WAVELETS

Jöran Bergh Fredrik Ekstedt Martin Lindberg

February 3, 1999

– Can't you look for some money somewhere? Dilly said.

Mr Dedalus thought and nodded.

– I will, he said gravely. I looked all along the gutter in O'Connell street. I'll try this one now.

James Joyce, *Ulysses*

Preface

Why write another book on wavelets?

Well, we think that presently the treatises on wavelets fall broadly into either the monograph or the handbook category. By a monograph we mean a comprehensive and fully referenced text, while by a handbook we mean a bare collection of recipes or algorithms.[1]

Thus, we perceived a need for an interpolation: a not too demanding text for those who wish to get acquainted with the basic mathematical ideas and techniques appearing in wavelet analysis, and at the same time get some perspective of in which contexts, and how, the theory is presently applied.

The mathematical prerequisites needed for the reader are, in our minds, on the advanced undergraduate level: we assume a working familiarity with linear algebra, Fourier series and integrals. However, to better justify the theory, also knowledge about convergence results for (Lebesgue) integrals would have been desirable.

The prerequisites consequently presented us with a dilemma. Mathematical presentations should be made with precision, and scrupulous attention to precision makes for too heavy demands on the reader we had in mind.

Our resolution of this dilemma was to suppress some of the mathematical detail, and then refer the reader to more comprehensive texts. Our aim has been to focus on the key ideas and techniques, and this has brought us to the compromise you now have before you.

It seems unavoidable to make some errors of judgement when selecting what to include and what to exclude. We have certainly made such errors. We would be very grateful for every suggestion for improvement in this - and any other - respect.

Göteborg in January 1999, JB FE ML

[1] In addition, there have appeared a few books on the subject aimed at the general public.

An Outline of the Contents

The book is divided into two parts: the first devoted to theory and the second to applications. The chapters on filter banks and multiresolution analysis are basic. Subsequent chapters are built on these, and can be read independently of each other. There is also an introduction, in which we roughly sketch some key ideas and applications.

In the theoretical part, we first review the foundations of signal processing, and go on with filter banks. The main chapter of this part treats multiresolution analysis and wavelets. We end the first part with chapters on wavelets in more than one dimension, lifting, and the continuous wavelet transform.

In the applications part, we first present some of the most well-known wavelet bases. Next, we discuss adaptive bases, compression and noise reduction, followed by wavelet methods for the numerical treatment of, *i.a.*, partial differential equations. Finally, we describe differentiability in wavelet representations, an application of the continuous wavelet transform, feature extraction, and some implementation issues.

At the conclusion of most chapters, we have put a section with suggestions for further reading. The suggestions are meant to give starting points to expand the subject matter, and/or obtain a better mathematical understanding.

Throughout, we have inserted exercises, which are meant to complement the text, and to provide the reader with opportunities to make a few manipulations to consolidate the intended message.

Contents

Chapter 1

Introduction

We here ask the reader to think of, for example, a sound signal as recorded by a microphone. We do this, since for most people it seems helpful to have a specific concrete application in mind, when trying to get acquainted with unfamiliar mathematical concepts.

In this introduction, we will attempt a first approximative answer to the following questions, making some comparisons with the classical Fourier methods, and presenting a few basic examples. In subsequent chapters, we intend to answer them in greater detail.

What are wavelets and when are they useful?

Wavelets may be seen as a complement to classical Fourier decomposition methods.

Classically, a signal (function) may be decomposed into its independent Fourier modes, each mode with a distinct frequency but having no specific localization in time.

Alternatively, the signal may be decomposed into its independent wavelet modes, each mode belonging mainly in a frequency band and having a certain localization in time. (This is related to – but different from – windowed Fourier transform methods.)

Thus, the wavelet decomposition trades the sharply defined frequency in the Fourier decomposition for a localization in time.

We define the Fourier transform of a function f by

$$\widehat{f}(\omega) = \int_{-\infty}^{\infty} f(t) e^{-i\omega t}\, dt = \langle f, e^{i\omega \cdot} \rangle$$

where $\langle f, g \rangle = \int f(t)\overline{g(t)}\, dt$.

Under certain mild conditions on the function f we have both a Fourier decomposition

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \langle f, e^{i\omega \cdot} \rangle \, e^{i\omega t} \, d\omega$$

and a wavelet decomposition

$$f(t) = \sum_{j,k=-\infty}^{\infty} \langle f, \psi_{j,k} \rangle \, \psi_{j,k}(t)$$

where $\psi_{j,k}(t) = 2^{j/2}\psi(2^j t - k)$ all are translations and dilations of the same function ψ.

In general, the function ψ is more or less localized both in time and in frequency, and with $\int \psi(t)\, dt = 0$: a cancellation/oscillation requirement. If ψ is well localized in time it has to be less so in frequency, due to the following inequality (coupled to Heisenberg's uncertainty relation in quantum mechanics).

$$\int |\psi(t)|^2 \, dt \le 2 \left(\int |t\psi(t)|^2 \, dt \right)^{1/2} \left((2\pi)^{-1} \int |\omega \widehat{\psi}(\omega)|^2 \, d\omega \right)^{1/2} \tag{1.1}$$

Roughly put, the contributions are, respectively,

$$\frac{1}{2\pi} \int_{I_j} \langle f, e^{i\omega \cdot} \rangle \, e^{i\omega t} \, d\omega$$

and

$$\sum_{k=-\infty}^{\infty} \langle f, \psi_{j,k} \rangle \, \psi_{j,k}(t)$$

from the frequency (octave) band $I_j = \{\omega;\ 2^{j-1} + 2^{j-2} < |\omega/\pi| < 2^j + 2^{j-1}\}$.

In the latter sum, each term is localized around $t = 2^{-j}k$ if $\psi(t)$ is localized around $t = 0$. The frequency content of $\psi_{j,k}$ is localized around $\omega = 2^j\pi$ if the funtion ψ has its frequency content mainly in a neighbourhood of $\omega = \pi$. (Recall that $\int \psi(t)\, dt = 0$ means $\widehat{\psi}(0) = 0$.) For the Haar wavelet, shown in Figure 1.1, the modulus of the Fourier transform is $4(\sin \omega/4)^2/\omega,\ \omega > 0$, illustrating this.

In contrast, the harmonic constituents $e^{i\omega t}$ in the Fourier representation have a sharp frequency ω, and no localization in time at all.

Thus, if $\langle f, e^{i\omega \cdot} \rangle$ is perturbed for some ω in the given frequency band I_j, then this will influence the behaviour at all times.

Conversely, if $\langle f, \psi_{j,k} \rangle$ is perturbed, then this will influence the behaviour in the given frequency band I_j mainly, and in a neighbourhood of $t = 2^{-j}k$ with a size comparable to 2^{-j} mainly.

Exercises 1.0

1.1. Prove inequality (1.1), using the identity

$$\psi(t) = D(t\psi(t)) - tD\psi(t)$$

followed by a partial integration, and an application of the the Cauchy-Schwarz inequality

$$|\langle f, g\rangle| \leq \|f\| \, \|g\|$$

where $\|f\|$ is defined by

$$\|f\| = \left(\int_{-\infty}^{\infty} |f(t)|^2 \, dt \right)^{1/2}$$

1.1 The Haar Wavelet and Approximation

The wavelet expansion converges to a given function, under certain mild conditions. The general strategy in our illustration here is first to indicate the approximation (in what sense we do not enter into here) of the given function in terms of the *scaling function* on a certain fine enough scale, and then express this chosen approximating function in terms of the *wavelet*.

Our presentation will also provide an argument for having a wavelet expansion of a given function in terms of wavelets, which all have integral 0, although this may seem contradictory at a first glance.

We will consider the *Haar wavelet* system in a simple case. The Haar wavelet and the corresponding *Haar scaling function* are, respectively, the functions ψ and φ shown in Figure 1.1. Both functions are 0 outside the interval $(0, 1)$.

Note the two fundamental scaling relations expressing the functions on one scale in terms of the scaling function φ on the halved scale.

$$\begin{aligned} \varphi(t) &= \varphi(2t) + \varphi(2t-1) \\ \psi(t) &= \varphi(2t) - \varphi(2t-1) \end{aligned}$$

Now φ and ψ are orthogonal, as well as $\{\varphi(t-k)\}_k$, and $\{\psi(t-k)\}_k$, in the scalar product $\langle f, g\rangle := \int f(t)\overline{g(t)}\, dt$.

Let $f(t) = t^2$, $0 < t < 1$, and $f(t) = 0$ elsewhere. We may approximate f by its mean values over the dyadic intervals $(k\,2^{-j}, (k+1)2^{-j})$. For $j = 2$ this is shown in Figure 1.2.

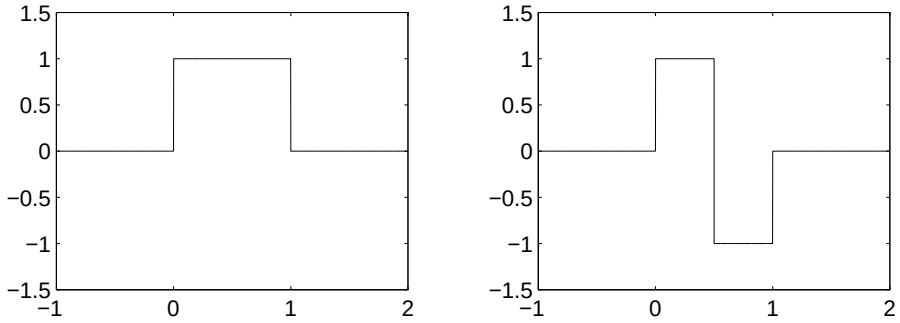

Figure 1.1: The functions φ (left) and ψ (right).

The approximations are integer translates of dilated scaling functions. The second mean value from the left is

$$\langle f, 2\varphi(2^2 \cdot -1)\rangle = \int_0^1 t^2 2\varphi(2^2 t - 1)\, dt$$

where the dilated scaling function is normalized to have its square integral equal to 1.

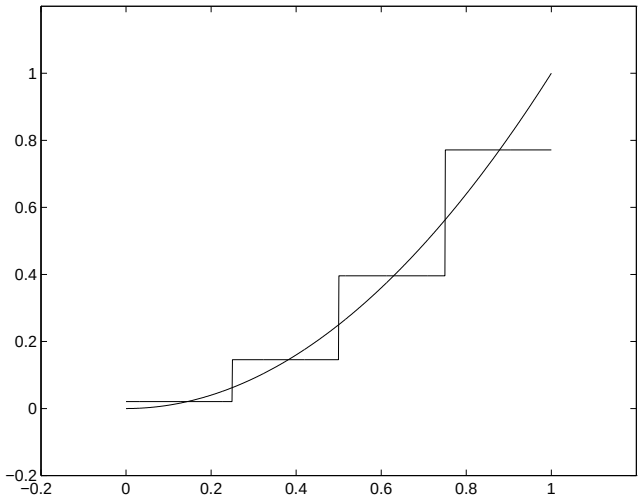

Figure 1.2: The function and its chosen step approximation.

Clearly, this approximation gets better with increasing j, and, choosing j large enough (fine enough scale), can be made as close as we wish (in L^2).

Now, for illustration purposes, we start from the (arbitrarily chosen) approximation case $j = 2$, as in Figure 1.2. Now comes a crucial step. We may

express the approximating step function with its mean values over the dyadic intervals with twice the length $2^{-(j-1)} = 2^{-1}$ and, at the same time, record the difference between the two approximating step functions. Note that the difference will be expressed in terms of the Haar wavelet on the same doubled scale in Figure 1.3.

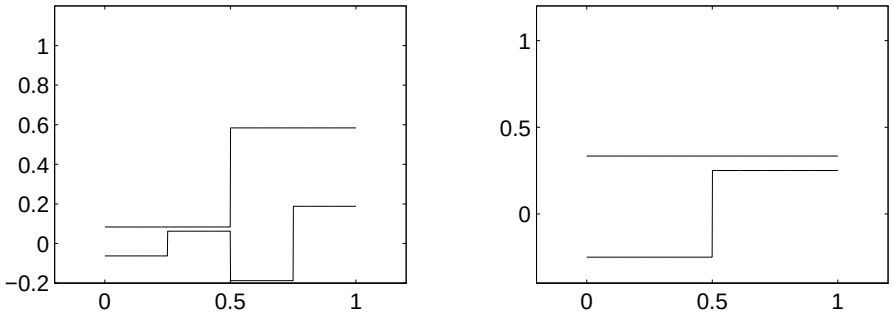

Figure 1.3: The first two successive means and differences at doubled scales.

We may repeat this procedure as many times as we please. When we take mean values over the intervals of length $2^j, j \leq 0$, only one value will be different from 0: $2^j/3$. See Figure 1.4. Correspondingly, the contribution in the wavelet expansion from these scales tend to 0 (in L^2) as $j \to -\infty$.

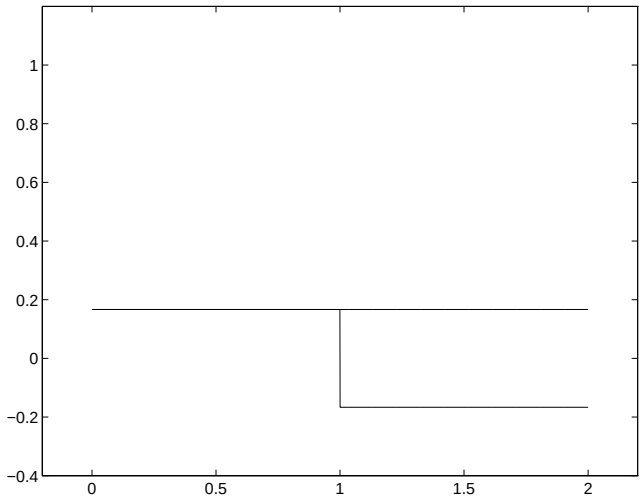

Figure 1.4: The third successive means and differences.

1.2 An Example of a Wavelet Transform

We have (rather arbitrarily) pasted together a function shown in Figure 1.5. The function consists of an sinusoidal part, a stepwise constant part, and a parabolic part.

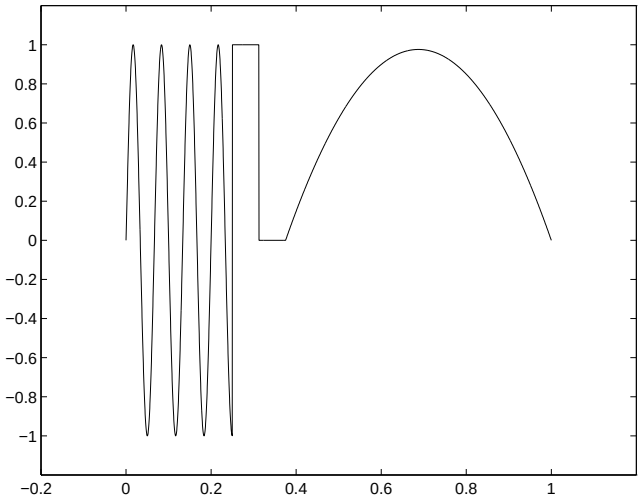

Figure 1.5: The given function.

The function is then decomposed in the plot to the left in Figure 1.6 into independent (orthogonal) parts, *a Multi-Resolution Decomposition*, each part essentially (but not exactly) in a frequency octave. In the plot to the right, the wavelet coefficients are shown correspondingly.

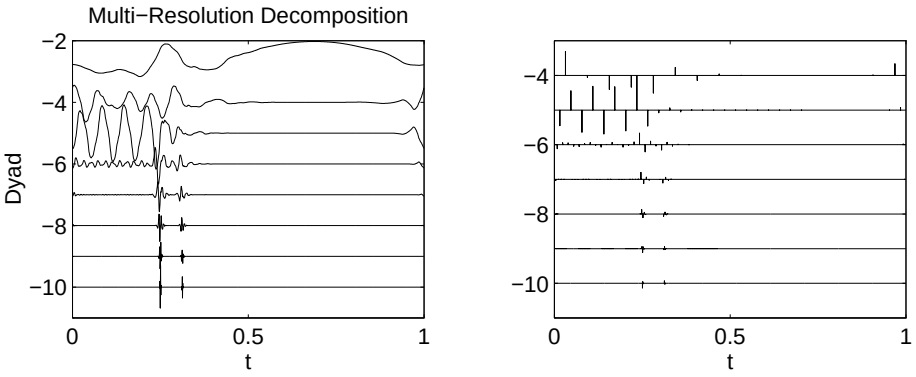

Figure 1.6: The decomposition (left) and the coefficients (right).

The horizontal axes are time, and the lowest graph represents the up-

permost half of the available frequency range, the next graph represents the upper half of the remaining lower half of the frequency range *etc.*

Note that the sharp changes in the function are clearly visible in the resolutions. A corresponding plot of the Fourier spectrum only reveals frequency peaks. Using time windows and doing Fourier transforms for each window will reveal the same features as the multiresolution decomposition, but choosing the right window size requires additional information in general.

Moreover, the number of operations in the multiresolution algorithm is linear in the number of samples of the signal, where the fast Fourier transform has an additional logarithmic factor.

1.3 Fourier *vs* Wavelet

We will now consider a sampled signal with the values $(0, 0, 1, 0)$ and period 4. We will compare the algorithmic implementations of the discrete Fourier and the wavelet methods and their respective frequency building blocks. In Exercise 1.4, we indicate how a translation influences the respective transforms: for the Fourier transform, we get a phase factor; for the wavelet transform, the effect of a translation is, in general, not transparent.

From a Fourier analysis aspect, this sequence may be taken as the values at the integers of the function

$$x(t) = 1/4 - 1/2 \cos \pi t/2 + 1/4 \cos \pi t$$

where the discrete Fourier sequence thus is $(1/4, -1/4, 1/4, -1/4)$: the first element, $1/4$, being the mean of the original sequence. These are calculated in the standard way: (x_n is the original sequence with $x_2 = 1$)

$$X_k = 1/4 \sum_{n=0}^{3} x_n e^{-2\pi i k n/4} \quad (k = 0, 1, 2, 3)$$

From a Haar wavelet analysis viewpoint, the sample sequence is encoded in the function

$$x(t) = 0\,\varphi(t) + 0\,\varphi(t-1) + 1\,\varphi(t-2) + 0\,\varphi(t-3)$$

where $\varphi(t) = 1$ $(0 < t < 1)$ and $= 0$ elsewhere. In Figure 1.7, the sample values thus are depicted to the right of their respective indices on the horizontal t axis.

We will now show how the different frequency bands contribute to the (different) functions.

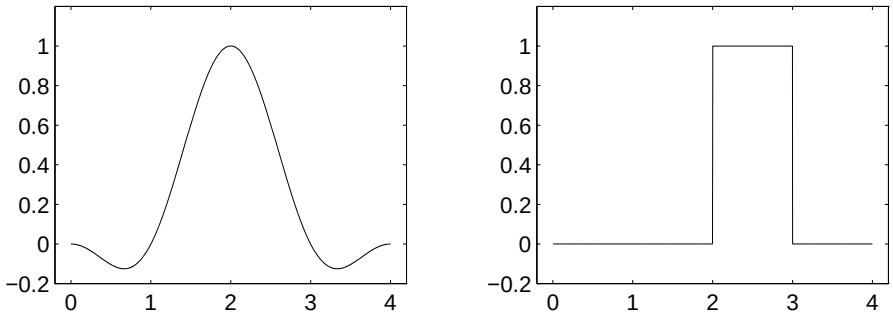

Figure 1.7: The Fourier (left) and the Haar (right) representations.

The Fourier component with highest frequency is $1/4 \cos \pi t$. The Haar wavelet component with highest frequency is obtained as follows.[1]

$$\begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

This measures the difference between the adjacent elements taken pairwise, and the resulting sequence is encoded in the function

$$Gx(t) = 0\,\psi(t/2) + 1\,\psi(t/2 - 1)$$

which is a high-frequency part, and where

$$\psi(t) = 1\,\varphi(2t) - 1\,\varphi(2t - 1)$$

is the Haar wavelet. Here the two coefficients ± 1 are the non-zero entries in the filter matrix. These two components are shown in Figure 1.8, where the localization is obvious in the Haar component, but it is not clear from the Fourier component.

The corresponding means are calculated in the analogous way.

$$\begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

This is encoded in the function

$$Hx(t) = 0\,\varphi(t/2) + 1/2\,\varphi(t/2 - 1)$$

[1] For notational simplicity, we have suppressed a normalizing factor $2^{1/2}$. See Exercise 1.2

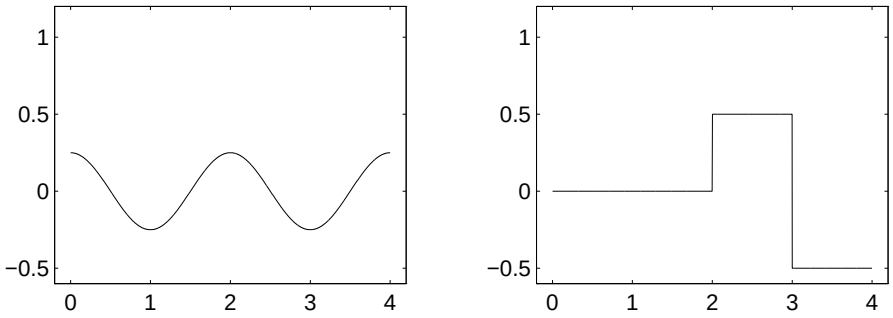

Figure 1.8: The Fourier (left) and the Haar (right) components.

which is the corresponding low-frequency part shown in Figure 1.9. (The factor 1/2 instead of the expected 1 comes from the suppressed normalizing factors mentioned before. See Exercise 1.2.)

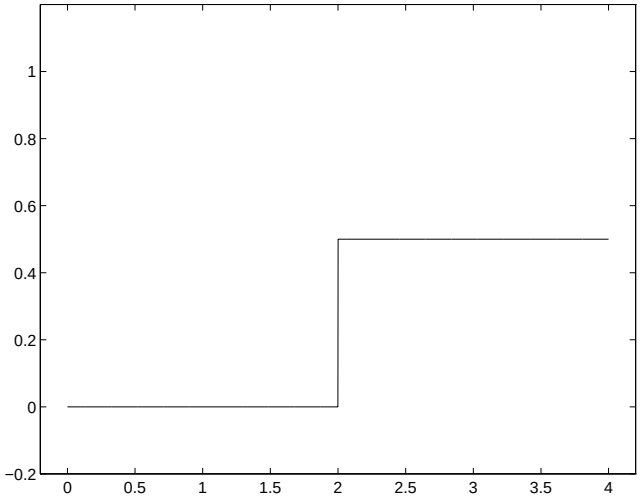

Figure 1.9: The corresponding mean (low-frequency) component Hx.

