left of a normal voice, right of a dysphonic voice. This image is taken from [25].



10.4.4 Computer graphics

As is illustrated by the subdivision schemes, wavelets, and more especially second generation wavelets are very appropriate for geometric modeling. Changing a wavelet coefficient will only change the surface locally and conversely, when a bump is added to some surface, this will only affect some wavelet coefficients locally. The theory is still being developed to handle wavelets for surfaces defined by scattered data.

10.4.5 Numerical analysis

For the solution of certain functional equations (integral or differential equations) the problem is often linearized which gives rise to large linear systems. These matrices can be considered as an image which may have a sparse representation in the wavelet domain, allowing for a sparse solution method.

In fact all multigrid methods used in numerical analysis fit perfectly in the idea of multiresolution analysis.

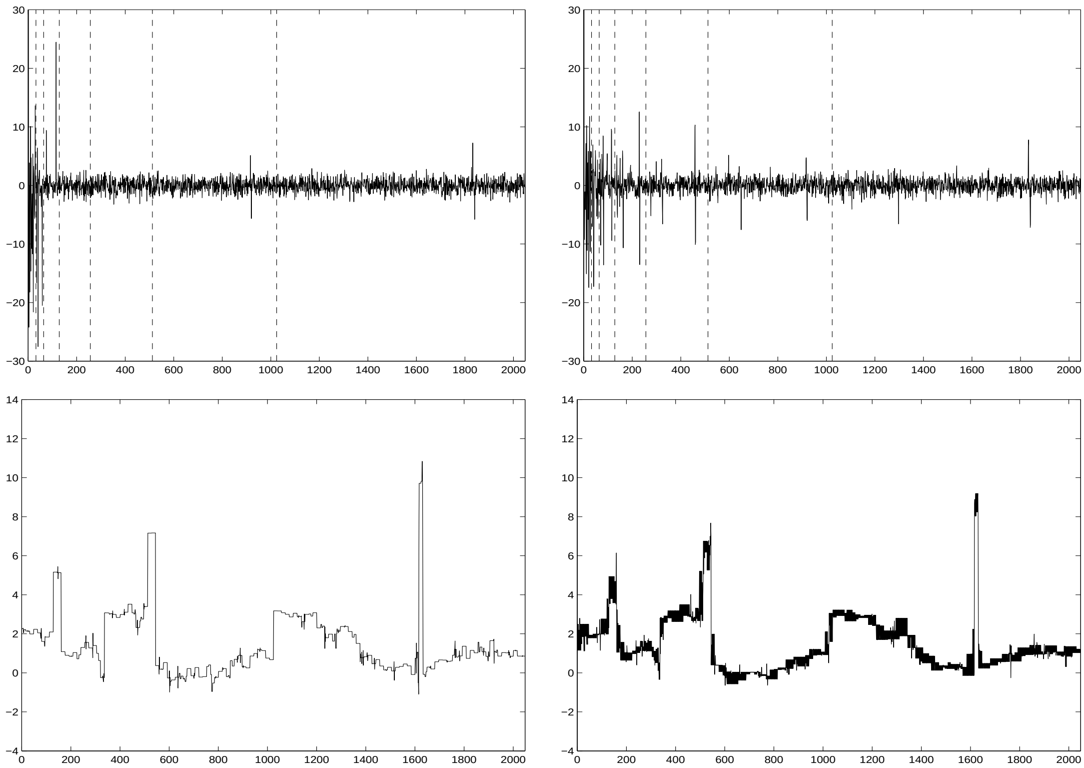
10.5 Exercises

1. We take the test signal from Figure 10.11 with white noise. We give in Figure 10.18 the wavelet transform coefficients for the Haar wavelet, thus for filter coefficients $h = \{1,\ 1\}$ (left) and the wavelet transform coefficients with the coefficients $h = \{1,\ 0,\ 0,\ 1\}$. We then denoise by setting all the wavelet coefficients on the 4 lowest scales to zero when they are in absolute value less than 1. At the bottom row the reconstructed signals are shown. Explain the differences.