Denoting the filter matrices above with the same letters G, H, and their respective adjoints G^*, H^*, it is easy to verify that (I is the the identity matrix)

$$\begin{aligned} G^*G + H^*H &= 2I \\ GH^* = HG^* &= 0 \\ HH^* = GG^* &= 2I \end{aligned}$$

The first equation shows that we may recover the original sequence from the sequences encoded in Gx and Hx, and the middle two express that the

functions Gx and Hx are orthogonal. (Note that the encoded sequences are not orthogonal as such, but that this is an orthogonality relation between the columns of G and the columns of H).

For the next highest frequency, the Fourier component is $-1/2 \cos \pi t/2$. (Since the sequence is real-valued, X_1 is the complex conjugate to $X_{-1} = X_3$.) The corresponding Haar wavelet component is calculated from the previous level means:

$$\begin{bmatrix} 1 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} -1 \end{bmatrix}$$

This measures the difference between (adjacent pairs of) means. The resulting sequence is encoded in the function (normalizing factor suppressed; see Exercise 1.2)

$$-1/4\, \psi(t/4)$$

This are shown in Figure 1.10.

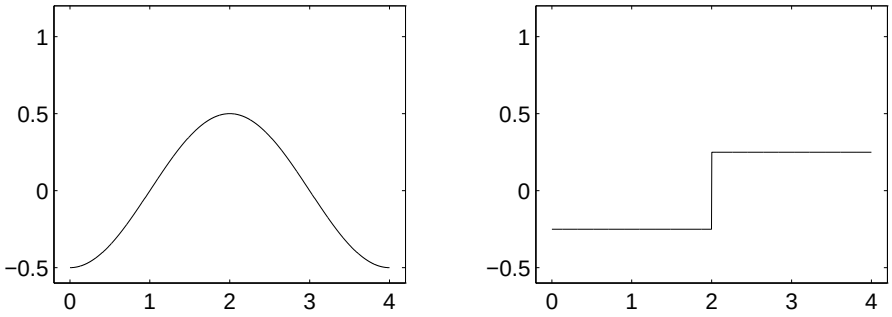

Figure 1.10: The next highest Fourier (left) and Haar (right) components.

The corresponding means are also calculated from the previous level means:

$$\begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \end{bmatrix}$$

This is encoded in the function $(0 < t < 4)$

$$\varphi(t/4) \equiv 1$$

which thus represents the mean value 1/4 (see Exercise 1.2) of the original sequence in Figure 1.11.

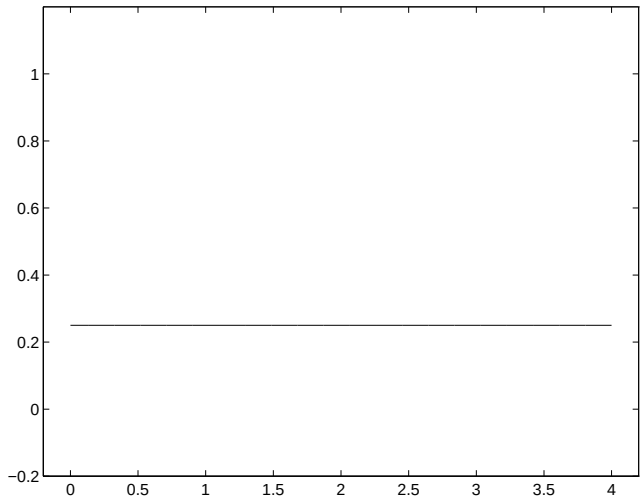

Figure 1.11: The mean in both Fourier and Haar representations.

The wavelet analysis may be seen as a successive procedure of taking mean values and differences at the double scale, recording the differences and repeating the process with the mean values. This may be repeated until the mean values are taken at a scale comparable to the length of the original sequence divided by the length of the filter. In the above example, we got exactly the mean value at the last step, as the length of the Haar filter is 2: the filter is represented by the two non-zero entries in the matrices G and H.

Exercises 1.3

1.2. Verify that the correct values appear when choosing the normalized functions in the encoding representations: for example, choosing the normalized $2^{1/2}\psi(2t-1)$ instead of $\psi(2t-1)$. 'Normalized' means that the integral of the squared function equals 1 (the L^2 norm is 1).

1.3. Compare the fast Fourier transform algorithm (FFT) and the above Haar wavelet transform, with regard to how the non-locality and the locality of the respective transform appear. This will already be apparent in the four-point case.

1.4. Work out what happens if another segment is chosen, that is, the four samples are $(0, 0, 0, 1)$ and $x_0 = x_1 = x_2 = 0,\ x_3 = 1$. Compare the influence on the Fourier and the wavelet representations.

1.4 Fingerprints and Image Compression

Storing fingerprints in an easily accessible archive means that the images have to be stored electronically, and in digital form. The grey-scale of the picture can be viewed as a function of two variables, $g(x, y)$. Of course, this function g has to be sampled into picture elements, pixels, which makes it defined on a plane array of, say, 512×512 points. These 512×512 greyscale values with a range of perhaps 64 shades available (as in Figure 1.12) then amounts to $2^{9+9+6} \approx 2$ Mbytes, just for one fingerprint.

Clearly, if the number of bits needed could be reduced to, say, 1%, then both storage, retrieval, and communication would come at a much reduced cost.

In fact, such a reduction or compression of the fingerprint information is presently done by the Federal Bureau of Investigation in the USA, using a wavelet technique. The success of the technique relies on the ability of the wavelets to detect and encode local changes in a function (picture) and this is exactly what constitutes the information of the fingerprint: the individual pattern of skin ridges on the fingertips in Figure 1.12.

Another related challenge is the efficient compression of video signals, *e.g.*, to facilitate public communication of videos in real time over the web. Algorithms for this compression are presently being commercially developed.

1.5 Noise Reduction

Another successful application of wavelet techniques is the suppression of noise in signals. In a sense, this is related to the picture compression with wavelets described in the previous section.

Considering, *e.g.*, *white noise*, this is usually thought of as a stochastic process, having realizations with a flat Fourier power spectrum. When the wavelets are independent as an orthonormal basis, this translates to the analoguous property for the wavelet coefficients. Thus, if the noise to be suppressed is known to be white, then a simple thresholding of the wavelet coefficients is a successful procedure in practice. (Thresholding means that all wavelet coefficients below the chosen threshold are put to 0.)

1.6 Notes

Wavelet analysis has been used in signal/image processing practice for less than two decades. Most of the mathematical ideas distinguishing wavelet analysis from classical Fourier analysis are less than a century old.

Figure 1.12: A fingerprint produced from a grey-scale digital representation.

A main reason for the applied mathematician's interest in wavelets is the increasing possibilities to perform calculations swiftly, *i.e.*, those which are realized in the development of yet more powerful computers.

For historical perspectives, we refer to the books by Meyer [24] [23], Daubechies [11], Kahane & Lemarié [21]. These books, excepting [24], give a comparatively fully detailed mathematical treatment of the theory.

The two books by Hubbard [19] and Meyer [24] are meant for a wider readership, as is an overview article by Jawerth and Sweldens [20].

There are also books with different emphases. We mention here the books by Strang and Nguyen [27] (signal processing), by Hernández and Weiss [16] (Fourier transform techniques), by Chui [7] [8] (splines, signal processing), and by Mallat [22] (signal processing).[2]

Information on the Web

There is much material on wavelets available on the Web/Internet. In particular, there is a periodical, Wavelet Digest, which is subscribable (free of charge), and a (free) MATLAB toolbox WAVELAB.

We refer the reader to the URL

`http://www.wavelet.org`

[2] These suggestions are not meant to be exhaustive.

Part I

Theory

Chapter 2

Signal Processing

In this chapter we review standard material from discrete-time signal processing. The concepts introduced here will be employed throughout Chapter 3 in the context of filter banks. Continuous-time signals (or functions) will be treated in Chapter 4, when we discuss wavelets. The first section defines discrete-time signals and filters. Then follow two sections on the Fourier and z-transform, and their applications to filter theory. Thereafter follows a section on linear phase and symmetric filters, which will play a key role later on in our study of symmetric wavelet bases. We end this chapter with a review of vector spaces, two-dimensional signal processing, and sampling of continuous-time signals.

2.1 Signals and Filters

A discrete-time *signal* x is a sequence of real or complex numbers

$$x = (x_k)_{k=-\infty}^{\infty} = (\dots, x_{-1}, x_0, x_1, \dots).$$

In most cases our signals will be real valued, but for the sake of generality we assume that they are complex valued. Mathematically, a signal is then a function $x : \mathbb{Z} \to \mathbb{C}$. Moreover, a signal $x \in \ell^2(\mathbb{Z})$ if it has finite energy, that is, $\sum_k |x_k|^2 < \infty$. In the next chapter we will have more to say about the space $\ell^2(\mathbb{Z})$ and discrete-time bases.

A *filter* H is an operator mapping an input signal x into an output signal $y = Hx$, and is often illustrated by a block diagram as in Figure 2.1. A filter H is *linear* if it satisfies the following two conditions for all input signals x and y, and real numbers a

$$H(x + y) = Hx + Hy, \tag{2.1a}$$

$$H(ax) = aHx. \tag{2.1b}$$

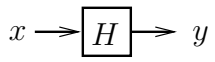

Figure 2.1: Block diagram of a filter.

A simple example of a linear filter is the delay operator D, which delays the input signal x one step. A delay of n steps is written D^n and is defined as

$$y = D^n x \quad \Leftrightarrow \quad y_k = x_{k-n}, \quad \text{for all } k \in \mathbb{Z}.$$

For a *time-invariant* (or shift-invariant) filter a delay in the input produces a corresponding delay in the output; so that for all input signals x

$$H(Dx) = D(Hx).$$

That is, the operator H commutes with the delay operator, $HD = DH$. From this it also follows that the filter is invariant to an arbitrary delay of n steps, $H(D^n x) = D^n(Hx)$.

Now, suppose H is a linear and time-invariant (LTI) filter. Let h be the output, or response, when the input is a *unit impulse*

$$\delta_k = \begin{cases} 1 & \text{for } k = 0, \\ 0 & \text{otherwise,} \end{cases}$$

that is, $h = H\delta$. The sequence (h_k) is called the *impulse response* of the filter. Since the filter is time-invariant we have

$$D^n h = H(D^n \delta).$$

So, if we write the input signal x of the filter as

$$x = \cdots + x_{-1}D^{-1}\delta + x_0\delta + x_1 D\delta + \cdots = \sum_n x_n D^n \delta,$$

and use the fact that the filter is linear, we can write the output signal $y = Hx$ as

$$\begin{aligned} y = H(\sum_n x_n D^n \delta) &= \sum_n x_n H(D^n \delta) \\ &= \sum_n x_n D^n h =: h * x. \end{aligned}$$

Here we have introduced a new operator, the *convolution* between h and x, defined by

$$y = h * x \quad \Leftrightarrow \quad y_k = \sum_n x_n (D^n h)_k = \sum_n x_n h_{k-n}. \tag{2.2}$$

From this we conclude that a LTI filter is uniquely determined by its impulse response, and that the output y always can be written as the convolution between the input x and the impulse response h,

$$y = Hx = h * x. \tag{2.3}$$

This demonstrates how the properties of linearity and time invariance motivates the definition of convolution. In the literature, it is common to restrict the meaning of the word filter to an operator that is both linear and time-invariant, and use the word operator in the more general case.

A *finite impulse response* (FIR) filter only has a finite number of coefficients different from zero. If a filter is not FIR, it is called *infinite impulse response* (IIR).

A LTI filter is *causal* if it satisfies $h_k = 0$ for $k < 0$. Non-causal filters are often referred to as non-realizable, since they require the knowledge of future values of the input signal. This is not necessarily a problem in applications, where the signal values might already be stored on a physical medium such as a CD-ROM. Further, FIR non-causal filters can always be delayed to make them causal. Later, when we develop the theory of filter banks and wavelets, it is convenient to work with non-causal filters.

The *correlation* $x \star y$ between two signals x and y is another sequence defined by

$$(x \star y)_k = \sum_n x_n y_{n-k} = \sum_n x_{n+k} y_n. \tag{2.4}$$

The result of correlating a signal with itself is referred to as the *autocorrelation* of the signal.

Example 2.1. An example of a LTI filter is the averaging filter defined by the impulse response

$$h_k = \begin{cases} 1/2 & \text{for } k = 0, 1, \\ 0 & \text{otherwise.} \end{cases}$$

This filter is FIR and causal. The output y is computed as the convolution between the input x and impulse response h,

$$\begin{aligned} y_k &= \sum_n h_n x_{k-n} \\ &= h_0 x_k + h_1 x_{k-1} \\ &= \frac{1}{2}(x_k + x_{k-1}). \end{aligned}$$

That is, the output is the time-average of the past two input values. □

Example 2.2. In filter banks the downsampling ($\downarrow 2$) and upsampling ($\uparrow 2$) operators are fundamental. The downsampling operator removes all odd-indexed values in a sequence, and the upsampling operator inserts a zero between every value in the sequence,

$$(\downarrow 2)x = (\dots, x_{-4}, x_{-2}, x_0, x_2, x_4, \dots),$$
$$(\uparrow 2)x = (\dots, 0, x_{-1}, 0, x_0, 0, x_1, 0, \dots).$$

These two operators are linear but *not* time-invariant. □

Exercises 2.1

2.1. Verify that the up- and downsampling operators are linear but not time-invariant.

2.2. A filter is *stable* if all bounded input signals produce a bounded output signal. A signal x is bounded if $|x_k| < C$ for all k and for some constant C. Prove that a LTI filter H is stable if and only if $\sum_k |h_k| < \infty$.

2.2 The z-transform

We will now introduce the z-transform, and later the Fourier transform. The action of filters in the time and frequency domain is fundamental in signal processing. Convolution in the time domain becomes a simple multiplication in the frequency domain. This is the key to the success of these transforms.

We define the z-transform of a discrete-time signal x by

$$X(z) = \sum_{k=-\infty}^{\infty} x_k z^{-k}, \quad z \in \mathbb{C}, \tag{2.5}$$

and occasionally we write this as $x \supset X(z)$. The series is convergent, and the function $X(z)$ is analytic for all complex z inside some annular domain, $r < |z| < R$, in the complex plane.

Example 2.3. Some simple sequences and their z-transforms are given by

1. $x = \delta$, $X(z) = 1$, (impulse)
2. $x = D^n\delta$, $X(z) = z^{-n}$, (delayed impulse)
3. $x_k = \begin{cases} 1, & k \geq 0, \\ 0, & k < 0, \end{cases}$ $X(z) = \dfrac{z}{z-1}$. (unit step)

□

Let us now see how different operations in the time domain are translated into the z-domain. A delay n steps corresponds to a multiplication by z^{-n}

$$x \supset X(z) \quad \Leftrightarrow \quad D^n x \supset z^{-n} X(z). \tag{2.6}$$

We will use the notation x^* to denote the time reverse of the signal x, $x_k^* = x_{-k}$, and we have

$$x \supset X(z) \quad \Leftrightarrow \quad x^* \supset X(z^{-1}). \tag{2.7}$$

The usefulness of the z-transform is largely contained in the *convolution theorem*. It states that convolution in the time domain corresponds to a simple multiplication in the z-domain.

Theorem 2.1. (**The convolution theorem**)

$$y = h * x \quad \Leftrightarrow \quad Y(z) = H(z)X(z). \tag{2.8}$$

□

The transform $H(z)$ of the impulse response of the filter is called the *transfer function* of the filter. This means that we can compute the output of a LTI filter by a simple multiplication in the z-domain. Often this is easier than directly computing the convolution. To invert the z-transform one usually uses tables, partial fraction expansion, and theorems. The correlation also has a corresponding relation on the transform side

$$y = x_1 \star x_2 \quad \Leftrightarrow \quad Y(z) = X_1(z)X_2(z^{-1}). \tag{2.9}$$

Example 2.4. Let us again consider the averaging filter from Example 2.1 given by $h_0 = h_1 = 1/2$. If we now compute the output in the z-domain we proceed as follows

$$\begin{aligned} H(z) &= \frac{1}{2} + \frac{1}{2}z^{-1}, \\ Y(z) &= H(z)X(z) = \frac{1}{2}X(z) + \frac{1}{2}z^{-1}X(z) \\ \Rightarrow \quad y_k &= \frac{1}{2}(x_k + x_{k-1}). \end{aligned}$$

Which is the same result as we obtained in the time domain. □

Exercises 2.2

2.3. Verify relations (2.6) and (2.7).

2.4. Prove that the correlation between between x and y can be written as the convolution between x and the time reverse y^* of y, that is, $x * y = x \star y^*$. Then prove relation (2.9).

2.3 The Fourier Transform

The Fourier transform of a signal gives us the frequency content of the signal. The frequency content often gives us valuable information about the signal not revealed in the time domain, such as the presence of oscillations. For filters, the Fourier transform of the impulse response tells us how different frequencies in a signal are amplified and shifted in phase.

The *discrete-time Fourier transform* is defined as

$$X(\omega) = \sum_{k=-\infty}^{\infty} x_k e^{-i\omega k}, \quad \omega \in \mathbb{R}. \tag{2.10}$$

It follows that $X(\omega)$ is 2π-periodic. Note that we, with an abuse of notation, use the same letter X to denote both the Fourier and z-transform of a signal x. From the context, and the different letters ω and z for the argument, it should be clear what we refer to. To obtain the signal values from the transform we use the inversion formula

$$x_k = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(\omega) e^{i\omega k} \, d\omega. \tag{2.11}$$

The *Parseval formula* tells us that the Fourier transform conserves the energy in the following sense

$$\sum_k |x_k|^2 = \frac{1}{2\pi} \int_{-\pi}^{\pi} |X(\omega)|^2 \, d\omega, \tag{2.12}$$

or more generally

$$\langle x, y \rangle = \sum_k x_k \overline{y_k} = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(\omega) \overline{Y(\omega)} \, d\omega. \tag{2.13}$$

From the definitions of the Fourier and z-transform, we see that we obtain the Fourier transform $X(\omega)$ from the z-transform $X(z)$ through the substitution $z = e^{i\omega}$. The convolution theorem for the z-transform therefore gives us a corresponding theorem in the frequency domain

$$y = h * x \quad \Leftrightarrow \quad Y(\omega) = H(\omega) X(\omega). \tag{2.14}$$

The Fourier transform $H(\omega)$ of the impulse response of a LTI filter is called the *frequency response* of the filter. An interesting property of LTI filters is that a pure frequency input produces a pure frequency output, but

with a different amplitude and phase. Let us see why this is so. If the input $x_k = e^{i\omega k}$, where $|\omega| \leq \pi$, the output y is

$$\begin{aligned} y_k &= \sum_n h_n x_{k-n} = \sum_n h_n e^{i\omega(k-n)} \\ &= e^{i\omega k} \sum_n h_n e^{-i\omega n} = e^{i\omega k} H(\omega). \end{aligned}$$

Write the complex number $H(\omega)$ in polar form, $H(\omega) = |H(\omega)|\, e^{i\phi(\omega)}$, and we get

$$y_k = |H(\omega)|\, e^{i(\omega k + \phi(\omega))}.$$

Thus, the output is also a pure frequency but with amplitude $|H(\omega)|$ and a phase delay of $-\phi(\omega)$. By plotting the *magnitude response* $|H(\omega)|$ and the *phase function* $\phi(\omega)$ for $|\omega| \leq \pi$, we see how the filter affects different frequency components of the signal. This is the reason for using the word filter in the first place; it filters out certain frequencies components of the input signal. A filter with magnitude response constant equal to one, $|H(\omega)| = 1$, is therefore called an *allpass* filter – all frequency components of the input signal are unaffected in magnitude (but not in phase).

Example 2.5. For the averaging (or lowpass) filter $h_0 = h_1 = 1/2$ we have

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

From this we see that the magnitude $|H(\omega)| = \cos(\omega/2)$ for $|\omega| < \pi$. To the left in Figure 2.2 the magnitude response is plotted. We see that high frequencies, near $\omega = \pi$, are multiplied by a factor close to zero and low frequencies, near $\omega = 0$, by a factor close to one. For the differencing (or highpass) filter

$$g_k = \begin{cases} 1/2 & \text{for } k = 0, \\ -1/2 & \text{for } k = 1, \\ 0 & \text{otherwise,} \end{cases}$$

we have $G(\omega) = (1 - e^{-i\omega})/2$. The magnitude response is plotted to the right in Figure 2.2. These two filters are the simplest possible examples of low- and highpass filters, respectively. □

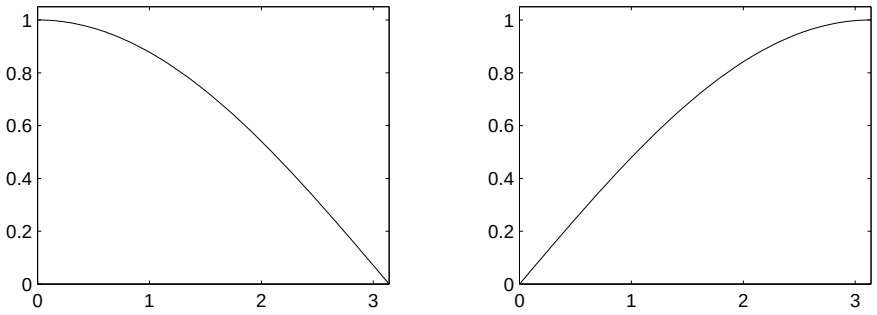

Figure 2.2: The magnitude responses of a low- and a highpass filter.

Example 2.6. The ideal lowpass filter suppresses frequencies above the cut-off at $\omega = \pi/2$ completely, and frequencies below this cut-off passes through unaffected. This filter is defined by the frequency response function

$$H(\omega) = \begin{cases} 1, & |\omega| < \pi/2, \\ 0, & \pi/2 < |\omega| < \pi. \end{cases}$$

It follows from the inversion formula (2.11) that the filter coefficients are samples of a sinc function

$$h_k = 2\,\mathrm{sinc}(k/2) = 2 \cdot \frac{\sin(\pi k/2)}{\pi k/2}. \tag{2.15}$$

□

Exercises 2.3

2.5. First, show that the filter coefficients of the ideal lowpass filter are given by (2.15). Then, compute the filter coefficients (g_k) of the ideal highpass filter

$$G(\omega) = \begin{cases} 0, & |\omega| < \pi/2, \\ 1, & \pi/2 < |\omega| < \pi. \end{cases}$$

2.4 Linear Phase and Symmetry

A filter has *linear phase* if its phase function $\phi(\omega)$ is a linear function. More generally, a filter also has linear phase if the phase function is piecewise linear with constant slope. The points of discontinuity of a linear phase filter occurs where $H(\omega) = 0$.

Linear phase is an important property in many applications, such as speech and sound processing, since a filter with linear phase delays different

frequency components an equal amount. Later, we will also see how linear phase in a filter bank corresponds to symmetric (and non-orthogonal) wavelets.

Example 2.7. From Example 2.5 we see that the phase of the lowpass filter $h_0 = h_1 = 1/2$ is

$$\phi(\omega) = -\omega/2, \quad |\omega| < \pi.$$

This is an example of a linear phase filter. □

If the filter coefficients of a filter H are symmetric around zero, so that $h_k = h_{-k}$, the frequency response will be real valued and even,

$$\begin{aligned} H(\omega) &= h_0 + h_1(e^{i\omega} + e^{-i\omega}) + h_2(e^{i2\omega} + e^{-i2\omega}) + \cdots \\ &= h_0 + 2h_1 \cos\omega + 2h_1 \cos 2\omega + \cdots . \end{aligned}$$

The filter then has zero phase, $\phi(\omega) = 0$. A filter that is antisymmetric around zero will have an imaginary, and odd, frequency response function with phase $\pi/2$ or $-\pi/2$. If $h_k = -h_{-k}$ we have $h_0 = 0$ and

$$\begin{aligned} H(\omega) &= h_1(-e^{i\omega} + e^{-i\omega}) + h_2(-e^{i2\omega} + e^{-i2\omega}) + \cdots \\ &= -2i(h_1 \sin\omega + h_2 \sin 2\omega + \cdots). \end{aligned}$$

Note that the sign of the factor $(h_1 \sin\omega + h_2 \sin 2\omega + \cdots)$ determines whether the phase is $\pi/2$ or $-\pi/2$, and that this depends on the frequency ω. ($-2i = 2e^{-i\pi/2}$.)

A causal filter can not have zero or constant phase. A causal filter can on the other hand be symmetric or antisymmetric, but not around zero. Causal filters have linear phase when they are symmetric or antisymmetric: $h_k = h_{N-k}$ or $h_k = -h_{N-k}$, respectively. Here we have assumed that we have a FIR filter with nonzero coefficients $h_0, h_1, \ldots, h_N$. We will then have a factor $e^{-iN\omega/2}$ in $H(\omega)$, and we see the linear term $-N\omega/2$ in the phase.

Example 2.8. The FIR filter with nonzero coefficients $h_0 = h_2 = 1/2$ and $h_1 = 1$ is symmetric, and

$$\begin{aligned} H(\omega) &= \frac{1}{2} + e^{-i\omega} + \frac{1}{2}e^{-i2\omega} \\ &= e^{-i\omega}(1 + \cos\omega). \end{aligned}$$

The filter has linear phase, $\phi(\omega) = -\omega$, since $1 + \cos\omega \geq 0$ for all ω. □

Recall that the frequency responses of symmetric and antisymmetric filters were even and odd, respectively; that is, $H(\omega) = \pm H(-\omega)$. For the z-transform this means that

$$H(z) = \pm H(z^{-1}).$$

Conclusion: for symmetric and antisymmetric filters the zeros of $H(z)$ must come in pairs as z_i and z_i^{-1}. When we construct wavelets later in this book, we will see that symmetric wavelets corresponds to symmetric filters in a filter bank.

The *group delay* of a filter is defined as

$$\tau(\omega) = -\frac{d\phi}{d\omega},$$

where $\phi(\omega)$ is the phase of the filter. The group delay measures the delay at the frequency ω.

Example 2.9. Suppose that the input x of the linear phase filter in Example 2.8 equals the sum of two pure frequencies, $x_k = e^{i\omega_1 k} + e^{i\omega_2 k}$. Since $\phi(\omega) = -\omega$ the group delay $\tau(\omega) = 1$ and the output y then equals

$$\begin{aligned} y_k &= H(\omega_1)e^{i\omega_1 k} + H(\omega_2)e^{i\omega_2 k} \\ &= |H(\omega_1)|\, e^{-i\omega_1} e^{i\omega_1 k} + |H(\omega_2)|\, e^{-i\omega_2} e^{i\omega_2 k} \\ &= |H(\omega_1)|\, e^{i\omega_1(k-1)} + |H(\omega_2)|\, e^{i\omega_2(k-1)}. \end{aligned}$$

We see that the two oscillations are both delayed one step. □

If the group delay is constant, different frequencies are delayed an equal amount. The filter then necessarily has linear phase.

Exercises 2.4

2.6. Show that the frequency response of the highpass filter $g_0 = 1/2$ and $g_1 = -1/2$ can be written as

$$G(\omega) = ie^{-i\omega/2} \sin(\omega/2).$$

Then compute and plot the magnitude and phase of $G(\omega)$. Note that the factor $\sin(\omega/2)$ is not positive for all $|\omega| < \pi$.

2.5 Vector Spaces

Our interest lies in the vector spaces $\mathbb{R}^n$ and $\mathbb{C}^n$. These are n-dimensional vector spaces over the real and complex numbers, respectively. A vector $x \in \mathbb{R}^n$, or $\mathbb{C}^n$, is represented as a column vector of n numbers, $x = (x_1 \ \dots \ x_n)^{\mathrm{T}}$. The *scalar product* of two vectors x and y is defined as

$$\langle x, y\rangle = x_1\overline{y_1} + \cdots + x_n\overline{y_n}, \tag{2.16}$$

and the vectors are *orthogonal* if $\langle x, y\rangle = 0$. The vectors $\varphi^{(1)}, \dots, \varphi^{(n)}$ is a *basis* of $\mathbb{R}^n$ (or $\mathbb{C}^n$) if every vector $x \in \mathbb{R}^n$ (or $\mathbb{C}^n$) can be written uniquely as

$$x = a_1\varphi^{(1)} + \cdots + a_n\varphi^{(n)}.$$

The numbers $a_1, \dots, a_n$ are the *coordinates* of x with respect to the basis $\varphi^{(1)}, \dots, \varphi^{(n)}$. A basis is *orthonormal* if

$$\langle \varphi^{(j)}, \varphi^{(k)}\rangle = \begin{cases} 1, & j = k, \\ 0, & j \neq k. \end{cases} \tag{2.17}$$

For an orthonormal basis the coordinates $a_k = \langle x, \varphi^{(k)}\rangle$, and then

$$x = \langle x, \varphi^{(1)}\rangle\varphi^{(1)} + \cdots + \langle x, \varphi^{(n)}\rangle\varphi^{(n)}.$$

The natural basis of $\mathbb{R}^n$ is orthonormal and given by the basis vectors

$$\delta_k^{(j)} = \begin{cases} 1, & k = j, \\ 0, & k \neq j. \end{cases}$$

The coordinates of a vector x with respect to this basis is $x_1, \dots, x_n$. If two vectors x and y have the coordinates $a_1, \dots, a_n$ and $b_1, \dots, b_n$ with respect to the orthonormal basis $\varphi^{(1)}, \dots, \varphi^{(n)}$, respectively, the scalar product between these two vectors can be computed as

$$\langle x, y\rangle = a_1\overline{b_1} + \cdots + a_n\overline{b_n}. \tag{2.18}$$

Example 2.10. In $\mathbb{R}^2$ the natural basis is given by two vectors: $\delta^{(1)} = (1, 0)^{\mathrm{T}}$ and $\delta^{(2)} = (0, 1)^{\mathrm{T}}$. Another orthonormal basis of $\mathbb{R}^2$ is the natural basis rotated 45 degrees counter-clockwise and has the basis vectors

$$\varphi^{(1)} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad \text{and} \quad \varphi^{(2)} = \frac{1}{\sqrt{2}}\begin{bmatrix} -1 \\ 1 \end{bmatrix}.$$

□

From elementary linear algebra we know that the coordinate transformation of a vector, with respect to two orthonormal bases, is represented by an orthogonal matrix. For assume that the vector $x \in \mathbb{R}^n$ has the coordinates $a = (a_1, \ldots, a_n)^{\mathrm{T}}$ and $\widetilde{a} = (\widetilde{a}_1, \ldots, \widetilde{a}_n)^{\mathrm{T}}$ with respect to the two bases $\varphi^{(1)}, \ldots, \varphi^{(n)}$ and $\widetilde{\varphi}^{(1)}, \ldots, \widetilde{\varphi}^{(n)}$, respectively. Then we can represent x in two ways,

$$x = a_1\varphi^{(1)} + \cdots + a_n\varphi^{(n)} = \widetilde{a}_1\widetilde{\varphi}^{(1)} + \cdots + \widetilde{a}_n\widetilde{\varphi}^{(n)}.$$

If we take the inner product of both sides of this equation with the basis vector $\varphi^{(j)}$ we obtain an expression for the coordinate a_j,

$$a_j = \sum_{k=1}^{n} \widetilde{a}_k \langle \widetilde{\varphi}^{(k)}, \varphi^{(j)} \rangle \quad \Leftrightarrow \quad a = P\widetilde{a}.$$

The equivalence follows from the definition of matrix multiplication, and the matrix P has elements $P_{jk} = \langle \widetilde{\varphi}^{(k)}, \varphi^{(j)} \rangle$. The matrix P is orthogonal since $PP^{\mathrm{T}} = P^{\mathrm{T}}P = I$. For a vector $x \in \mathbb{C}^n$ the above also holds true, but the matrix P is hermitian $PP^* = P^*P = I$. Here P^* is the adjoint, or the complex conjugate transpose, of P: $P^*_{jk} = \overline{P}_{kj}$. Note that for a hermitian, or orthogonal, matrix we have

$$\langle x, Py \rangle = \langle P^*x, y \rangle, \quad \text{for all } x, y \in \mathbb{C}^n.$$

Example 2.11. Let us see how to transform the coordinates of a vector $x \in \mathbb{R}^2$ from the natural coordinate system to the one given in Example 2.10. First we write x as

$$x = a_1\varphi^{(1)} + a_2\varphi^{(2)} = x_1\delta^{(1)} + x_2\delta^{(2)},$$

and then we compute the elements of P,

$$P_{ij} = \langle \delta^{(j)}, \varphi^{(i)} \rangle = \varphi^{(i)}_j,$$

to get the full transformation matrix

$$a = Px \quad \Leftrightarrow \quad \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}.$$

Note that it is very easy to find the transformation from a to x (since P is orthogonal), $x = P^{-1}a = P^{\mathrm{T}}a$. □

A very interesting and useful transformation on $\mathbb{C}^n$ is the *discrete Fourier transform*. For a vector $x \in \mathbb{C}^n$ the transform $X \in \mathbb{C}^n$ is defined as

$$X_j = \frac{1}{n} \sum_{k=0}^{n} x_k W^{-jk}, \quad \text{where } W = e^{i2\pi/n}.$$

To make this transformation orthogonal we should replace $1/n$ by $1/\sqrt{n}$, but the convention is to define it as we have. We obtain x from X through the inversion formula

$$x_k = \sum_{j=0}^{n} X_j W^{jk}.$$

2.6 Two-Dimensional Signal Processing

We will now extend some of the previous definitions and results to two dimensions. For instance, we will define the two-dimensional discrete Fourier transform.

In image processing it is natural to work with sampled functions of two variables. These functions are defined on the integer grid

$$\mathbb{Z}^2 := \{(k_x \; k_y)^{\mathrm{T}} : k_x, k_y \in \mathbb{Z}\}.$$

The integer grid $\mathbb{Z}^2$ can be seen as a subspace of $\mathbb{R}^2$ with integer elements, and an element $k \in \mathbb{Z}^2$ is sometimes called a multi-index. A grey-scale image can then be represented [1] as a function $f : \mathbb{Z}^2 \to \mathbb{R}$. Also, we write f_k rather than $f(k)$ for the values of f.

A two-dimensional filter H is an operator mapping an input function f into an output function g. Linearity for two-dimensional filters is defined exactly as in one dimension. The shift operator S^n, where $n = (n_x \; n_y)^{\mathrm{T}} \in \mathbb{Z}^2$, is defined by

$$g = S^n f \quad \Leftrightarrow \quad g_k = f_{k-n}, \quad \text{for all } k \in \mathbb{Z}^2.$$

For a shift-invariant filter

$$H(S^n f) = S^n(Hf), \quad \text{for all } n \in \mathbb{Z}^2,$$

and all functions f.

[1] Of course, an $N \times N$ image is nonzero only in a finite region, $0 \leq k_x, k_y < N$.

Suppose H is a linear and shift-invariant filter. Let h be the output when the input is a two-dimensional unit impulse

$$\delta_k = \begin{cases} 1 & \text{for } k = (0\ 0)^{\mathrm{T}}, \\ 0 & \text{otherwise,} \end{cases}$$

that is, $h = H\delta$. Just as for signals we can now write the output $g = Hf$ as a convolution (in two dimensions)

$$g = \sum_{n \in \mathbb{Z}^2} f_n S^n h =: h * f.$$

The two-dimensional discrete Fourier transform is defined as

$$F(\xi, \eta) = \sum_{k \in \mathbb{Z}^2} f_k e^{-i(\xi k_x + \eta k_y)}, \quad \xi, \eta \in \mathbb{R},$$

and we then have the 2-D convolution theorem:

$$g = h * f \quad \Leftrightarrow \quad G(\xi, \eta) = H(\xi, \eta) F(\xi, \eta).$$

Two-dimensional filters will be further discussed in Chapter 5 on wavelets in higher dimensions.

2.7 Sampling

Here we will briefly explain the mathematical basis of exchanging a function for its sample values at, say, the integers. The fundamental result is Poisson's summation formula.

For a continuous-time signal $f(t)$ we define its *Fourier transform* as

$$\widehat{f}(\omega) = \int_{-\infty}^{\infty} f(t) e^{-i\omega t}\, dt,$$

and the inverse transform is given by

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \widehat{f}(\omega) e^{i\omega t}\, d\omega.$$

Theorem 2.2 (Poisson's summation formula). Assume that the function f and its Fourier transform both are continuous and (for simplicity) have quadratic decay at infinity. Then holds

$$\sum_k \widehat{f}(\omega - 2k\pi) = \sum_l f(l) e^{-il\omega} \tag{2.19}$$

The *proof* is Exercise 2.8.

□

Now, if the function f has been lowpass filtered with cut-off frequency π, that is, $\widehat{f}(\omega) = 0$ for $\omega \geq \pi$, then holds in the pass-band $|\omega| < \pi$

$$\widehat{f}(\omega) = \sum_l f(l) e^{-il\omega}$$

Here we have complete information about the Fourier transform $\widehat{f}(\omega)$ in terms of the sample values $f(l)$. Thus, applying the inverse Fourier transform, we get a formula, the *Sampling Theorem*, reconstructing the function from its sample values.

$$\begin{aligned} f(t) &= 1/(2\pi) \int_{-\pi}^{\pi} \sum_l f(l) e^{-il\omega} e^{i\omega t} \, d\omega \\ &= \sum_l f(l) \sin \pi(t-l)/(\pi(t-l)) \end{aligned}$$

If the condition $\widehat{f}(\omega) = 0$ for $|\omega| \geq \pi$ is violated, then there will be interference (*alias effects*) from adjacent terms in (2.19), and $\widehat{f}(\omega)$ will not, in general, be retrievable from the sample values at the integers.

Exercises 2.7

2.7. Consider the function $f(t) = \sin \pi t$. What happens when this is sampled at the integers? Relate this to the conditions in the Sampling Theorem.

2.8. Prove Poisson's summation formula (2.19), noting that the left-hand side has period 2π, and that the right-hand side is a Fourier series with the same period.

2.9. Work out Poisson's summation formula (2.19) when the function $f(t)$ is sampled at the points $t = 2^{-J}k$. What should be the maximum cut-off frequency in this case? ($\omega = 2^J \pi$)

Chapter 3

Filter Banks

The study of filter banks in signal processing was one of the paths that led to wavelets. In signal processing one has for a long time developed different systems of basis functions to represent signals. In many applications it is desirable to have basis functions that are well localized in time *and* in frequency. For computational purposes these functions should also have a simple structure and allow for fast computations. Wavelets satisfy all these criteria.

The fast computation and representation of functions in wavelet bases is intimately connected to filter banks. As a matter of fact, the so-called fast wavelet transform is performed as a repeated application of the low- and highpass filters in a filter bank. Filter banks operate in discrete time and wavelets in continuous time. Wavelets are discussed in the next chapter.

3.1 Discrete-Time Bases

We would like to carry over the elements of linear algebra (see Section 2.5) in finite dimensions to infinite-dimensional signals. A basic question is to determine whether a sequence of signals $(\varphi^{(n)})_{n=-\infty}^{\infty}$ is a *basis*, that is, if all signals x can be written uniquely as

$$x = \sum_n c_n \varphi^{(n)}, \tag{3.1}$$

for some numbers (c_n). Here we encounter a problem not present in finite dimensions. The series expansion involves an infinite number of terms, and this series must converge. It turns out that we can not find a basis for all signals, and we will here restrict ourselves to those with finite energy. Then, we can carry over all the concepts from finite-dimensional vector spaces in a fairly straightforward way.

Convergence and Completeness

To discuss convergence we need a measure of the size of a signal. The norm of a signal provides us with such a measure. Just as in finite dimensions there are different norms, but the one we will use is the *energy norm*,

$$\|x\| = \Big(\sum_k |x_k|^2\Big)^{1/2}.$$

A vector space equipped with a norm is called a *normed space*. Now, a sequence of signals $(x^{(n)})_{n=1}^{\infty}$ is said to *converge* (in the energy norm) to x if

$$\big\|x^{(n)} - x\big\| \to 0, \quad \text{as } n \to \infty.$$

Accordingly, the equality in (3.1) means that the sequence of partial sums

$$s^{(N)} = \sum_{n=-N}^{N} c_n \varphi^{(n)},$$

is convergent with limit x, that is, $\big\|s^{(N)} - x\big\| \to 0$, as $N \to \infty$.

Remark. In the definition of convergence we assumed that we had a candidate for the limit of the sequence. It would be convenient to have a definition, or test, of convergence that does not involve the limit explicitly. A fundamental property of the real and complex number systems is that a sequence of numbers is convergent if and only if it is a Cauchy sequence. In a general normed space, a sequence $(x^{(n)})_{n=1}^{\infty}$ is a *Cauchy sequence* if for all $\epsilon > 0$ there is an N such that

$$\big\|x^{(n)} - x^{(m)}\big\| < \epsilon, \quad \text{for all } n, m > N.$$

Loosely speaking, a Cauchy sequence is a sequence where the vectors are getting closer and closer together, or a sequence that is trying to converge. In a general normed space all Cauchy sequences does not have to converge. An example is $\mathbb{Q}$, the set of all rational numbers. A normed space where all Cauchy sequences are convergent is called a *Banach space*, or a *complete* normed space. The finite dimensional vector spaces $\mathbb{R}^n$ and $\mathbb{C}^n$ are complete. The space of all signals with finite energy,

$$\ell^2(\mathbb{Z}) = \{x : \mathbb{Z} \to \mathbb{C} \mid \|x\| < \infty\},$$

is also complete. In this chapter we will assume that all signals are contained in this space. □

Hilbert Spaces and Orthonormal Bases

Just as in finite dimensions, it is convenient to work with orthogonal bases. To discuss orthogonality we need to impose further structure on our space $\ell^2(\mathbb{Z})$, by defining an inner product between two vectors. The *inner product* between two signals x and y in $\ell^2(\mathbb{Z})$ is defined as

$$\langle x, y \rangle = \sum_k x_k \overline{y_k}.$$

Convergence of this infinite sum follows from the *Cauchy-Schwarz inequality*,

$$|\langle x, y \rangle| \leq \|x\| \, \|y\| \, .$$

Note that we now can write the norm of $x \in \ell^2(\mathbb{Z})$ as

$$\|x\|^2 = \langle x, x \rangle.$$

A complete space with an inner product is called a *Hilbert space*. We have only looked at the particular Hilbert spaces $\mathbb{R}^n$, $\mathbb{C}^n$, and $\ell^2(\mathbb{Z})$ so far, but in the chapters on wavelets below we also work with the space $L^2(\mathbb{R})$ of all continuous-time signals with finite energy.

We now have all the tools necessary to define orthonormal bases for our signals. First, recall definition (3.1) of a basis. A basis $(\varphi^{(n)})_{n=-\infty}^{\infty}$ is *orthonormal* if

$$\langle \varphi^{(j)}, \varphi^{(k)} \rangle = \begin{cases} 1, & j = k, \\ 0, & j \neq k. \end{cases}$$

For an orthonormal basis the coordinates of a signal x are given by

$$c_n = \langle x, \varphi^{(n)} \rangle. \tag{3.2}$$

It follows that the signal can be written as

$$x = \sum_n \langle x, \varphi^{(n)} \rangle \varphi^{(n)}. \tag{3.3}$$

Equation (3.2) is referred to as an *analysis* and equation (3.3) as a *synthesis* of the signal x, respectively.

3.2 The Discrete-Time Haar Basis

How do we find bases for our space $\ell^2(\mathbb{Z})$, and what properties do we want these bases to have? Usually we want the basis functions to be well localized both in time and in frequency. The coordinates of a signal in the basis will then provide a measure of the strength of the signal for different time and frequency intervals. We will construct bases starting from two prototype basis functions and their even translates. These bases are further characterized by the fact that the coordinates in the new basis can be computed using a filter bank. This is another important property, since it means that the coordinate transformation can be computed efficiently.