2. Prove that if the variance of the signal is a constant, then the variance of the wavelet transform is a constant in each resolution level. Is this true for any wavelet transform (orthogonal, biorthogonal,…)?

3. Consider a signal s with correlation matrix C and suppose S is the 1D wavelet transform of s and that S has correlation matrix D. Prove that D is a 2D wavelet transform of C. Is it a square or a rectangular transform?

Figure 10.18: Haar and dilated-Haar transform and reconstruction after thresholding of a noisy signal



Chapter 11

Software and internet

There are several software packages that deal with wavelets

1. MATLAB Wavelet Toolbox This toolbox is based on the book by Strang and Nguyen [21]. `http://saigon.ece.wisc.edu/~waveweb/Tutorials/book.html` `http://www.mathworks.com/products/wavelet/`

2. WaveLab MATLAB package for various wavelet manipulations. Includes reference to Mallat's book [16]. `http://www-stat.stanford.edu/~wavelab/`

3. WaveBox MATLAB package for various wavelet manipulations. `http://www.wavbox.com/`

4. RWT (Rice MATLAB Wavlet toolbox) MATLAB package for wavelets and filter banks. `http://www-dsp.rice.edu/software/RWT/`

5. LiftPack C package for lifting scheme and several wavelet manipulations. `http://www.cs.sc.edu/~fernande/liftpack/`

6. WAILI (Wavelet transform with integer lifting) Implements a basic library in C++. Especially 2D wavelet transforms for image processing. Can also handle large images. `http://www.cs.kuleuven.ac.be/~wavelets/`

Several individual "waveletters" or institutions have collected a whole lot of information on the internet. Their web pages collect links to on-line wavelet introductions, wavelet bibliography, java-applets and other interactive demonstration software, homepages of wavelet people, preprint servers, etc. The above software links and some examples of waveletters home pages and many other sites, some of them interactive with e.g. wavelet applets, can be found on the course's home page. Please consult

`http://www.cs.kuleuven.ac.be/~ade/WWW/WAVE/`

Bibliography

[1] A.N. Akansu and R.A. Haddad. *Multiresolution signal decomposition: transforms, subbands and wavelets.* Academic Press/Harcourt Brace Jovanovich, 1992.

[2] C.S. Burrus, R.A. Gopinath, and H. Guo. *Introduction to Wavelets and Wavelet Transforms: A Primer.* Prentice Hall, 1998.

[3] R. Calderbank, I. Daubechies, W. Sweldens, and B.-L. Yeo. Losless image compression using integer to integer wavelet transforms. In *International Conference on Image Processing (ICIP), Vol. I*, pages 596–599. IEEE Press, 1997.

[4] R. Calderbank, I. Daubechies, W. Sweldens, and B.-L. Yeo. Wavelet transforms that map integers to integers. *Appl. Comput. Harmonic Anal.*, 5(3):332–369, 1998.

[5] S.S. Chen, D.L. Donoho, and M.A. Saunders. Atomic decomposition by basis pursuit. Technical report, Dept. of Statistics, Stanford Univ., February 1996.

[6] C.K. Chui. *An Introduction to Wavelets*, volume 1 of *Wavelet Analysis and its Applications.* Academic Press, Boston, 1992.

[7] A. Cohen, I. Daubechies, and J.C. Feauveau. Biorthogonal bases of compactly supported wavelets. *Comm. Pure Appl. Math.*, 45(5):485–560, 1992.

[8] I. Daubechies and W. Sweldens. Factoring wavelet transforms into lifting steps. *J. Fourier Anal. Appl.*, 4:245–267, 1998.

[9] P. De Gersem, B. De Moor, and Moonen M. Applications of the continuous wavelet transform in the processing of musical signals. In *Proc. of the 13th International Conference on Digital Signal Processing (DSP97)*, pages 563–566, 1997.

[10] R.A. DeVore and B.J. Lucier. Wavelets. *Acta Numerica*, 1:1–56, 1992.

[11] S. Jaffard and Y. Meyer. Wavelet methods for pointwise regularity and local oscillations of functions. *Mem. Amer. Math. Soc.*, 123(587), 1996.