The Basis Functions

The discrete-time Haar basis is an example of the special class of orthogonal bases that are related to filter banks. These orthogonal bases are characterized by the fact that they are formed as the even translates of two prototype basis functions φ and ψ. For the Haar basis these two functions are given by,

$$\varphi_k = \begin{cases} 1/\sqrt{2} & \text{for } k = 0, 1, \\ 0 & \text{otherwise,} \end{cases} \quad \text{and,} \quad \psi_k = \begin{cases} 1/\sqrt{2} & \text{for } k = 0, \\ -1/\sqrt{2} & \text{for } k = 1, \\ 0 & \text{otherwise.} \end{cases}$$

The basis functions $(\varphi^{(n)})$ are now formed as the even translates of these two prototypes (see Figure 3.1),

$$\varphi_k^{(2n)} = \varphi_{k-2n}, \quad \text{and,} \quad \varphi_k^{(2n+1)} = \psi_{k-2n}.$$

The coordinates of a signal x in this new basis are consequently,

$$\begin{aligned} y_n^{(0)} &:= c_{2n} = \langle x, \varphi^{(2n)} \rangle = \frac{1}{\sqrt{2}}(x_{2n} + x_{2n+1}), \\ y_n^{(1)} &:= c_{2n+1} = \langle x, \varphi^{(2n+1)} \rangle = \frac{1}{\sqrt{2}}(x_{2n} - x_{2n+1}), \end{aligned} \tag{3.4}$$

in other words, weighted averages and differences of pairwise values of x. Another way of interpreting this is to say that we take pairwise values of x, and then rotate the coordinate system in the plane ($\mathbb{R}^2$) 45 degrees counterclockwise. Here, we have also introduced the sequences $y^{(0)}$ and $y^{(1)}$ consisting of the even- and odd-indexed coordinates, respectively. The basis functions is an orthonormal basis of $\ell^2(\mathbb{Z})$, and we can therefore reconstruct

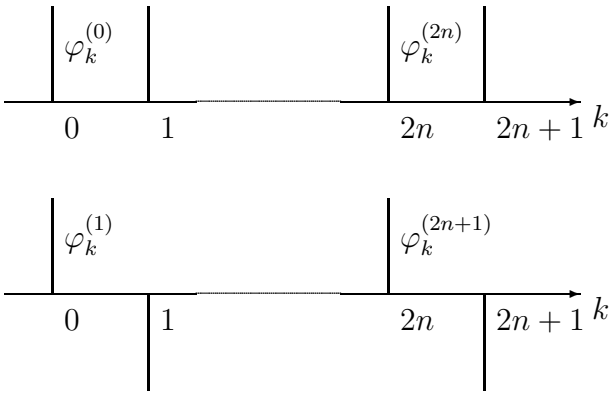

Figure 3.1: The discrete-time Haar basis.

the signal values as

$$\begin{aligned} x_k &= \sum_n c_n \varphi_k^{(n)} \\ &= \sum_n y_n^{(0)} \varphi_k^{(2n)} + \sum_n y_n^{(1)} \varphi_k^{(2n+1)} \\ &= \sum_n y_n^{(0)} \varphi_{k-2n} + \sum_n y_n^{(1)} \psi_{k-2n}. \end{aligned} \tag{3.5}$$

Analysis

We will now show how we can use a filter bank to compute the coordinates in (3.4). If we define the impulse responses of two filters H and G as

$$h_k = \varphi_k, \quad \text{and,} \quad g_k = \psi_k,$$

and if we let h^* and g^* denote the time-reverses of these filters, we can write the inner products in (3.4) as a convolution,

$$\begin{aligned} y_n^{(0)} &= \sum_k x_k \varphi_k^{(2n)} = \sum_k x_k h_{k-2n} = \sum_k x_k h_{2n-k}^* \\ &= (x * h^*)_{2n}. \end{aligned}$$

Similarly, we get $y_n^{(1)} = (x * g^*)_{2n}$.

The conclusion is that we can compute $y^{(0)}$ and $y^{(1)}$ by filtering x with H^* and G^*, respectively, and then downsampling the output of these two filters (see Figure 3.2). Downsampling removes all odd-indexed values of a signal, and we define the downsampling operator $(\downarrow 2)$ as

$$(\downarrow 2)x = (\dots, x_{-2}, x_0, x_2, \dots).$$

Thus, we have

$$y^{(0)} = (\downarrow 2)H^*x, \quad \text{and} \quad y^{(1)} = (\downarrow 2)G^*x.$$

Here, H^* and G^* denote the low- and highpass filters with impulse responses h^* and g^*, respectively. These two filters are non-causal since their impulse responses are the time-reverses of causal filters. This is not necessarily a problem in applications, where the filters can always be made causal by delaying them a certain number of steps; the output is then delayed an equal number of steps.

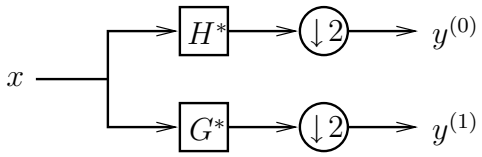

Figure 3.2: The analysis part of a filter bank.

Synthesis

So far we have seen how we can analyze a signal, or compute its coordinates, in the Haar basis using a filter bank. Let us now demonstrate how we can synthesize, or reconstruct, a signal from the knowledge of its coordinates. From the definition of the filters H and G, and from the reconstruction formula (3.5) we get

$$\begin{aligned} x_k &= \sum_n y_n^{(0)} \varphi_{k-2n} + \sum_n y_n^{(1)} \psi_{k-2n} \\ &= \sum_n y_n^{(0)} h_{k-2n} + \sum_n y_n^{(1)} g_{k-2n} \\ &= (v^{(0)} * h)_k + (v^{(1)} * g)_k, \end{aligned}$$

where $v^{(0)} = (\uparrow 2)y^{(0)}$ and $v^{(1)} = (\uparrow 2)y^{(1)}$, see Exercise 3.1. Here the upsampling operator $(\uparrow 2)$ is defined as

$$(\uparrow 2)y = (\dots, y_{-1}, 0, y_0, 0, y_1, \dots).$$

Hence, the signal x is the sum of to signals,

$$\begin{aligned} x &= v^{(0)} * h + v^{(1)} * g \\ &= H\left((\uparrow 2)y^{(0)}\right) + G\left((\uparrow 2)y^{(1)}\right) =: x^{(0)} + x^{(1)}. \end{aligned}$$

The signals $x^{(0)}$ and $x^{(1)}$ are obtained by first upsampling $y^{(0)}$ and $y^{(1)}$, and then filtering the result with H and G, respectively; see Figure 3.3. On the other hand, if we look back at the reconstruction formula (3.5), we can write x as

$$\begin{aligned} x &= x^{(0)} + x^{(1)} \\ &= \sum_n \langle x, \varphi^{(2n)} \rangle \varphi^{(2n)} + \sum_n \langle x, \varphi^{(2n+1)} \rangle \varphi^{(2n+1)}, \end{aligned}$$

which means that $x^{(0)}$ and $x^{(1)}$ are the orthogonal projections of x onto the subspaces spanned by the even and odd basis functions, respectively.

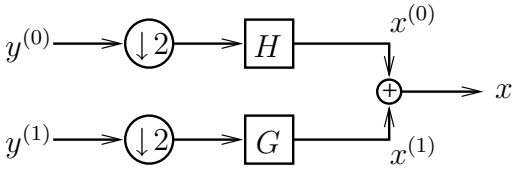

Figure 3.3: The synthesis part of a filter bank.

Exercises 3.2

3.1. Show that $(v * h)_k = \sum_n y_n h_{k-2n}$, where $v = (\uparrow 2)y$. Observe that $v_{2n} = y_n$.

3.3 The Subsampling Operators

In this section we study the effect of down- and upsampling on a signal. Specifically, we arrive at formulas for how the z- and Fourier transform of a signal change, when the signal is subsampled.

Downsampling

The downsampling operator $(\downarrow 2)$ removes all odd-indexed values of a signal, and is consequently defined as

$$(\downarrow 2)x = (\ldots, x_{-2}, x_0, x_2, \ldots).$$

If we let $y = (\downarrow 2)x$ we have $y_k = x_{2k}$, and in the z-domain we get (see Exercise 3.2)

$$Y(z) = \frac{1}{2}\left[X\left(z^{1/2}\right) + X\left(-z^{1/2}\right)\right]. \tag{3.6}$$

The corresponding relation in the frequency domain is

$$Y(\omega) = \frac{1}{2}\left[X\left(\frac{\omega}{2}\right) + X\left(\frac{\omega}{2} + \pi\right)\right].$$

From these relations we see that we get an *alias* component in the spectrum (frequency domain) of y from the term $X(\omega/2 + \pi)$. The filters before the upsampling operators will reduce this alias effect and if the filters are ideal low- and highpass filters, respectively, the alias component will be completely deleted.

Upsampling

The upsampling operator $(\uparrow 2)$ inserts a zero between every value of a signal,

$$(\uparrow 2)y = (\ldots, y_{-1}, 0, y_0, 0, y_1, \ldots).$$

If we let $u = (\uparrow 2)y$ it is easy to verify (see Exercise 3.2) that

$$U(z) = Y\left(z^2\right), \tag{3.7}$$

and for the Fourier transform we get

$$U(\omega) = Y(2\omega).$$

The spectrum $U(\omega)$ of u is thus a dilated version of $Y(\omega)$ by a factor two. This will result in the appearance of an image in $U(\omega)$. The low- and highpass filters after the upsampling reduce the effect of the image, and if the filters are ideal low- and highpass filters the image will be completely deleted.

Down- and Upsampling

By combining the results from the previous two sections, we obtain a relation between a signal x and the down- and upsampled signal

$$u = (\uparrow 2)(\downarrow 2)x = (\ldots, x_{-2}, 0, x_0, 0, x_2, \ldots).$$

In the z-domain we have

$$U(z) = \frac{1}{2}\left[X(z) + X(-z)\right],$$

and in the Fourier domain

$$U(\omega) = \frac{1}{2}\left[X(\omega) + X(\omega + \pi)\right].$$

Exercises 3.3

3.2. Prove relations (3.6) and (3.7).

3.3. Show that if $Y(z) = X\left(z^{1/2}\right)$ then $Y(\omega) = X(\omega/2)$, and if $U(z) = Y\left(z^2\right)$ then $U(\omega) = Y(2\omega)$.

3.4. Assume that a filter H has the transfer function $H(z)$. Show that the time-reverse H^* of the filter has the transfer function $H^*(z) = H\left(z^{-1}\right)$.

3.5. Consider a signal x with a 2π-periodic Fourier transform $X(\omega)$. Plot $X(\omega)$, $X(\omega/2)$, and $X(\omega/2 + \pi)$ so that you understand how the alias component appears in the downsampled signal $y = (\downarrow 2)x$.

Similarly, consider a signal y and plot $Y(\omega)$ and $U(\omega) = Y(2\omega)$, so that you see the appearance of the image in the spectrum of $u = (\uparrow 2)y$.

3.4 Perfect Reconstruction

Above we studied the Haar basis which is an example of a special type of discrete-time bases. These bases were characterized by the fact that they were formed as the even translates of two prototype basis functions. We saw how low- and highpass filters, followed by downsampling, in the analysis part of a filter bank gave us the coordinates of the signal in the new basis. Similarly, upsampling followed by low- and highpass filters gave us the expansion of the signal in the new basis.

A filter bank consists of an analysis and a synthesis part as depicted in Figure 3.4. The goal is to find conditions on the low- and highpass filters so that the output $\widehat{x}$ equals the input x. This is called *perfect reconstruction*. For a general filter bank, the impulse responses of the low- and highpass filters in the synthesis part equals two prototype basis functions in a corresponding discrete-time basis. If the basis is orthogonal, the analysis filters are the time-reverses of the synthesis filters. More generally, we can have a biorthogonal basis, and then the analysis filters are denoted as $\widetilde{H}$ and $\widetilde{G}$.

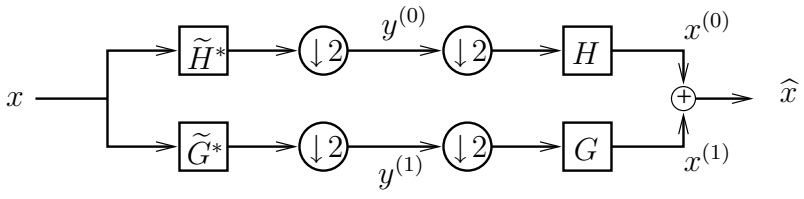

Figure 3.4: Filter bank.

Reconstruction Conditions

The results of the previous section for the down- and upsampling give us the following expressions for the z-transform of $x^{(0)}$ and $x^{(1)}$

$$X^{(0)}(z) = \frac{1}{2}H(z)\left[X(z)\widetilde{H}^*(z) + X(-z)\widetilde{H}^*(-z)\right],$$
$$X^{(1)}(z) = \frac{1}{2}G(z)\left[X(z)\widetilde{G}^*(z) + X(-z)\widetilde{G}^*(-z)\right].$$

Adding these together, we obtain an expression for the z-transform of $\widehat{x}$

$$\begin{aligned}\widehat{X}(z) = &\frac{1}{2}\left[H(z)\widetilde{H}^*(z) + G(z)\widetilde{G}^*(z)\right]X(z)\\ &+\frac{1}{2}\left[H(z)\widetilde{H}^*(-z) + G(z)\widetilde{G}^*(-z)\right]X(-z),\end{aligned}$$

where we have grouped the terms with the factors $X(z)$ and $X(-z)$ together, respectively.

From this we see that we get perfect reconstruction, that is $x = \widehat{x}$, if the factor in front of $X(z)$ equals one, and the factor in front of $X(-z)$ equals zero. We thus get the following two conditions on the filters

$$H(z)\widetilde{H}^*(z) + G(z)\widetilde{G}^*(z) = 2, \qquad \text{(no distortion)} \tag{3.8}$$

$$H(z)\widetilde{H}^*(-z) + G(z)\widetilde{G}^*(-z) = 0. \qquad \text{(alias cancellation)} \tag{3.9}$$

The first condition ensures that there is no distortion of the signal, and the second condition that the alias component $X(-z)$ is cancelled. These two conditions will appear again when we study wavelets. This is the key to the connection between filter banks and wavelets. Due to different normalizations, the right-hand side in the no distortion condition equals 1 for wavelets though.

Alias Cancellation and the Product Filter

At this stage we will define the highpass filters in terms of the lowpass filters, so that the alias cancellation condition (3.9) automatically is satisfied. We let the highpass filters equal

$$G(z) = -z^{-L}\widetilde{H}^*(-z), \quad \text{and} \quad \widetilde{G}(z) = -z^{-L}H^*(-z), \tag{3.10}$$

where L is an arbitrary odd integer. In Exercise 3.6 below you are to verify that this choice cancels the alias, and then you will also see why L has to be odd. In this book we will in most cases choose $L = 1$. Choosing the

highpass filters in this way is a sufficient condition to ensure alias cancellation. In Chapter 4 we motivate this choice further when discussing biorthogonal bases.

If we substitute the alias cancellation choice (3.10) into the no distortion condition (3.8), we get a single condition on the two lowpass filters for perfect reconstruction,

$$H(z)\widetilde{H}^*(z) + H(-z)\widetilde{H}^*(-z) = 2.$$

We now define the *product filter* $P(z) = H(z)\widetilde{H}^*(z)$, and this finally reduces the perfect reconstruction condition to

$$P(z) + P(-z) = 2. \tag{3.11}$$

The left hand side of this equation can be written

$$P(z) + P(-z) = 2p_0 + 2\sum_n p_{2n} z^{-2n}.$$

From this we conclude that all even powers in $P(z)$ must be zero, except the constant term which should equal one. The odd powers all cancel and are the design variables in a filter bank.

The design of a perfect reconstruction filter bank is then a question of finding a product filter $P(z)$ satisfying condition (3.11). Once such a product filter has been found, it is factored in some way as $P(z) = H(z)\widetilde{H}^*(z)$. The highpass filters are then given by equation (3.10).

Example 3.1. Let us see if the discrete-time Haar basis satisfies the perfect reconstruction condition. The filters are given by

$$H(z) = \widetilde{H}(z) = \frac{1}{\sqrt{2}}(1 + z^{-1}), \qquad G(z) = \widetilde{G}(z) = \frac{1}{\sqrt{2}}(1 - z^{-1}).$$

The product filter is then

$$\begin{aligned} P(z) &= H(z)\widetilde{H}^*(z) = H(z)\widetilde{H}(z^{-1}) \\ &= \frac{1}{2}(z + 2 + z^{-1}). \end{aligned}$$

This product filter indeed satisfies the perfect reconstruction condition, since all even powers equal zero except the constant term $p_0 = 1$. □

Orthogonal Filter Banks

In the first part of this chapter we saw that for orthogonal discrete-time bases the corresponding filters in the filter bank were related as

$$H(z) = \widetilde{H}(z), \quad \text{and} \quad G(z) = \widetilde{G}(z).$$

Such a filter bank is consequently called orthogonal. For orthogonal filter banks we can write the perfect reconstruction condition solely in terms of the synthesis lowpass filter $H(z)$. The relation between the analysis and synthesis filters implies that

$$P(z) = H(z)\widetilde{H}^*(z) = H(z)H(z^{-1}),$$

that is, the sequence p is the autocorrelation of h (see Chapter 2). In the Fourier domain we have

$$P(\omega) = H(\omega)\overline{H(\omega)} = |H(\omega)|^2 \geq 0.$$

This means that $P(\omega)$ is even and real-valued. Orthogonality thus implies that the coefficients in the product filter must be symmetric, that is, $p_n = p_{-n}$. This follows from Section 2.4 on linear phase and symmetry. In the Fourier domain we can now write the perfect reconstruction condition (3.11) as

$$|H(\omega)|^2 + |H(\omega + \pi)|^2 = 2. \tag{3.12}$$

This condition will appear again when we study orthogonal wavelet bases. As we mentioned earlier, the only difference is that for wavelets the left-hand side should equal 1.

Biorthogonal Bases

Recall that an orthogonal filter bank could be seen as a realization of the expansion of a signal into a special type of discrete-time basis. This basis was formed by the even translates of two basis function φ and ψ, where $\varphi_k = h_k$ and $\psi_k = g_k$. And we had

$$x = \sum_n \langle x, \varphi^{(2n)} \rangle \varphi^{(2n)} + \sum_n \langle x, \varphi^{(2n+1)} \rangle \varphi^{(2n+1)},$$

where $\varphi_k^{(2n)} = \varphi_{k-2n}$ and $\varphi_k^{(2n+1)} = \psi_{k-2n}$. Now, a biorthogonal filter bank corresponds to the biorthogonal expansion

$$x = \sum_n \langle x, \widetilde{\varphi}^{(2n)} \rangle \varphi^{(2n)} + \sum_n \langle x, \widetilde{\varphi}^{(2n+1)} \rangle \varphi^{(2n+1)},$$

Here, $\widetilde{\varphi}_k^{(2n)} = \widetilde{\varphi}_{k-2n}$ and $\widetilde{\varphi}_k^{(2n+1)} = \widetilde{\psi}_{k-2n}$; $\widetilde{\varphi}_k = \widetilde{h}_k$ and $\widetilde{\psi}_k = \widetilde{g}_k$.

Exercises 3.4

3.6. Verify that the alias cancellation choice (3.10) of the highpass filters implies that condition (3.9) is satisfied.

3.7. There exists a filter bank that is even simpler than the one based on the Haar basis – the *lazy filter bank*. It is orthogonal and given by the lowpass filter $H(z) = z^{-1}$. What are the corresponding highpass filters? What is the product filter, and does it satisfy the perfect reconstruction condition?

The signals $y^{(0)}$ and $y^{(1)}$ are then the odd- and even-indexed values of x, respectively. Verify this! What are the signals $x^{(0)}$ and $x^{(1)}$? and is their sum equal to x?

3.5 Design of Filter Banks

The discussion in the previous section showed that the construction of a perfect reconstruction filter bank can be reduced to the following three steps:

1. Find a product filter $P(z)$ satisfying $P(z) + P(-z) = 2$.
2. Factor, in some way, the product filter into the two lowpass filters, $P(z) = H(z)\widetilde{H}^*(z)$.
3. Define the highpass filters as

$$G(z) = -z^{-L}\widetilde{H}^*(-z), \quad \text{and} \quad \widetilde{G}(z) = -z^{-L}H^*(-z),$$

   where L is an arbitrary odd integer.

The coefficients in the product filter satisfies $p_0 = 1$ and $p_{2n} = 0$. In an orthogonal filter bank we also have the symmetry condition $p_n = p_{-n}$. The simplest example of such a product filter came from the Haar basis, where

$$P(z) = \frac{1}{2}(z + 2 + z^{-1}).$$

The question is now how to find product filters of higher orders. Such product filters would, in turn, give us low- and highpass filters of higher orders, since $P(z) = H(z)\widetilde{H}^*(z)$. There are usually several different ways to factor $P(z)$ into $H(z)$ and $\widetilde{H}^*(z)$. For an orthogonal filter bank we have $H(z) = \widetilde{H}(z)$, and in the more general case we have a so-called biorthogonal filter bank.

In this section we will describe how to construct one family of product filters discovered by the mathematician Ingrid Daubechies. There are other types of product filters, and we will discuss these later in Chapter 8 (which defines several families of wavelet bases).

The Daubechies Product Filter

In 1988 Daubechies proposed a symmetric product filter of the following form

$$P(z) = \left(\frac{1+z}{2}\right)^N \left(\frac{1+z^{-1}}{2}\right)^N Q_N(z),$$

where $Q_N(z)$ is a symmetric polynomial with $2N-1$ powers in z,

$$Q_N(z) = a_{N-1}z^{N-1} + \cdots + a_1 z + a_0 + a_1 z^{-1} + \ldots a_{N-1}z^{1-N}.$$

The polynomial $Q_N(z)$ is chosen so that $P(z)$ satisfies the perfect reconstruction condition, and it is unique.

So far, no conditions have actually stated that H and $\widetilde{H}$ should be lowpass filters, or that G and $\widetilde{G}$ should be highpass filters. But we see that the Daubechies product filter is chosen so that $P(z)$ has a zero of order $2N$ for $z = -1$, that is, $P(\omega)$ has a zero of order $2N$ for $\omega = \pi$. This means that $P(z)$ is the product of two lowpass filters. As we will see when we present the wavelet theory, the number of zeros are related to the approximation properties of wavelet bases. This is where the theory of wavelets has influenced the design of filter banks. There is nothing in the discrete-time theory that suggest why there should be more than one zero at $z = -1$ for the lowpass filters.

Let us illustrate with two examples.

Example 3.2. For $N = 1$ we obtain the product filter for the Haar basis,

$$P(z) = \left(\frac{1+z}{2}\right)\left(\frac{1+z^{-1}}{2}\right) Q_1(z).$$

Here the condition $p_0 = 1$ implies $Q_1(z) = a_0 = 2$ and we have

$$P(z) = \frac{1}{2}(z + 2 + z^{-1}).$$

□

Example 3.3. For $N = 2$ we get the next higher-order product filter

$$P(z) = \left(\frac{1+z}{2}\right)^2 \left(\frac{1+z^{-1}}{2}\right)^2 Q_2(z).$$

Here $Q_2(z) = a_1 z + a_0 + a_1 z^{-1}$, and if we substitute this expression into $P(z)$ and simplify we get

$$\begin{aligned} P(z) = \frac{1}{16}(a_1 z^3 + (a_0 + 4a_1)z^2 + (4a_0 + 7a_1)z + (6a_0 + 8a_1) \\ + (4a_0 + 7a_1)z^{-1} + (a_0 + 4a_1)z^{-2} + a_1 z^{-3}). \end{aligned}$$

The perfect reconstruction conditions $p_0 = 1$ and $p_2 = 0$ give the linear system

$$\begin{cases} 6a_0 & + & 8a_1 & = & 16, \\ a_0 & + & 4a_1 & = & 0, \end{cases}$$

with solution $a_0 = 4$ and $a_1 = -1$. We then have

$$P(z) = \frac{1}{16}(-z^3 + 9z + 16 + 9z^{-1} - z^{-3}).$$

□

Factorization

Let us first assume that we want to construct an orthogonal filter bank using the symmetric Daubechies product filter. Then, since $P(z) = H(z)H(z^{-1})$, we know that the zeros of $P(z)$ always come in pairs as z_k and z_k^{-1}. When we factor $P(z)$ we can, for each zero z_k, let either $(z - z_k)$ or $(z - z_k^{-1})$ be a factor of $H(z)$. If we always choose the zero that is inside or on the unit circle, $|z_k| \leq 1$, then $H(z)$ is called the minimum phase factor of $P(z)$.

Now, suppose we also want the filter $H(z)$ to be symmetric. Then the zeros of $H(z)$ must come together as z_k and z_k^{-1}. But this contradicts the orthogonality condition except for the Haar basis, where both zeros are at $z = -1$. Thus, *orthogonal filter banks can not have symmetric filters.* [1]

In a biorthogonal basis, or filter bank, we factor the product filter as $P(z) = H(z)\widetilde{H}^*(z)$. There are several ways of doing so, and we then obtain several different filter banks for a given product filter. In most cases, we want the filters $H(z)$ and $\widetilde{H}^*(z)$ to be symmetric, unless we are designing an orthogonal filter bank, that is.

Finally, since we want both $H(z)$ and $\widetilde{H}^*(z)$ to have real coefficients, we always let the complex conjugate zeros z_k and $\overline{z_k}$ belong to either $H(z)$ or $\widetilde{H}^*(z)$.

Let us again illustrate this with an example.

Example 3.4. For $N = 2$ the Daubechies product filter was given by

$$\begin{aligned} P(z) &= \frac{1}{16}(-z^3 + 9z + 16 + 9z^{-1} - z^{-3}) \\ &= \left(\frac{1+z}{2}\right)^2 \left(\frac{1+z^{-1}}{2}\right)^2 (-z + 4 - z^{-1}). \end{aligned}$$

[1] We have assumed that all filters are FIR. A filter bank with IIR symmetric filters can be orthogonal.

This polynomial has four zeros at $z = -1$, one at $z = 2 - \sqrt{3}$, and one at $z = 2 + \sqrt{3}$. Two possible factorizations of this product filter are:

1. Orthogonal and non-symmetric,

$$H(z) = \widetilde{H}(z) = \\ = \frac{1}{4\sqrt{2}} \left((1+\sqrt{3}) + (3+\sqrt{3})z^{-1} + (3-\sqrt{3})z^{-2} + (1-\sqrt{3})z^{-3} \right).$$

2. Biorthogonal and symmetric,

$$H(z) = \frac{1}{2\sqrt{2}} \left(z + 2 + z^{-1} \right), \\ \widetilde{H}(z) = \frac{1}{4\sqrt{2}} \left(-z^2 + 2z + 6 + 2z^{-1} - z^{-2} \right).$$

□

3.6 Notes

There are two books that we direct the reader to, where the wavelet theory is started from the study of filter banks. These are *Wavelets and Filter Banks* by Nguyen and Strang [27], and *Wavelets and Subband Coding* by Kovacevic and Vetterli [30]. These books are appropriate for an engineer or an undergraduate student with a background in signal processing.

Chapter 4

Multiresolution Analysis

This chapter is devoted to the concept of Multi-Resolution Analysis (MRA). As the name suggests, the basic idea is to analyze a function at different resolutions, or scales. Wavelets enter as a way to represent the difference between approximations at different scales.

We begin the chapter with some basic mathematical concepts for continuous-time signal processing. We then turn to the definition of scaling functions, multiresolution analysis and wavelets. Thereafter, we study orthogonal and biorthogonal wavelet bases. Finally, we discuss the approximation properties of scaling functions and wavelets, which provide a way to construct filter banks.

4.1 Projections and Bases in $L^2(\mathbb{R})$

In this section, we present some mathematical concepts that are indispensable for a full understanding of multiresolution analysis and wavelets. We do not always give complete proofs, since this would entail too much mathematical detail. Thus, we state the key definitions and results, and then we discuss them suppressing some formal justifications.

The Hilbert Space $L^2(\mathbb{R})$

For discrete-time signals, we worked within the space $\ell^2(\mathbb{Z})$ of finite-energy signals. In continuous time, the energy of a signal f is given by integrating $|f(t)|^2$ over, for example, the real line. Thus the space of signals in continuous time with finite energy may be given by the following definition.

Definition 4.1. We define $L^2(\mathbb{R})$ as the set of all functions $f(t)$ such that

$$\int_{-\infty}^{\infty} |f(t)|^2 \, dt < \infty.$$

□

This is a linear space equipped with the norm

$$\|f\| = \left(\int_{-\infty}^{\infty} |f(t)|^2 \, dt \right)^{1/2}.$$

Three fundamental properties of the norm are

1. $\|f\| \geq 0$ and $\|f\| = 0 \Rightarrow f = 0$,
2. $\|cf\| = |c| \, \|f\|$, for $c \in \mathbb{C}$,
3. $\|f + g\| \leq \|f\| + \|g\|$. (The triangle inequality)

The L^2-norm can be used as an error measure. If $\widetilde{f}$ is an approximation of f, one way to quantify the error is $\|f - \widetilde{f}\|$. We say that a sequence (f_n) of functions *converge (in L^2)* to the *limit* f, $f_n \to f$ as $n \to \infty$, if $\|f_n - f\| \to 0$ as $n \to \infty$. In other words, the functions f_n become better and better approximations to f (in L^2).

The L^2-norm is induced by the *scalar product*

$$\langle f, g \rangle = \int_{-\infty}^{\infty} f(t) \overline{g(t)} \, dt, \tag{4.1}$$

that is, we have $\|f\| = \sqrt{\langle f, f \rangle}$. That the integral (4.1) is finite is guaranteed by the Cauchy-Schwarz inequality

$$|\langle f, g \rangle| \leq \|f\| \, \|g\|.$$

The existence of a scalar product makes it possible to talk about *orthogonality*; two functions f and g are said to be orthogonal if $\langle f, g \rangle = 0$.

Remark. The space $L^2(\mathbb{R})$ endowed with the scalar product is *complete*: if we have a *Cauchy sequence*, $\|f_n - f_m\| \to 0$ as $m, n \to \infty$, then this sequence *converges in* $L^2(\mathbb{R})$ to a limit f: $\|f_n - f\| \to 0$ as $n \to \infty$. A normed vector space with a scalar product, which is also complete, is termed a *Hilbert space*. Thus, $L^2(\mathbb{R})$ is a Hilbert space, as well as the spaces $\mathbb{R}^n$, $\mathbb{C}^n$, and $\ell^2(\mathbb{Z})$. □

The space $L^2(\mathbb{R})$ contains all physically realizable signals. It is also the natural setting for the continuous-time Fourier transform:

$$\mathcal{F}f(\omega) = \widehat{f}(\omega) = \int_{-\infty}^{\infty} f(t)e^{-i\omega t}\,dt.$$

For the Fourier transform the Parseval formula holds

$$\int_{-\infty}^{\infty} |\widehat{f}(\omega)|^2\,d\omega = 2\pi \int_{-\infty}^{\infty} |f(t)|^2\,dt,$$

or $\|\mathcal{F}f\| = (2\pi)^{1/2}\,\|f\|$. The quantity $|\widehat{f}(\omega)|^2/(2\pi)$ can then be interpreted as the *energy density* at frequency ω. Integrating this energy density over all frequencies gives the total energy of the signal, according to Parseval's formula. We finally remark that Parseval's formula is a special case of the seemingly more general Plancherel's formula

$$\int_{-\infty}^{\infty} \widehat{f}(\omega)\overline{\widehat{g}(\omega)}\,d\omega = 2\pi \int_{-\infty}^{\infty} f(t)\overline{g(t)}\,dt,$$

or $\langle \widehat{f}, \widehat{g}\rangle = 2\pi\langle f, g\rangle$.

Closed Subspaces and Projections

We will consider subspaces of signals in the space $L^2(\mathbb{R})$. A typical example is the class of band-limited signals, *i.e.*, signals f with $\widehat{f}(\omega) = 0$ for $|\omega| > \Omega$, where Ω is some fixed *cut-off frequency*.

A linear subspace V is *closed* if $f_n \in V$, $f_n \to f$ implies $f \in V$. The space of band-limited functions is closed. For closed subspaces it is possible to define a *projection operator* onto that space. The idea behind projections is to approximate an arbitrary signal f with a signal v in the subspace V. We want the optimal approximation, which minimizes $\|f - v\|$, $v \in V$. A unique minimizing w in V can be shown to exist. This justifies the following definition.

Definition 4.2. The (orthogonal) projection of f onto the closed subspace V is the unique $w \in V$ such that

$$\|f - w\| \le \|f - v\|\,, \quad \text{for all } v \in V.$$

The projection of f onto V is denoted $P_V f$. □

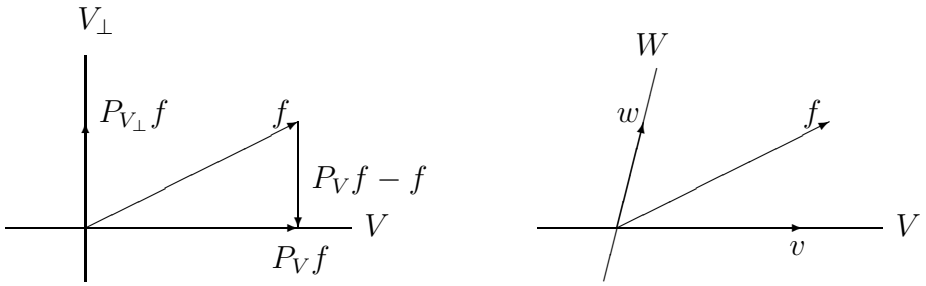

Figure 4.1: Projections in two dimensions.

A useful characterization of the projection $P_V f$ is that $f - P_V f$ is orthogonal to V:

$$\langle f - P_V f, v \rangle = 0, \quad \text{for all } v \in V.$$

This may be illustrated by projections in the plane as in Figure 4.1. If we define the *orthogonal complement* to V as

$$V^\perp = \{u \in L^2(\mathbb{R}) : \langle u, v \rangle = 0, \text{ for all } v \in V\},$$

we have the orthogonal decomposition $f = P_V f + P_{V^\perp} f$.

We will sometimes use the full term *orthogonal projection*, since there are other (oblique) projections. If W is any closed subspace such that every f can be uniquely decomposed as $f = v + w$, where $v \in V$ and $w \in W$, we say that v is the projection of f onto V along W (see Figure 4.1).

Example 4.1. The orthogonal projection onto the space V of band-limited functions, is given by removing everything above the frequency Ω:

$$\widehat{P_V f}(\omega) = \begin{cases} \widehat{f}(\omega) & \text{if } |\omega| < \Omega, \\ 0 & \text{otherwise.} \end{cases} \tag{4.2}$$

We leave it as an exercise to verify this. □

Riesz Bases

The notion of a basis in linear spaces extends from finite dimensions and $\ell^2(\mathbb{Z})$ to $L^2(\mathbb{R})$. We say that a collection $\{\varphi_k\}_{k\in\mathbb{Z}}$ of functions is a *basis* for a linear subspace V if any function $f \in V$ can be written uniquely as

$$f = \sum_k c_k \varphi_k. \tag{4.3}$$

We also say that V is *spanned* by the functions φ_k. The sum (4.3) should be interpreted as the limit of finite sums when the number of terms goes to infinity. More precisely, $\|f - s_K\| \to 0$ as $K \to \infty$, where s_K is the finite sum [1]

$$s_K = \sum_{k=-K}^{K} c_k \varphi_k.$$

In other words, the energy of $f - s_K$ goes to zero when more and more terms in the sum are added. A fundamental fact about the scalar product that we will use throughout the book is:

$$\langle \sum_k c_k \varphi_k, g \rangle = \sum_k c_k \langle \varphi_k, g \rangle. \tag{4.4}$$

The proof is outlined in Exercise 4.4.

The numbers (c_k) are the *coefficients* of f in the basis $\{\varphi_k\}$. The computation of these coefficients from f is called the *analysis* of f in the basis $\{\varphi_k\}$. The reconstruction of f from (c_k) is referred to as *synthesis*. We want analysis and synthesis in the basis $\{\varphi_k\}$ to be numerically stable and therefore make the following definition:

Definition 4.3. A basis $\{\varphi_k\}$ of a subspace V is said to be a *Riesz basis* if, for $f = \sum_k c_k \varphi_k$,

$$A \|f\|^2 \leq \sum_k |c_k|^2 \leq B \|f\|^2,$$

holds, where $0 < A \leq 1 \leq B$ are constants which do not depend on f. □

To motivate why this implies numerical stability, we consider an approximation $\widetilde{f}$ of a signal f. If the signal itself has coefficients (c_k) and its approximation has coefficients $(\widetilde{c}_k)$, we have

$$A\|f - \widetilde{f}\|^2 \leq \sum_k |c_k - \widetilde{c}_k|^2 \leq B\|f - \widetilde{f}\|^2.$$

Small errors in the signal will then give small errors in the coefficients and *vice versa*, provided that A^{-1} and B are of moderate size. A perhaps more relevant result involves relative errors,

$$\frac{\|f - \widetilde{f}\|}{\|f\|} \leq \sqrt{\frac{B}{A}} \frac{\|c - \widetilde{c}\|}{\|c\|} \quad \text{and} \quad \frac{\|c - \widetilde{c}\|}{\|c\|} \leq \sqrt{\frac{B}{A}} \frac{\|f - \widetilde{f}\|}{\|f\|}.$$

[1] Strictly speaking, we should have used two indices K_1 and K_2 that independently go to infinity.

Here, $\|c\|$ is the $\ell^2(\mathbb{Z})$-norm. The number $\sqrt{B/A}$ has a name: the *condition number*. It gives an upper bound on how much relative errors can grow when passing between f and its coefficients (c_k). Since we always have $A \leq B$, the condition number must be at least one. The optimal case occurs when the condition number is 1, $A = B = 1$. We then have an orthonormal (ON) basis. For ON-bases we have $\langle \varphi_k, \varphi_l \rangle = \delta_{k,l}$, where we have used the *Kronecker delta symbol*

$$\delta_{k,l} = \begin{cases} 1 & \text{if} \quad k = l, \\ 0 & \text{otherwise.} \end{cases}$$

Taking scalar products with φ_l in (4.3) gives $c_l = \langle f, \varphi_l \rangle$ and thus every $f \in V$ can be written as

$$f = \sum_k \langle f, \varphi_k \rangle \varphi_k. \tag{4.5}$$

For the orthogonal projection P_V onto V, it is easy to show that, for any $f \in L^2(\mathbb{R})$, $\langle P_V f, \varphi_k \rangle = \langle f, \varphi_k \rangle$. We then have

$$P_V f = \sum_k \langle f, \varphi_k \rangle \varphi_k.$$

There is a generalization of the formula (4.5) for Riesz bases. We then assume a *dual* Riesz basis $\{\widetilde{\varphi}_k\}$ such that $\langle \varphi_k, \widetilde{\varphi}_l \rangle = \delta_{k,l}$. The bases $\{\varphi_k\}$ and $\{\widetilde{\varphi}_k\}$ are said to be *biorthogonal*. In this biorthogonal case,

$$f = \sum_k \langle f, \widetilde{\varphi}_k \rangle \varphi_k.$$

We may define a projection operator (not orthogonal) through

$$P_V f = \sum_k \langle f, \widetilde{\varphi}_k \rangle \varphi_k.$$

If we denote by $\widetilde{V}$ the linear space spanned by the dual basis functions $\widetilde{\varphi}_k$, P_V is the projection onto V along $\widetilde{V}^{\perp}$.

In general, there is no unique choice of the dual basis $\{\widetilde{\varphi}_k\}$. But if we require that the linear space spanned by the dual basis equals V, there is just one such basis. This is, for instance, the case when $V = L^2(\mathbb{R})$.

Exercises 4.1

4.1. Show that the set of band-limited functions is a subspace of $L^2(\mathbb{R})$. Also show that it is closed (quite difficult).

4.2. Verify that the projection operator in Example 4.1 is given by (4.2). Hint: Show that

$$0 = \langle f - P_V f, v\rangle = \frac{1}{2\pi}\langle \widehat{f} - \widehat{P_V f}, \widehat{v}\rangle,$$

for each band-limited v.

4.3. Show that if $\{\varphi_k\}$ is an orthonormal basis for V, and P_V is the orthogonal projection onto V, then holds, for any f,

$$\langle P_V f, \varphi_k\rangle = \langle f, \varphi_k\rangle.$$

4.4. Prove (4.4). Hint: Let f be the sum in (4.3) and let s_K be the finite sum. First show that

$$\langle s_K, g\rangle = \sum_{k=-K}^{K} c_k \langle \varphi_k, g\rangle.$$

which is straightforward. Then follows

$$\begin{aligned}\left|\langle f, g\rangle - \sum_{k=-K}^{K} c_k\langle \varphi_k, g\rangle\right| &= |\langle f, g\rangle - \langle s_K, g\rangle| \\ &= |\langle f - s_K, g\rangle| \\ &\le \|f - s_K\|\,\|g\| \to 0 \text{ as } K \to \infty.\end{aligned}$$

4.2 Scaling Functions and Approximation

The central idea in a multiresolution analysis (MRA) is to approximate functions at different scales, or levels of resolution. These approximations are provided by the *scaling function*, which is sometimes also called the *approximating function*.

The Haar Scaling Function

The Haar scaling function is the simplest example; it is defined by

$$\varphi(t) = \begin{cases} 1 & \text{if } 0 < t < 1, \\ 0 & \text{otherwise.}\end{cases}$$

With this scaling function, we get piecewise constant approximations. For instance, we can approximate a function f with a function f_1 that is piecewise constant on the intervals $(k/2, (k+1)/2)$, $k \in \mathbb{Z}$. This function can be written as

$$f_1(t) = \sum_k s_{1,k}\varphi(2t-k). \tag{4.6}$$

This should be obvious since $\varphi(2t-k)$ equals 1 on $(k/2, (k+1)/2)$ and 0 otherwise (see figure 4.2). The subspace of functions in $L^2(\mathbb{R})$ of the form

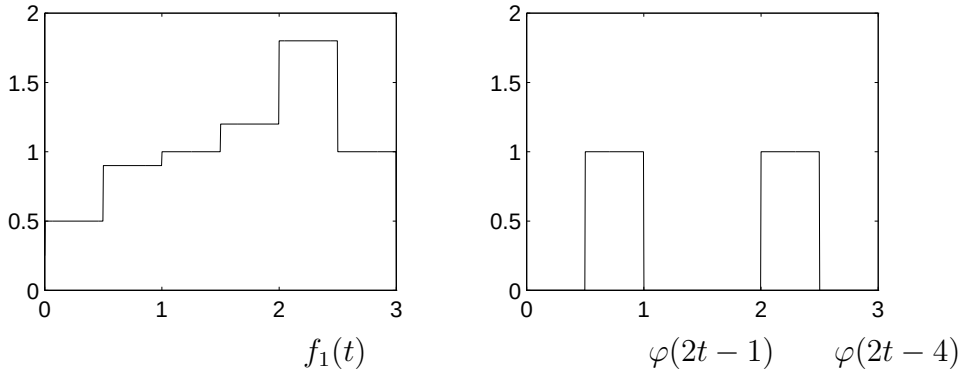

Figure 4.2: A function in V_1 and scaling functions at scale 1/2.

(4.6) is denoted V_1. The coefficients $(s_{1,k})$ may be chosen as the mean values of f over the intervals $(k/2, (k+1)/2)$,

$$s_{1,k} = 2\int_{k/2}^{(k+1)/2} f(t)\,dt = 2\int_{-\infty}^{\infty} f(t)\varphi(2t-k)\,dt.$$

The approximation (4.6) is in fact the orthogonal projection of f onto V_1 having the ON basis $\{2^{1/2}\varphi(2t-k)\}$. For notational convenience, the factor $2^{1/2}$ is included in $s_{1,k}$.

The coefficients $(s_{1,k})$ could also be chosen as the sample values of f at $t = k/2$, $s_{1,k} = f(k/2)$.

We may approximate f on twice a coarser scale by a function f_0, that is piecewise constant on the intervals $(k, k+1)$,

$$f_0(t) = \sum_k s_{0,k}\varphi(t-k). \tag{4.7}$$

If the coefficients $(s_{0,k})$ are choosen as mean values over the intervals

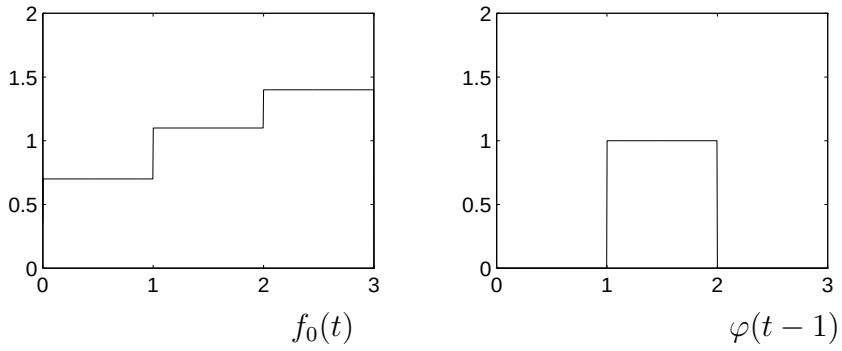

Figure 4.3: A function in V_0 and a scaling function at scale 1.