[12] M. Jansen, M. Malfait, and A. Bultheel. Generalized cross validation for wavelet thresholding. *Signal Processing*, 56:33–44, 1997.

[13] B. Jawerth and W. Sweldens. An overview of wavelet based multiresolution analyses. *SIAM Rev.*, 36(3):377–412, 1994.

[14] G. Kaiser. *A friendly guide to wavelets.* Birkhäuser Verlag, 1994.

[15] A.K. Louis, P. Maass, and A. Rieder. *Wavelets: Theory and applications.* John Wiley, 1997.

[16] S. Mallat, editor. *A wavelet tour of signal processing.* Academic Press, 1998.

[17] S. Mallat and Z. Zhang. Matching pursuit with time-frequency dictionaries. Technical report, Courant Institute of Mathematical Sciences, 1993.

[18] S.G. Mallat and S. Zhong. Characterization of signals from multiscale edges. *IEEE Trans. Patt. Anal. Machine Intell.*, 14(7):710–732, 1992.

[19] Y. Meyer. *Wavelets and operators*, volume 37 of *Cambridge Studies in Advanced Mathematics.* Cambridge University Press, 1992.

[20] L. Prasad and S.S. Iyengar. *Wavelet Analysis with Applications to Image Processing.* CRC Press, Boca Raton, Florida, 1997.

[21] G. Strang and T. Nguyen. *Wavelets and filter banks.* Wellesley-Cambridge Press, Box 812060, Wellesley MA 02181, fax 617-253-4358, 1996.

[22] W. Sweldens. The lifting scheme: A new philosophy in biorthogonal wavelet constructions. In A.F. Laine and M. Unser, editors, *Wavelet Applications in Signal and Image Processing III*, pages 68–79, 68-79, 1995. Proc. SPIE 2569.

[23] W. Sweldens and P. Schröder. Building your own wavelets at home. In *Wavelets in Computer Graphics*, ACM SIGGRAPH Course Notes. ACM, 1996.

[24] A. Teolis. *Computational signal processing with wavelets.* Birkhäuser Verlag, 1998.

[25] G. Van de Wouwer. *Wavelets for multiscale texture analysis.* PhD thesis, University of Antwerp, 1998.

[26] M. Vetterli and J. Kovacevic. *Wavelets and subband coding.* Applied Mathematics and Mathematical Computation Series. Prentice Hall, Englewood Cliffs, 1995.

[27] M.V. Wickerhauser. *Adapted wavelet analysis from theory to software.* A.K. Peters, 289 Linden Street, Wellesley, MA 02181, 1994.

[28] M.V. Wickerhauser and R.R. Coifman. Entropy based methods for best basis selection. *IEEE Trans. Inf. Th.*, 38(2):719–746, 1992.

List of Acronyms

BER	Bit Error Rate
BIBO	Bounded Input Bounded Output
CDF	Cohen-Daubechies-Feauveau (wavelets)
CWT	Continuous Wavelet Transform
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
DWT	Discrete Wavelet Transform
ECG	ElectroCardioGram
FDMA	Frequency Division Multiple Access
FFT	Fast Fourier Transform
FIR	Finite Impulse Response
FWT	Fast Wavelet Transform
GCD	Greatest Common Divisor
GCI	Glottal Closure Instances
GCV	Generalized Cross Validation
GIS	Geographic Information System
HDTV	High Definition TeleVision
IIR	Infinite Impulse Response
MRA	MultiResolution Analysis
MSE	Mean Square Error
NMR	Nuclear Magnetic Resonance
OWT	Overcomplete Wavelet Transform
PR	Perfect Reconstruction
PSNR	Peak Signal to Noise Ratio
QMF	Quadrature Mirror Filter
RWT	Redundant Wavelet Transform
SNR	Signal to Noise Ratio
STFT	Short Time Fourier Transform
SURE	Stein Unbiased Risk Estimate
TDMA	Time Division Multiple Access
WFT	Windowed Fourier Transform
WPMA	Wavelet Packet Multile Access

Index