$(k, k+1)$, it is easy to verify the relation,

$$s_{0,k} = \frac{1}{2}(s_{1,2k} + s_{1,2k+1}).$$

The linear space of the functions in (4.7) is denoted V_0. More generally, we can get piecewise constant approximations on intervals $(2^{-j}k, 2^{-j}(k+1))$, $j \in \mathbb{Z}$, with functions of the form

$$f_j(t) = \sum_k s_{j,k}\varphi(2^j t - k).$$

The corresponding linear space of functions is denoted V_j. The spaces V_j are referred to as *approximation spaces* at the scale 2^{-j}.

The Definition of Multiresolution Analysis

A disadvantage with the Haar scaling function is that it generates discontinuous approximations. Another scaling function is the *hat function*

$$\varphi(t) = \begin{cases} 1+t & \text{if } -1 \le t \le 0, \\ 1-t & \text{if } 0 \le t \le 1, \\ 0 & \text{otherwise.} \end{cases}$$

Taking linear combinations of $\varphi(t-k)$ as in (4.7) gives continuous, piecewise linear approximations.

The approximation spaces V_j in this case consist of continuous functions that are piecewise linear on the intervals $(2^{-j}k, 2^{-j}(k+1))$. This will often give better approximations compared to the Haar scaling function.

We now put forward a general framework for the construction of scaling functions and approximation spaces. This is the notion of a multiresolution analysis.

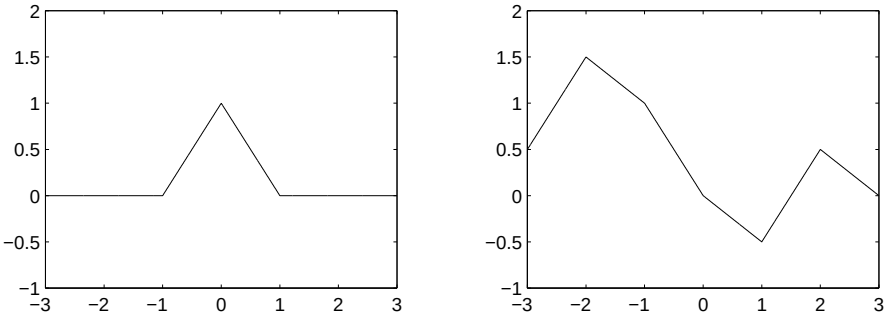

Figure 4.4: The hat scaling function and a function in the corresponding V_0 space.

Definition 4.4. A multiresolution analysis (MRA) is a family of closed subspaces V_j of $L^2(\mathbb{R})$ with the following properties:

1. $V_j \subset V_{j+1}$ for all $j \in \mathbb{Z}$,
2. $f(t) \in V_j \Leftrightarrow f(2t) \in V_{j+1}$ for all $j \in \mathbb{Z}$,
3. $\bigcup_j V_j$ is dense in $L^2(\mathbb{R})$,
4. $\bigcap_j V_j = \{0\}$,
5. There exists a *scaling function* $\varphi \in V_0$, such that $\{\varphi(t-k)\}$ is a Riesz basis for V_0.

□

The first condition just states that functions in V_{j+1} contain more details than functions in V_j: in a certain sense, we add information when we approximate a function at a finer scale. The second condition says that V_{j+1} approximates functions at twice a finer scale than V_j, and also gives a connection between the spaces V_j. The fifth conditions requires the approximation spaces to be spanned by scaling functions. Let us introduce the dilated, translated, and normalized scaling functions

$$\varphi_{j,k}(t) = 2^{j/2}\varphi(2^j t - k).$$

After a scaling by 2^j, it is easy to see that for fixed j, the scaling functions $\varphi_{j,k}$ constitute a Riesz basis for V_j. Thus, every $f_j \in V_j$ can be written as

$$f_j(t) = \sum_k s_{j,k}\varphi_{j,k}(t). \tag{4.8}$$

The reason for the factors $2^{j/2}$ is that all scaling functions have equal norms, $\|\varphi_{j,k}\| = \|\varphi\|$.

The remaining two conditions are of a more technical nature. They are needed to ensure that the wavelets, which we will introduce soon, give a Riesz basis for $L^2(\mathbb{R})$. The third basically says that any function can be approximated arbitrarily well with a function $f_j \in V_j$, if we just choose the scale fine enough. This is what is meant by density. Finally, the fourth condition says, loosely speaking, that the only function that can be approximated at an arbitrarily coarse scale is the zero function.

Properties of the Scaling Function

The definition of MRA imposes quite strong restrictions on the scaling function. In addition, there are some other properties we want the scaling function to have. It should be localized in time, which means a fast decay to zero as $|t| \to \infty$. Preferably, it should have *compact support*, that is, be zero outside a bounded interval. This localization property ensures that the coefficients $(s_{j,k})$ in the approximations (4.8) contain local information about f. Note that the scaling functions we have seen so far, the Haar and hat scaling functions, both have compact support. We further want the scaling function to have integral one,

$$\int_{-\infty}^{\infty} \varphi(t)\, dt = 1. \tag{4.9}$$

This condition has to do with the approximation properties of the scaling function, something we will discuss in more detail in Section 4.7.

Let us now return to the definition of MRA and see what it implies for the scaling function. Since the scaling function belongs to V_0, it also belongs to V_1, according to Condition 1. Thus, it can be written as

$$\varphi(t) = 2\sum_k h_k \varphi(2t - k), \tag{4.10}$$

for some coefficients (h_k). This is the *scaling equation* . Taking the Fourier transform of the scaling equation gives us (Exercise 4.10)

$$\widehat{\varphi}(\omega) = H(\omega/2)\widehat{\varphi}(\omega/2), \tag{4.11}$$

where

$$H(\omega) = \sum_k h_k e^{-ik\omega}.$$

Letting $\omega = 0$ and using $\widehat{\varphi}(0) = 1$, we get $H(0) = \sum h_k = 1$, and we see that the coefficients (h_k) may be interpreted as an averaging filter. In fact, as we will later see, also $H(\pi) = 0$ holds, and thus H is a lowpass filter. It can be shown that the scaling function is uniquely defined by this filter together with the normalization (4.9). As a matter of fact, repeating (4.11) and again using $\widehat{\varphi}(0) = 1$ yields (under certain conditions) the *infinite product formula*

$$\widehat{\varphi}(\omega) = \prod_{j>0} H(\omega/2^j). \tag{4.12}$$

The properties of the scaling function are reflected in the filter coefficients (h_k), and scaling functions are usually constructed by designing suitable filters.

Example 4.2. The B-spline of order N is defined by the convolution

$$S_N(t) = \chi(t) * \ldots * \chi(t), \quad (N \text{ factors})$$

where $\chi(t)$ denotes the Haar scaling function. The B-spline is a scaling function (Exercise 4.7) for each N. The translated functions $S_N(t-k)$ gives $N-2$ times continuously differentiable, piecewise polynomial approximations of degree $N-1$. The cases $N = 1$ and $N = 2$ corresponds to the Haar and the hat scaling functions, respectively. When N grows larger, the scaling functions become more and more regular, but also more and more spread out. We will describe wavelets based on B-spline scaling functions in Chapter 8. □

Example 4.3. The sinc function

$$\varphi(t) = \operatorname{sinc} t = \frac{\sin \pi t}{\pi t},$$

is another scaling function (Exercise 4.8). It does not have compact support, and the decay as $|t| \to \infty$ is very slow. Therefore, is it not used in practice. It has interesting theoretical properties though.

It is in a sense dual to the Haar scaling function, since its Fourier transform is given by the box function

$$\widehat{\varphi}(\omega) = \begin{cases} 1 & \text{if } -\pi < \omega < \pi, \\ 0 & \text{otherwise.} \end{cases}$$

It follows that every function in V_0 is band-limited with cut-off frequency π. In fact, the Sampling Theorem (*cf.* Section 2.7) states that every such band-limited function f can be reconstructed from its sample values $f(k)$ *via*

$$f(t) = \sum_k f(k) \operatorname{sinc}(t-k).$$

Thus, V_0 equals the set of band-limited functions with cut-off frequency π. The V_j spaces, by scaling, become the set of functions band-limited to the frequency bands $(-2^j\pi, 2^j\pi)$. □

Exercises 4.2

4.5. Show that $f(t) \in V_j \Leftrightarrow f(2^{-j}t) \in V_0$.

4.6. Verify that $\{\varphi_{j,k}\}_{k\in\mathbb{Z}}$ is a Riesz basis for V_j for fixed j, given that $\{\varphi_{0,k}\}_{k\in\mathbb{Z}}$ is a Riesz basis for V_0.

4.7. Derive the scaling equation for the spline scaling functions. Hint: Work in the Fourier domain and show that $\varphi(\omega) = H(\omega/2)\varphi(\omega/2)$, where $H(\omega)$ is 2π-periodic. Calculate the coefficients (h_k).

4.8. Derive the scaling equation for the sinc scaling function. Hint: Work as in the previous problem.

4.9. Show that the scaling equation can be written more generally as

$$\varphi_{j,k} = \sqrt{2}\sum_l h_l \varphi_{j+1,l+2k}.$$

4.10. Prove identity (4.11).

4.11. Verify (4.12). Why do we need $\widehat{\varphi}(0) = 1$?

4.3 Wavelets and Detail Spaces

We will now turn to a description of the difference between two successive approximation spaces in a multiresolution analysis: the wavelet or detail spaces.

The Haar Wavelet

A multiresolution analysis allows us to approximate functions at different levels of resolution. Let us look at the approximations of a function f at two consecutive scales, $f_0 \in V_0$ and $f_1 \in V_1$. The approximation f_1 contains more details than f_0 and the difference is the function

$$d_0 = f_1 - f_0.$$

We return again to the Haar system. Here, f_1 is piecewise constant on the intervals $(k/2,(k+1)/2)$ with values $s_{1,k}$, and f_0 is piecewise constant on $(k,k+1)$ with values $s_{0,k}$ that are pairwise mean values of the $s_{1,k}$:s

$$s_{0,k} = \frac{1}{2}(s_{1,2k} + s_{1,2k+1}). \tag{4.13}$$

In Figure 4.5, we have plotted the function d_0 in the Haar system. It

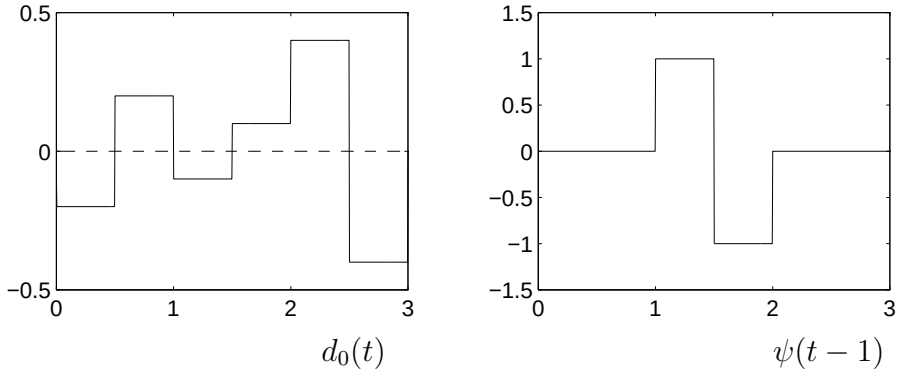

Figure 4.5: The Haar wavelet and a function in the corresponding W_0 space

is piecewise constant on the intervals $(k/2,(k+1)/2)$ so $d_0 \in V_1$. Let us consider the interval $(k,k+1)$ in Figure 4.6.

Denote d_0:s value on the first half by $w_{0,k}$. The value on the second half is then $-w_{0,k}$ and they both measure the deviation of f_1 from its mean value on the interval $(k,k+1)$:

$$w_{0,k} = \frac{1}{2}(s_{1,2k} - s_{1,2k+1}). \tag{4.14}$$

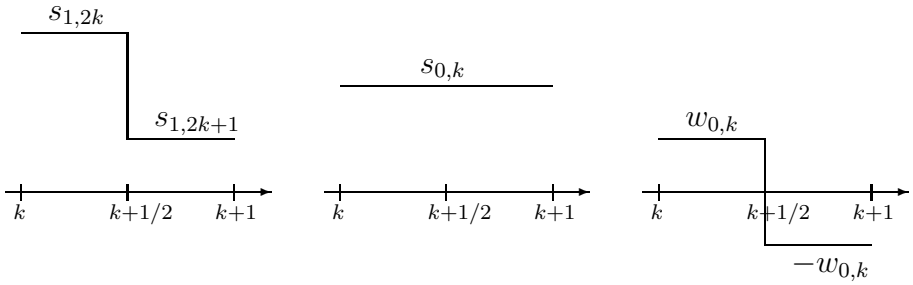

Figure 4.6: Two scales and their difference

The *Haar wavelet* is defined by

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

Using this wavelet we can write d_0 as

$$d_0(t) = \sum_k w_{0,k}\psi(t-k), \tag{4.15}$$

with *wavelet coefficients* $(w_{0,k})$ at scale $1 = 2^0$. We want to generalize this to any MRA, and therefore make the following definition.

Definition 4.5. For a general MRA, a function ψ is said to be a *wavelet* if the *detail space* W_0 spanned by the functions $\psi(t-k)$ complements V_0 in V_1. By this we mean that any $f_1 \in V_1$ can be uniquely written as $f_1 = f_0 + d_0$, where $f_0 \in V_0$ and $d_0 \in W_0$. We write this formally as $V_1 = V_0 \oplus W_0$. Finally, we require the wavelets $\psi(t-k)$ to be a Riesz basis for W_0. □

Note that the space W_0 need not be unique. However, the decomposition $f_1 = f_0 + d_0$ is unique, once the wavelet ψ (the space W_0) is chosen. However, if we require W_0 to be orthogonal to V_0, then W_0 is uniquely determined.

Example 4.4. When the scaling function is the hat function the wavelet can be chosen as the function in V_1 with values at the half-integers

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

This wavelet is shown in Figure 4.7. In this case, V_0 and W_0 are not orthogonal. □

Example 4.5. For the sinc scaling function from Example 4.3, we choose the wavelet as

$$\widehat{\psi}(\omega) = \begin{cases} 1 & \text{if } \pi < |\omega| < 2\pi, \\ 0 & \text{otherwise.} \end{cases}$$

It is easy to see that $\psi(t) = \operatorname{sinc} 2t - \operatorname{sinc} t$ (Exercise 4.14). This is the *sinc wavelet*. The space W_0 will be the set of functions that are band-limited to the frequency band $\pi < |\omega| < 2\pi$. More generally, the space W_j will contain all functions that are band-limited to the frequency band $2^j\pi < |\omega| < 2^{j+1}\pi$. □

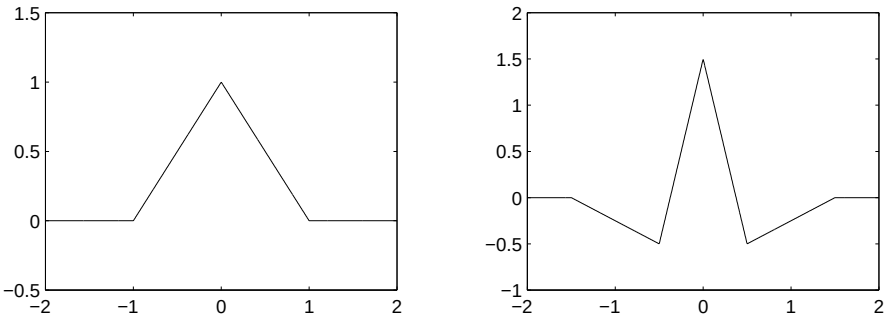

Figure 4.7: The hat scaling function and a piecewise linear wavelet spanning W_0.

Properties of the Wavelet

The function ψ is sometimes called the *mother wavelet*. As with the scaling function, we want it to be localized in time. We also want it to have integral zero

$$\int \psi(t)\, dt = 0,$$

or $\widehat{\psi}(0) = 0$, and it thus has to oscillate. This is also referred to as the cancellation property and it is connected to having wavelets represent differences. The function ψ will be a wave that quickly emerges and dies off, hence the term wavelet (small wave). Since $\psi \in V_1$ it can be written as

$$\psi(t) = 2\sum_k g_k \varphi(2t - k), \tag{4.16}$$

for some coefficients (g_k). This is the *wavelet equation* . A Fourier transform gives

$$\widehat{\psi}(\omega) = G(\omega/2)\widehat{\varphi}(\omega/2), \tag{4.17}$$

where

$$G(\omega) = \sum g_k e^{-ik\omega}.$$

Using $\widehat{\varphi}(0) = 1$ and $\widehat{\psi}(0) = 0$, we get that $G(0) = \sum g_k = 0$. Thus the coefficients (g_k) can be interpreted as a difference filter. Later we will see that also $G(\pi) = 1$ holds, and G is in fact a highpass filter. The wavelet and all its properties are determined by this filter, given the scaling function.

The Wavelet Decomposition

The dilated and translated wavelets $\psi_{j,k}$ are defined as

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

The detail spaces W_j are defined as the the set of functions of the form

$$d_j(t) = \sum_k w_{j,k}\psi_{j,k}(t). \tag{4.18}$$

From Definition 4.5 it follows that any $f_1 \in V_1$ can be decomposed as $f_1 = f_0 + d_0$, where $f_0 \in V_0$ is an approximation at twice a coarser scale, and $d_0 \in W_0$ contains the lost details. After a scaling with 2^j, we see that a function $f_{j+1} \in V_{j+1}$ can be decomposed as $f_{j+1} = f_j + d_j$, where $f_j \in V_j$ and $d_j \in W_j$, that is, $V_{j+1} = V_j \oplus W_j$.

Starting at a finest scale J, and repeating the decomposition $f_{j+1} = f_j + d_j$ until a certain level j_0, we can write any $f_J \in V_J$ as

$$\begin{aligned} f_J(t) &= d_{J-1}(t) + d_{J-2}(t) + \ldots d_{j_0}(t) + f_{j_0}(t) \\ &= \sum_{j=j_0}^{J-1}\sum_k w_{j,k}\psi_{j,k}(t) + \sum_k s_{j_0,k}\varphi_{j_0,k}(t). \end{aligned}$$

We can express this in terms of approximation and detail spaces as

$$V_J = W_{J-1} \oplus W_{J-2} \oplus \ldots \oplus W_{j_0} \oplus V_{j_0}.$$

Using the fourth condition in the definition of MRA one can show that f_{j_0} goes to 0 in L^2 when $j_0 \to -\infty$. The third condition now implies that, choosing J larger and larger, we can approximate a function f with approximations f_J that become closer and closer to f. Letting $J \to \infty$ therefore gives us the *wavelet decomposition* of f

$$f(t) = \sum_{j,k} w_{j,k}\psi_{j,k}(t). \tag{4.19}$$

We have thus indicated how to prove that $\{\psi_{j,k}\}$ is a basis for $L^2(\mathbb{R})$. However, it still remains to construct the highpass filter G determining the mother wavelet ψ.

The decomposition $V_{j+1} = V_j \oplus W_j$ above is not unique. There are many ways to choose the wavelet ψ and the corresponding detail spaces W_j. Each such choice corresponds to a choice of the highpass filter G. In the next section, we will describe a special choice, which gives us an *orthogonal system*

or orthogonal wavelet basis. This corresponds to choosing H and G as filters in an orthogonal filter bank. Thereafter we discuss the more general case, *biorthogonal systems*, which corresponds to biorthogonal filter banks.

We conclude this section by looking at the wavelet decomposition in the frequency domain. We saw earlier, that for the sinc scaling function, the V_j spaces are the spaces of functions band-limited to the frequency bands $(0, 2^j\pi)$ (actually $(-2^j\pi, 2^j\pi)$, but we ignore negative frequencies to simplify the discussion). The detail spaces are the sets of band-limited functions in the frequency bands $(2^j\pi, 2^{j+1}\pi)$. The wavelet decomposition can in this case be seen as a decomposition of the frequency domain as in Figure 4.8. For

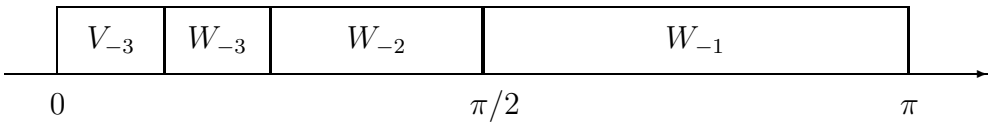

Figure 4.8: The wavelet decomposition in the frequency domain.

other wavelets, this frequency decomposition should be interpreted approximately, since the wavelets $\psi_{j,k}$ have frequency contents outside the band $(2^j\pi, 2^{j+1}\pi)$. In other words, the frequency bands are overlapping. The amount of overlapping will depend on the mother wavelet.

Exercises 4.3

4.12. Verify that the wavelet in Example 4.4 will span the difference between V_1 and V_0.

4.13. Verify that, for $j \neq 0$, each function $f_{j+1} \in V_{j+1}$ can be written as $f_{j+1} = f_j + d_j$ where $d_j \in W_j$. (For $j = 0$ the statement is true by definition.)

4.14. Verify that $\psi(t) = \operatorname{sinc} 2t - \operatorname{sinc} t$, when ψ is the sinc wavelet in Example 4.3. Hint: Work in the Fourier domain.

4.4 Orthogonal Systems

In this section, the wavelet space W_0 is to be orthogonal to the approximation space V_0. This means that we will get an orthogonal decomposition of V_1 into $V_1 = V_0 \oplus W_0$, and ultimately we will arrive at an orthonormal wavelet basis in $L^2(\mathbb{R})$.

Orthogonality Conditions

The first requirement is that the scaling functions $\varphi(t-k)$ constitute an orthogonal basis for V_0, that is,

$$\int_{-\infty}^{\infty} \varphi(t-k)\varphi(t-l)\,dt = \delta_{k,l}.$$

Using the scaling equation (4.10) we can transform this to a condition on the coefficients (h_k) (see Exercise 4.15):

$$\sum_l h_l h_{l+2k} = \delta_k/2. \tag{4.20}$$

We also require the wavelets $\psi(t-k)$ to form an orthogonal basis for W_0,

$$\int_{-\infty}^{\infty} \psi(t-k)\psi(t-l)\,dt = \delta_{k,l}.$$

Expressed in the coefficients (g_k) this becomes

$$\sum_l g_l g_{l+2k} = \delta_k/2. \tag{4.21}$$

Finally, functions in V_0 must be orthogonal to functions in W_0 which is formally written as $V_0 \perp W_0$. (Figure 4.9). Then the scaling functions

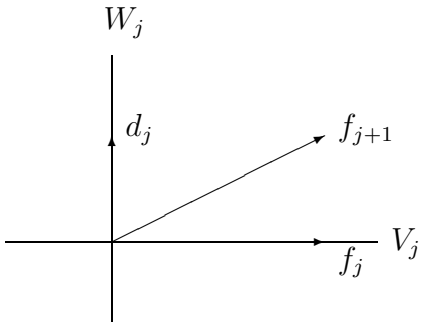

Figure 4.9: $V_j \perp W_j$

$\varphi(t-k)$ have to be orthogonal to each wavelet $\psi(t-l)$:

$$\int_{-\infty}^{\infty} \varphi(t-k)\psi(t-l)\,dt = 0, \quad \text{for all } k \text{ and } l.$$

For the filter coefficients this means that (see Exercise 4.17)

$$\sum_m h_{m+2k} g_{m+2l} = 0. \tag{4.22}$$

All of the above orthogonality conditions can be transposed to an arbitrary scale. Using the scalar product notation we have

$$\begin{aligned} \langle \varphi_{j,k}, \varphi_{j,l} \rangle &= \delta_{k,l}, \\ \langle \psi_{j,k}, \psi_{j,l} \rangle &= \delta_{k,l}, \\ \langle \varphi_{j,k}, \psi_{j,l} \rangle &= 0. \end{aligned}$$

In other words, $\{\varphi_{j,k}\}$ is an orthonormal basis for V_j, $\{\psi_{j,k}\}$ for W_j, and $V_j \perp W_j$. The approximation $f_j \in V_j$ of a function f can be chosen as the orthogonal projection onto V_j, which we denote by P_j. It can be computed as

$$f_j = P_j f = \sum_k \langle f, \varphi_{j,k} \rangle \varphi_{j,k}.$$

The detail d_j becomes the projection of f onto W_j,

$$d_j = Q_j f = \sum_k \langle f, \psi_{j,k} \rangle \psi_{j,k}.$$

In terms of the filter functions $H(\omega)$ and $G(\omega)$, (4.20) - (4.22) becomes

$$\begin{aligned} |H(\omega)|^2 + |H(\omega+\pi)|^2 &= 1, \\ |G(\omega)|^2 + |G(\omega+\pi)|^2 &= 1, \\ H(\omega)\overline{G(\omega)} + H(\omega+\pi)\overline{G(\omega+\pi)} &= 0. \end{aligned} \tag{4.23}$$

As we will see in the next section, these are the conditions on $\sqrt{2}\,H$ and $\sqrt{2}\,G$ to be low- and highpass filters in an orthogonal filter bank[2]. This is a crucial observation, since construction of orthogonal scaling functions and wavelets becomes equivalent with the construction of orthogonal filter banks.

Example 4.6. We now give a concrete example on how to construct orthogonal wavelets from an orthogonal filter bank. We start with the Daubechies product filter with $N = 5$ (see Section 3.5). We then use the minimum phase orthogonal factorization, and this gives us the so called Daubechies-10 orthogonal filter bank. The corresponding scaling function and wavelet can

[2] Note that the factor $\sqrt{2}$ is included in the filters in Chapter 3. Then the right-hand sides of 4.23 are replaced by 2

now be computed numerically, using the *cascade algorithm*, to be explained in Chapter 8. We have plotted the Daubechies-10 scaling function and mother wavelet in Figure 4.10. Note that they are smoother than the hat scaling function and the corresponding wavelet (and, of course, smoother than in the Haar case). The price is the increased support width. In Chapter 8, we will describe most wavelet bases that have been used in applications. □

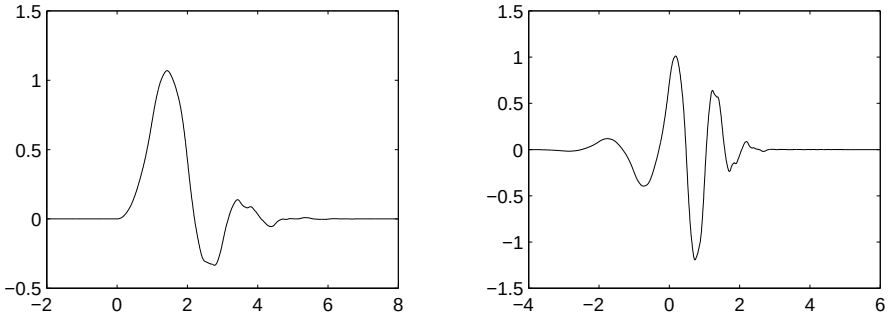

Figure 4.10: The Daubechies-10 scaling function and wavelet

Characterizations in the Orthonormal Case

As we now will see, orthogonal filter bank does not always lead to an orthonormal wavelet basis.

First of all, the infinite product (4.12) has to converge. If it does converge, it need not produce a function in L^2. Consider, for example, the most simple low-pass filter $h_k = \delta_k$, the identity filter. Together with the high-pass filter $g_k = \delta_{k-1}$, this gives an orthogonal filter bank. It is called the lazy filter bank, since it does not do anything except splitting the input signal into even- and odd-indexed samples. The filter function is $H(\omega) = 1$, and the infinite product formula implies that $\widehat{\varphi}(\omega) = 1$. The scaling function is then the Dirac delta distribution, which is not even a function. Even if the product yields a function in L^2, we have to make sure that also conditions 3 and 4 in the definition of MRA (Definition 4.4) are satisfied. An additional simple sufficient condition on the low-pass filter $H(\omega)$, determining the scaling function, to satisfy 3 and 4 is $H(\omega) > 0$ for $|\omega| \leq \pi/2$ (and $\hat{\varphi}(0) = 1$).

Now we are about to give precise charactarizations on scaling functions and wavelets in an orthogonal MRA. All equations are to hold almost everywhere (a.e.). We advice readers not familiar with this concept, to interpret almost everywhere as "everywhere except a finite number of points".

Theorem 4.1. The following three conditions (a.e.) on a function φ with $\|\varphi\| = 1$ are equivalent to φ being a scaling function for an orthogonal multiresolution analysis of $L^2(\mathbb{R})$.

1. $\sum_k |\widehat{\varphi}(\omega + 2\pi k)|^2 = 1,$
2. $\lim_{j\to\infty} \widehat{\varphi}(2^{-j}\omega) = 1,$
3. $\widehat{\varphi}(2\omega) = H(\omega)\widehat{\varphi}(\omega),$ for some 2π- periodic function H.

□

The first condition is equivalent to the orthonormality of the integer translates $\varphi(t-k)$. The second is essentially the approximation property of the MRA, and the third is the scaling equation.

Theorem 4.2. The following three conditions (a.e.) on a function ψ with $\|\psi\| = 1$ are equivalent to having an orthonormal basis $\psi_{j,k}$ connected to a multiresolution analysis of $L^2(\mathbb{R})$.

1. $\sum_j \left|\widehat{\psi}(2^j\omega)\right|^2 = 1,$
2. $\sum_{j\geq 0} \widehat{\psi}(2^j\omega)\overline{\widehat{\psi}(2^j(\omega + 2k\pi))} = 0, \quad$ for all odd integers k,
3. $\sum_{j\geq 1}\sum_k \left|\widehat{\psi}(2^j(\omega + 2k\pi))\right|^2 = 1.$

□

The first two conditions are equivalent to the orthonormal basis property of $\psi_{j,k}$ in $L^2(\mathbb{R})$, and the third then relates to a corresponding scaling function. Broadly speaking, the first condition means that linear combinations of $\psi_{j,k}$ are dense in $L^2(\mathbb{R})$, and the second then relates to the orthogonality.

Exercises 4.4

4.15. Verify that the condition on the scaling functions $\varphi(t-k)$ to be orthogonal implies (4.20). Hint: First, you may assume $l = 0$ (why?). Then,

show that

$$\begin{aligned}
\delta_k &= \int_{-\infty}^{\infty} \varphi(t)\varphi(t-k)\,dt \\
&= \int_{-\infty}^{\infty} \left(2\sum_l h_l \varphi(2t-l)\right)\left(2\sum_m h_m \varphi(2t-2k-m)\right)\,dt \\
&= 4\sum_{l,m} \int_{-\infty}^{\infty} h_l h_m \varphi(2t-l)\varphi(2t-2k-m)\,dt \\
&= 2\sum_{l,m} \int_{-\infty}^{\infty} h_l h_m \varphi(t-l)\varphi(t-2k-m)\,dt \\
&= 2\sum_l h_l h_{l+2k}.
\end{aligned}$$

4.16. Show that (4.20) is equivalent to the first equation in (4.23).

4.17. Show that the condition on the scaling functions $\varphi(t-k)$ to be orthogonal to the wavelets $\psi(t-k)$ implies (4.22).

4.5 The Discrete Wavelet Transform

We are now about to describe how to compute scaling and wavelet coefficients $(s_{j_0,k})$ and $(w_{j,k})$ in the wavelet decomposition

$$f_J(t) = \sum_{j=j_0}^{J-1} \sum_k w_{j,k}\psi_{j,k}(t) + \sum_k s_{j_0,k}\varphi_{j_0,k}(t).$$

The computation of the coefficients is done with filter banks. For the sake of simplicity we derive this connection between MRA's and filter banks in the orthogonal case. The biorthogonal case is entirely similar, the notation just becomes a bit more cumbersome.

We assume that we know the scaling coefficients $s_{J,k} = \langle f, \varphi_{J,k}\rangle$ of a function f at a certain finest scale J. In practice, we usually only have sample values $f(2^{-J}k)$ available, and we have to compute the scaling coefficients numerically from these sample values. This is known as *pre-filtering*. It is common practice to simply replace the scaling coefficients with the sample values. The effect of doing so, and other aspects of pre-filtering, will be treated in Chapter 15.

The Forward Wavelet Transform

Assume that we know the scaling coefficients $s_{j+1,k} = \langle f, \varphi_{j+1,k} \rangle$ at an arbitrary scale $j+1$. We split f_{j+1} into a coarser approximation f_j and details d_j:

$$f_{j+1}(t) = f_j(t) + d_j(t),$$

or written out more explicitly

$$\sum_k s_{j+1,k} \varphi_{j+1,k}(t) = \sum_k s_{j,k} \varphi_{j,k}(t) + \sum_k w_{j,k} \psi_{j,k}(t), \tag{4.24}$$

where $s_{j,k} = \langle f, \varphi_{j,k} \rangle$ and $w_{j,k} = \langle f, \psi_{j,k} \rangle$. A scalar multiplication on both sides with $\varphi_{j,l}$ together with the orthogonality conditions for scaling functions and wavelets gives us (Exercise 4.18)

$$s_{j,k} = \sum_l s_{j+1,l} \langle \varphi_{j+1,l}, \varphi_{j,k} \rangle$$

From the scaling equation

$$\varphi_{j,k} = \sqrt{2} \sum_m h_m \varphi_{j+1,m+2k}$$

we obtain

$$\langle \varphi_{j+1,l}, \varphi_{j,k} \rangle = \sqrt{2} \sum_m h_m \langle \varphi_{j+1,l}, \varphi_{j+1,m+2k} \rangle = \sqrt{2}\, h_{l-2k}.$$

With a similar calculation for the wavelet coefficients, we have derived the formulas

$$s_{j,k} = \sqrt{2} \sum_l h_{l-2k} s_{j+1,l} \quad \text{and} \quad w_{j,k} = \sqrt{2} \sum_l g_{l-2k} s_{j+1,l}. \tag{4.25}$$

Scaling and wavelet coefficients at the coarser scale are thus computed by sending the scaling coefficients at the finer scale through the analysis part of the orthogonal filter bank with lowpass and highpass filter $\sqrt{2}\,H$ and $\sqrt{2}\,G$.

Repeating this recursively, starting with the coefficients $(s_{J,k})$, gives the wavelet coefficients $(w_{j,k})$ for $j = j_0, \ldots, J-1$, and the scaling coefficients $(s_{j_0,k})$ at the coarsest scale. This recursive scheme is called the *Fast Forward Wavelet Transform*. If we start with N scaling coefficients at the finest scale, the computational effort is roughly $4MN$ operations, where M is the filter length. Compare this with the FFT algorithm, where the computational effort is $2N \log N$ operations.

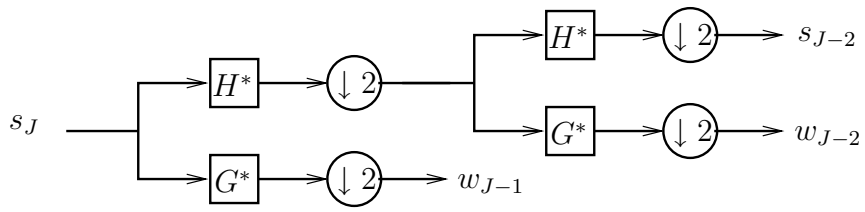

Figure 4.11: The Forward Wavelet Transform as an iterated filter bank

The forward wavelet transform amounts to computing

$$f(t) \approx f_J(t) = \sum_{j=j_0}^{J-1} \sum_k \langle f, \psi_{j,k} \rangle \psi_{j,k}(t) + \sum_k \langle f, \varphi_{j_0,k} \rangle \varphi_{j_0,k}(t).$$

Letting $j_0 \to -\infty$ and $J \to \infty$ gives us

$$f(t) = \sum_{j,k} \langle f, \psi_{j,k} \rangle \psi_{j,k}(t),$$

an expansion of f in the ON-basis $\{\psi_{j,k}\}$.

The Inverse Wavelet Transform

There is, of course, an inverse wavelet transform as well. Since we compute $s_j = (s_{j,k})_{k \in \mathbb{Z}}$ and $w_j = (w_{j,k})_{k \in \mathbb{Z}}$ from s_{j+1} via the analysis part of a filter bank, s_{j+1} can be reconstructed by feeding s_j and w_j into the synthesis part,

$$s_{j+1,k} = \sqrt{2} \sum_l (h_{k-2l} s_{j,k} + g_{k-2l} w_{j,k}).$$

From s_{j_0} and w_{j_0}, we can thus reconstruct s_{j_0+1}, which together with w_{j_0+1} gives us s_{j_0+2} and so on, until we finally arrive at the s_J. This is the *Fast Inverse Wavelet Transform*, see Figure 4.12. Note that we use neither the scaling function nor the wavelet explicitly in the forward or inverse wavelet transform, only the orthogonal filter bank.

To recover the sample values $f(2^{-J}k)$ from the fine-scale coefficients $(s_{J,k})$ we need to do a *post-filtering* step. This will be be discussed in Chapter 15.

Exercises 4.5

4.18. Verify all the steps in the derivation of the filter equations (4.25).

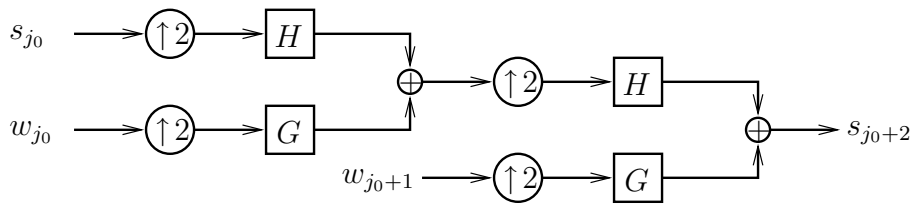

Figure 4.12: The Inverse Wavelet Transform as an iterated filter bank

4.19. Assume that $\{\psi_{j,k}\}$ is an orthonormal basis in $L^2(R)$. Put $\Psi(t) = 2^{1/2}\psi(2t)$, and thus $\Psi_{j,k} = \psi_{j+1,2k}$. Show that $\{\Psi_{j,k}\}$ becomes an orthonormal system, which is not a (Riesz) basis. Hint: It misses all $\psi_{j,2l+1}$, at least. That is, the functions $\psi_{j,2l+1}$ all have coefficients 0. Why does this contradict $\{\Psi_{j,k}\}$ being a Riesz basis?

4.6 Biorthogonal Systems

Biorthogonality Conditions

In a biorthogonal system we also have a *dual* MRA with scaling functions $\widetilde{\varphi}_{j,k}$ and wavelets $\widetilde{\psi}_{j,k}$, and corresponding approximation and detail spaces $\widetilde{V}_j$ and $\widetilde{W}_j$. The dual scaling function $\widetilde{\varphi}$ satisfies a scaling equation,

$$\widetilde{\varphi}(t) = 2\sum \widetilde{h}_k \widetilde{\varphi}(2t-k). \tag{4.26}$$

For the dual mother wavelet holds the dual wavelet equation

$$\widetilde{\psi}(t) = 2\sum \widetilde{g}_k \widetilde{\varphi}(2t-k). \tag{4.27}$$

We have the following *biorthogonality conditions* on the scaling functions and wavelets:

$$\begin{aligned}
\langle \varphi_{j,k}, \widetilde{\varphi}_{j,l} \rangle &= \delta_{k,l}, \\
\langle \psi_{j,k}, \widetilde{\psi}_{j,l} \rangle &= \delta_{k,l}, \\
\langle \varphi_{j,k}, \widetilde{\psi}_{j,l} \rangle &= 0, \\
\langle \widetilde{\varphi}_{j,k}, \psi_{j,l} \rangle &= 0.
\end{aligned}$$

The two latter conditions mean that $V_j \perp \widetilde{W}_j$ and $\widetilde{V}_j \perp W_j$ respectively (see Figure 4.13). After some calculations similar to those in the orthogonal case

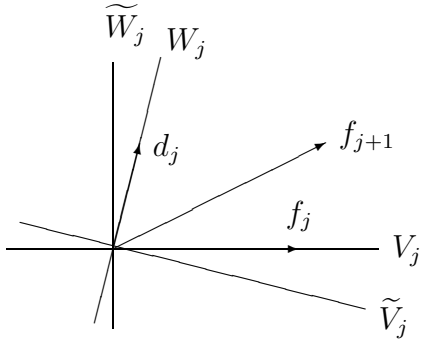

Figure 4.13: $V_j \perp \widetilde{W}_j$ and $\widetilde{V}_j \perp W_j$

we arrive at the following biorthogonality conditions for the filter functions $H(\omega), \widetilde{H}(\omega), G(\omega)$ and $\widetilde{G}(\omega)$:

$$
\begin{array}{rcl}
\widetilde{H}(\omega)\overline{H(\omega)} + \widetilde{H}(\omega+\pi)\overline{H(\omega+\pi)} &=& 1, \\
\widetilde{G}(\omega)\overline{G(\omega)} + \widetilde{G}(\omega+\pi)\overline{G(\omega+\pi)} &=& 1, \\
\widetilde{G}(\omega)\overline{H(\omega)} + \widetilde{G}(\omega+\pi)\overline{H(\omega+\pi)} &=& 0, \\
\widetilde{H}(\omega)\overline{G(\omega)} + \widetilde{H}(\omega+\pi)\overline{G(\omega+\pi)} &=& 0.
\end{array}
\tag{4.28}
$$

If we introduce the *modulation matrix*

$$
M(\omega) = \begin{bmatrix} H(\omega) & H(\omega+\pi) \\ G(\omega) & G(\omega+\pi) \end{bmatrix},
$$

and similarly for $\widetilde{M}(\omega)$, we can write Equation (4.28) in the more compact form

$$
\overline{M(\omega)}\widetilde{M}(\omega)^{\mathrm{T}} = I.
$$

We also must have $\widetilde{M}(\omega)^{\mathrm{T}}\overline{M(\omega)} = I$, and written out this becomes

$$
\begin{array}{rcl}
\widetilde{H}(\omega)\overline{H(\omega)} + \widetilde{G}(\omega)\overline{G(\omega)} &=& 1, \\
\widetilde{H}(\omega)\overline{H(\omega+\pi)} + \widetilde{G}(\omega)\overline{G(\omega+\pi)} &=& 0.
\end{array}
\tag{4.29}
$$

Hence, the four equations in (4.28) can actually be reduced to two. The latter equations are the perfect reconstruction conditions (3.8)-(3.9) for filter banks, transformed to the Fourier domain (Exercise 4.21). It means that $\sqrt{2}\,H$, $\sqrt{2}\,\widetilde{H}$, $\sqrt{2}\,G$, and $\sqrt{2}\,\widetilde{G}$ are high- and lowpass filters in a biorthogonal filter bank.

We will now derive a connection between the low- and highpass filters. Cramer's rule gives us (see Exercise 4.22)

$$\widetilde{H}(\omega) = \frac{\overline{G(\omega+\pi)}}{\overline{\Delta(\omega)}} \quad \text{and} \quad \widetilde{G}(\omega) = \frac{\overline{H(\omega+\pi)}}{\overline{\Delta(\omega)}}, \tag{4.30}$$

where $\Delta(\omega) = \det M(\omega)$. In practice we often want finite filters, which corresponds to wavelets and scaling functions having compact support. Then one can show that $\Delta(\omega)$ has the form $\Delta(\omega) = Ce^{-Li\omega}$ for some odd integer L and constant C with $|C| = 1$. Different choices give essentially the same wavelet: the only thing that differs is an integer translation and the constant C. A common choice is $C = 1$ and $L = 1$ which gives the *alternating flip* construction (*cf.* Equation 3.10):

$$G(\omega) = -e^{-i\omega}\overline{\widetilde{H}(\omega+\pi)} \quad \text{and} \quad \widetilde{G}(\omega) = -e^{-i\omega}\overline{H(\omega+\pi)}, \tag{4.31}$$

or, expressed in the filter coefficients,

$$g_k = (-1)^k \widetilde{h}_{1-k} \quad \text{and} \quad \widetilde{g}_k = (-1)^k h_{1-k}.$$

This leaves us with the task of designing proper lowpass filters that satisfy the first equation in (4.28). It is then easy to verify (Exercise 4.23) that the remaining equations are satisfied.

We mentioned above that finite filters correspond to wavelets and scaling functions having compact support. This is based on the *Paley-Wiener theorem*. Taking this as a fact, and assuming that the lowpass filters have lengths M and $\widetilde{M}$, it is straightforward to show that φ and $\widetilde{\varphi}$ are zero outside intervals of length $M-1$ and $\widetilde{M}-1$ (Exercise 4.24). Further, both ψ and $\widetilde{\psi}$ are supported on intervals of length $(M+\widetilde{M}-2)/2$.

The Discrete Biorthogonal Wavelet Transform

The computation of the wavelet coefficients in a biorthogonal system does not differ much from the orthogonal case, the only difference being that we use a biorthogonal filter bank. The derivation is essential the same, and we therefore leave it entirely as an exercise to the reader.

In the biorthogonal case we start from the following approximation at scale 2^{-J}:

$$f_J(t) = \sum_k \langle f, \widetilde{\varphi}_{J,k}\rangle \varphi_{J,k}(t).$$

Note that we take inner products with the dual scaling functions $\widetilde{\varphi}_{J,k}$. This is a non-orthogonal projection onto V_J, along $\widetilde{V}_J^{\perp}$.

An approximation f_{j+1} can be split into two parts, a coarser approximation $f_j \in V_j$ and details $d_j \in W_j$:

$$f_{j+1}(t) = f_j(t) + d_j(t) = \sum_k s_{j,k}\varphi_{j,k}(t) + \sum_k w_{j,k}\psi_{j,k}(t).$$

The scaling coefficients $s_{j,k} = \langle f, \widetilde{\varphi}_{j,k}\rangle$ and wavelet coefficients $w_{j,k} = \langle f, \widetilde{\psi}_{j,k}\rangle$ are computed by feeding the scaling coefficients $s_{j+1,k} = \langle f, \widetilde{\varphi}_{j+1,k}\rangle$ into the biorthogonal filter bank:

$$s_{j,k} = \sqrt{2}\sum_l \widetilde{h}_{l-2k}s_{j+1,k} \quad \text{and} \quad w_{j,k} = \sqrt{2}\sum_l \widetilde{g}_{l-2k}s_{j+1,k}. \tag{4.32}$$

Repeating this recursively, starting with s_J, until a certain coarsest level j_0, we end up with

$$f_J(t) = \sum_{j=j_0}^{J-1}\sum_k \langle f, \widetilde{\psi}_{j,k}\rangle\psi_{j,k}(t) + \sum_k \langle f, \widetilde{\varphi}_{j_0,k}\rangle\varphi_{j_0,k}(t). \tag{4.33}$$

Letting $j_0 \to -\infty$ and $J \to \infty$, we get

$$f(t) = \sum_{j,k}\langle f, \widetilde{\psi}_{j,k}\rangle\psi_{j,k}(t).$$

To reconstruct the fine-scale coefficients s_J, we feed recursively scaling coefficients s_j and wavelet coefficients w_j into the synthesis part of the biorthogonal filter bank,

$$s_{j+1,k} = \sqrt{2}\sum_l (h_{k-2l}s_{j,k} + g_{k-2l}w_{j,k}).$$

Exercises 4.6

4.20. Derive one of the equations in (4.28), using the same kind of calculations as in the orthogonal case. Start with deriving the corresponding identity for the filter coefficients.

4.21. Verify that (4.21) are the perfect reconstruction conditions for filter banks.

4.22. Verify (4.30).

4.23. Show that the alternating flip construction leads to a biorthogonal filter bank, provided that the first equation in (4.28) is satisfied.

4.24. Let the lowpass filters be FIR with filter lengths M and $\widetilde{M}$. Assume that φ and $\widetilde{\varphi}$ are zero outside $[0, A]$ and $[0, \widetilde{A}]$. Use the scaling equations to show that $A = M - 1$ and $\widetilde{A} = \widetilde{M} - 1$. Then, use the wavelet equations to show that both ψ and $\widetilde{\psi}$ are zero outside $[0, (M + \widetilde{M} - 2)/2]$.

4.25. Derive the filter equations (4.32).

4.7 Approximation and Vanishing Moments

In this section, we will focus on the approximation properties of scaling functions and wavelets. Closely related to this is the property of the wavelets having vanishing moments, which produces the capability of wavelets to compress signals.

Approximation Properties of Scaling Functions

The scaling functions are to be used to approximate general functions. Therefore they need to have certain approximation properties.

It turns out that the scaling function φ has the approximation property (*cf.* also Chapter 12)

$$\|f - P_j f\| \leq C\, 2^{-j\alpha}\, \|D^\alpha f\|$$

for $\alpha \leq N - 1$, when it reproduces polynomials up to order $N - 1$:

$$\sum_k k^\alpha \varphi(t - k) = t^\alpha, \text{ for } \alpha = 0, \ldots, N - 1. \tag{4.34}$$

This means that[3] $t^\alpha \in V_j$, for each j and $\alpha = 0, \ldots, N - 1$. The integer N is the *order* of the multiresolution analysis.

Vanishing Moments

The polynomial reproducing property (4.34) is maybe not so interesting in its own right, but rather since it is connected to the dual wavelets having vanishing moments. If $t^\alpha \in V_j$, we then have $t^\alpha \perp \widetilde{W}_j$, since $V_j \perp \widetilde{W}_j$. This

[3] With a slight abuse of notation, since t^α does not belong to $L^2(\mathbb{R})$

means that $\langle t^\alpha, \widetilde{\psi}_{j,k}\rangle = 0$, for every wavelet $\widetilde{\psi}_{j,k}$. Written out more explicitly, we have

$$\int t^\alpha \widetilde{\psi}_{j,k}(t)\,dt = 0, \text{ for } n = 0, \dots, N-1.$$

We say that the dual wavelets have N *vanishing moments*. Having N vanishing moments can equivalently be stated as the Fourier transform having a zero of order N at $\omega = 0$,

$$D^\alpha \widehat{\widetilde{\psi}}(0) = 0, \text{ for } n = 0, \dots, N-1.$$

Using the relation $\widehat{\widetilde{\psi}}(2\omega) = \widetilde{G}(\omega)\widehat{\widetilde{\varphi}}(\omega)$ and $\widehat{\widetilde{\varphi}}(0) = 1$ we see that $\widetilde{G}(\omega)$ must have a zero of order N at $\omega = 0$. From (4.30) it then follows that $H(\omega)$ must be of the form

$$H(\omega) = \left(\frac{e^{-i\omega}+1}{2}\right)^N Q(\omega), \tag{4.35}$$

for some 2π-periodic function $Q(\omega)$. The larger N we choose, the sharper transition between pass-band and stop-band we get.

In the same way, we can show that the filter function $\widetilde{H}(\omega)$ must be of the form

$$\widetilde{H}(\omega) = \left(\frac{e^{-i\omega}+1}{2}\right)^{\widetilde{N}} \widetilde{Q}(\omega), \tag{4.36}$$

where $\widetilde{N}$ denotes the number of vanishing moments for ψ, or the order of the dual multiresolution analysis. The factorizations (4.35) and (4.36) are a starting point in the design of filter banks. The functions $Q(\omega)$ and $\widetilde{Q}(\omega)$ then have to be chosen so that the biorthogonality conditions are satisfied. Also, different properties of the wavelets depend on this choice. This is discussed further in Chapter 8.

It is the vanishing moments that give wavelets the compression ability. We will give a brief explanation here. Let $\widetilde{\psi}_{j,k}$ be a (fine-scale) wavelet with N vanishing moments. We may then think of $\widetilde{\psi}_{j,k}$ as being zero outside a small interval of size proportional to 2^{-j}. Suppose that the signal has α continuous derivatives on that interval. Then it can be approximated there with an $(\alpha-1)$-degree Taylor polynomial $P_{\alpha-1}(t)$ with error of order $O(2^{-j\alpha})$,

and we get

$$\begin{aligned}\langle f, \widetilde{\psi}_{j,k}\rangle &= \int f(t)\widetilde{\psi}_{j,k}(t)\,dt \\ &= \int P_{\alpha-1}(t)\widetilde{\psi}_{j,k}(t)\,dt + O(2^{-j\alpha}) \\ &= O(2^{-j\alpha}),\end{aligned}$$

provided that $\alpha \leq N$. So where the signal is smooth, the fine-scale wavelet coefficients will be almost 0, and need not be stored. We need only keep the fine-scale coefficients, where the signal has abrupt changes, discontinuities *etc.*

Vanishing moments of the dual wavelet $\widetilde{\psi}$ are also connected to the regularity of the mother wavelet ψ. It can be shown that the dual wavelet must have at least as many vanishing moments as the mother wavelet has derivatives. More vanishing moments leads to smoother mother wavelets. The regularity of ψ also depends on the choice of Q. The number of dual vanishing moments $\widetilde{N}$ for the wavelet ψ is not as important as N, since we do not take scalar products with the wavelets $\psi_{j,k}$. Further, $\widetilde{N}$ affects the regularity of the dual wavelets, and this is not as important as the regularity of the wavelets $\psi_{j,k}$, which are used as basis functions in the wavelet decomposition. Therefore, N is usually chosen larger than $\widetilde{N}$.

Exercises 4.7

4.26. Assuming that the function $H(\omega)$ has a zero of order N at $\omega = \pi$, show, using scaling, that $D^\alpha\widehat{\varphi}(2k\pi) = 0$ for all integers $k \neq 0$ and $0 \leq \alpha \leq N-1$. Show further that $D^\alpha\widehat{\widetilde{\psi}}(4k\pi) = 0$ for all integers k and $0 \leq \alpha \leq N-1$.

4.27. Show, using Poisson's summation formula and the previous exercise, that for $0 \leq \alpha \leq N-1$

$$\sum_k k^\alpha \varphi(t-k) = t^\alpha$$

when the function $H(\omega)$ has a zero of order N at $\omega = \pi$. The scaling function φ thus reproduces polynomials of degree at most $N-1$.

4.28. Prove, using Poisson's summation formula and scaling, the two identities

$$\begin{aligned}\sum_k (-1)^k \psi(t-k) &= \sum_l \psi(t/2 - l + 1/4),\\ \sum_k (-1)^k \varphi(t-k) &= \sum_l (-1)^l \psi(t - l - 1/2).\end{aligned}$$

4.8 Notes

The idea of approximating an image at different scales, and storing the difference between these approximations, appeared already in the pyramid algorithm of Burt and Adelson 1983. At the same time, the theory of wavelets had started to make progress, and several wavelet bases had been constructed, among others by the French mathematician Yves Meyer. This made the French engineer Stephane Mallat realize a connection between wavelets and filter banks. Together with Meyer he formulated the definition of multiresolution analyses. This connection led to a breakthrough in wavelet theory, since it gave both new constructions of wavelet bases and fast algorithms. The Belgian mathematician Ingrid Daubechies constructed the first family of wavelets within this new framework in 1988, and many different wavelets have been constructed since.

The overview article by Jawerth and Sweldens [20] is a good start for further reading and understanding of MRA and wavelets. It also contains an extensive reference list. The books *Ten Lectures on Wavelets* by Ingrid Daubechies [11] and *A First Course on Wavelets* by Hernandez & Weiss [16], give a more mathematically complete description. Among other things, they contain conditions on low- and highpass filters to generate Riesz bases of wavelets. A detailed discussion about this can also be found in the book by Strang & Nguyen [27].

Chapter 5

Wavelets in Higher Dimensions

Grey-scale images can be represented by functions of two variables. At each point (x, y) in the image, $f(x, y)$ is the grey-scale value at that point. In order to use wavelets for image processing we therefore need to extend the wavelet transform to functions of several variables.

There are basically two ways of doing this, separable or non-separable wavelet transforms. Separable transforms are constructed automatically from one-dimensional transforms, and are in general easier to work with. We will focus mainly on separable transforms.

However, the separable bases are connected to symmetries under rotations by 90^o only. If greater rotational symmetry is desired, the constructions (non-separable) become more involved.

5.1 The Separable Wavelet Transform

Separable wavelets are directly constructed from one-dimensional wavelets. We will only consider two dimensions, but it should be obvious how the construction extends to higher dimensions. We start with the most simple case.

The Two-dimensional Haar System

Let us consider a function $f_1(x, y)$ that is piecewise constant on the squares $k_x/2 < x < (k_x + 1)/2$, $k_y/2 < y < (k_y + 1)/2$ with values $s_{1,k}$. Here we use the multi-index notation $k = (k_x, k_y) \in \mathbb{Z}^2$; $k_x, k_y \in \mathbb{Z}$. We can think of $s_{1,k}$ as the pixel values in a digital grey-scale image. We try to reduce the amount of information by approximating four neighbouring values with their

mean value

$$s_{0,k} = \frac{1}{4}(s_{1,2k} + s_{1,2k+e_x} + s_{1,2k+e_y} + s_{1,2k+e}), \tag{5.1}$$

where $e_x = (1,0)$, $e_y = (0,1)$, and $e = (1,1)$.

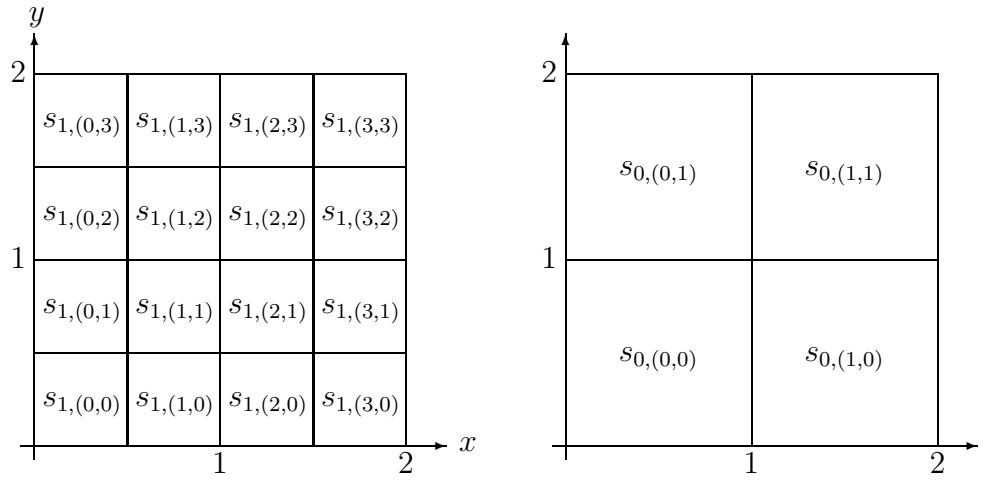

Figure 5.1: Averages in the two-dimensional Haar transform

This amounts to approximate f_1 with a function f_0 that is piecewise constant on the squares $k_x < x < k_x + 1$, $k_y < y < k_y + 1$ (Figure 5.1). Obviously, we lose information doing this approximation, and the question is how this information can be represented. In the one-dimensional Haar transform, we replace two neighbouring scaling coefficients with their mean value, and then store the lost information as the difference. In the present case, we replace four values with their mean value, and therefore we need three 'differences'. In the Haar system, we compute

$$\begin{aligned} w^H_{0,k} &= \frac{1}{4}(s_{1,2k} + s_{1,2k+e_x} - s_{1,2k+e_y} - s_{1,2k+e}), \\ w^V_{0,k} &= \frac{1}{4}(s_{1,2k} - s_{1,2k+e_x} + s_{1,2k+e_y} - s_{1,2k+e}), \\ w^D_{0,k} &= \frac{1}{4}(s_{1,2k} - s_{1,2k+e_x} - s_{1,2k+e_y} + s_{1,2k+e}). \end{aligned}$$

In Figure 5.2, we have scetched the 'polarity' of these differences, and the averaging. The superscripts H,V,D are shorthands for *horizontal*, *vertical*,

and *diagonal*, respectively. The coefficients w_0^H can be seen as differences in the y-direction, and they will be able to 'detect' horizontal edges: thus, the superscript H. Similarily, the coefficients w_0^V can be seen as differences in the x-direction, and vertical edges will show up in w_0^V. Diagonal edges will be present in w_0^D. Away from edges, these differences will generally be negligible. This is how compression becomes possible.

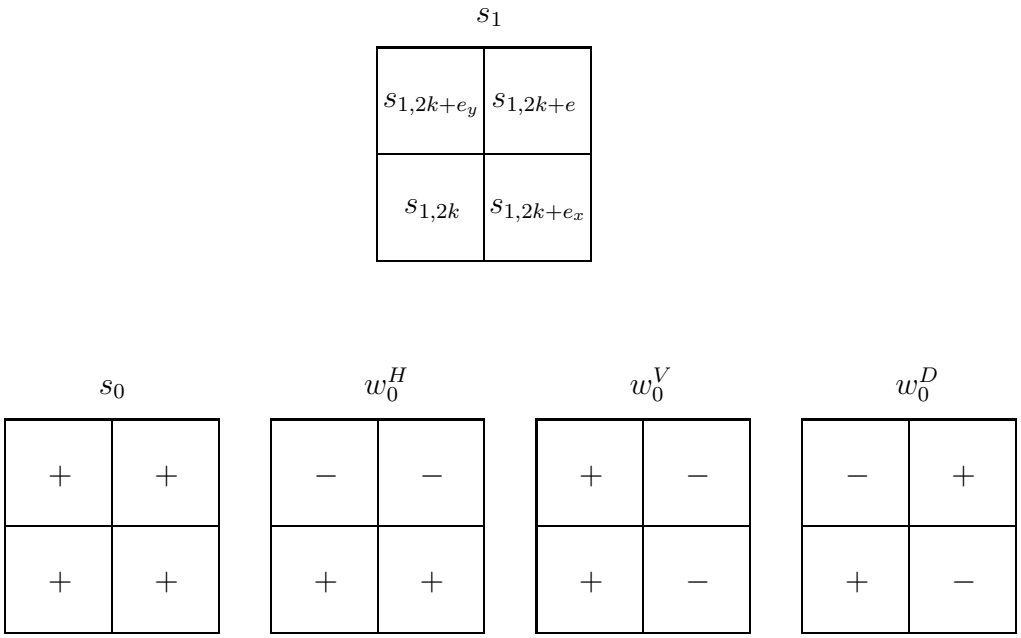

Figure 5.2: Averages and differences in the two-dimensional Haar transform

The averaging in Equation (5.1) is the low-pass filtering step in the two-dimensional Haar transform. It can be decomposed into two one-dimensional lowpass filtering steps. First, we apply a lowpass filtering with subsamling in the x-direction which gives us

$$\frac{1}{2}(s_{1,2k} + s_{1,2k+e_x}) \quad \text{and} \quad \frac{1}{2}(s_{1,2k+e_y} + s_{1,2k+e}).$$

Averaging these two, that is, averaging in the y-direction then gives us $s_{0,k}$. Further, the three wavelet coefficient sequences are the result of applying low- and highpass filters in the x- and y-directions. The whole process is shown in Figure 5.3, where H and G denote the Haar filters. The superscripts x and y indicate that filtering and subsampling are done in the x- and y-directions.

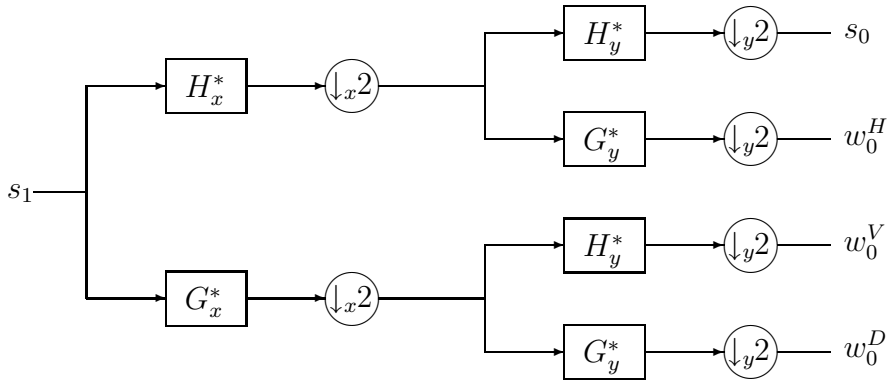

Figure 5.3: The two-dimensional Haar transform as two one-dimensional Haar transforms.

General Separable Wavelet Transforms

The decomposition of the two-dimensional Haar transform into two one-dimensional filtering steps provides a generic way to construct a two-dimensional wavelet transform from a one-dimensional. We start from a set of biorthogonal filters H, G, $\widetilde{H}$, and $\widetilde{G}$. We then define two-dimensional low- and highpass synthesis filters by

$$\begin{aligned} \mathbf{H} &= H_x H_y, \\ \mathbf{G}_H &= G_x H_y, \\ \mathbf{G}_V &= H_x G_y, \\ \mathbf{G}_D &= G_x G_y. \end{aligned}$$

Analysis filters $\widetilde{\mathbf{H}}$, $\widetilde{\mathbf{G}}_H$, $\widetilde{\mathbf{G}}_V$, and $\widetilde{\mathbf{G}}_D$ are defined analogously. For notational convenience, from now on, we let the operators H_x, H_y, *etc.* include upsampling. For instance, H_x is an upsamling in the x-direction followed by a filtering in the x-direction. Given scaling coefficients $s_{j+1} = (s_{j+1,k})_{k\in\mathbb{Z}^2}$, we compute averages and wavelet coefficients

$$\begin{array}{lll} s_j &= \widetilde{\mathbf{H}}^* s_{j+1} &= \widetilde{H}_y^* \widetilde{H}_x^* s_{j+1}, \\ w_j^H &= \widetilde{\mathbf{G}}_H^* s_{j+1} &= \widetilde{G}_y^* \widetilde{H}_x^* s_{j+1}, \\ w_j^V &= \widetilde{\mathbf{G}}_V^* s_{j+1} &= \widetilde{H}_y^* \widetilde{G}_x^* s_{j+1}, \\ w_j^D &= \widetilde{\mathbf{G}}_D^* s_{j+1} &= \widetilde{G}_y^* \widetilde{G}_x^* s_{j+1}. \end{array} \tag{5.2}$$

Here, the transposed operators include downsampling; $\widetilde{H}_x^*$ becomes a filtering in the x-direction with the reversed filter $\widetilde{H}^*$, followed by a downsampling. We thus compute scaling and wavelet coefficients by first applying the anal-

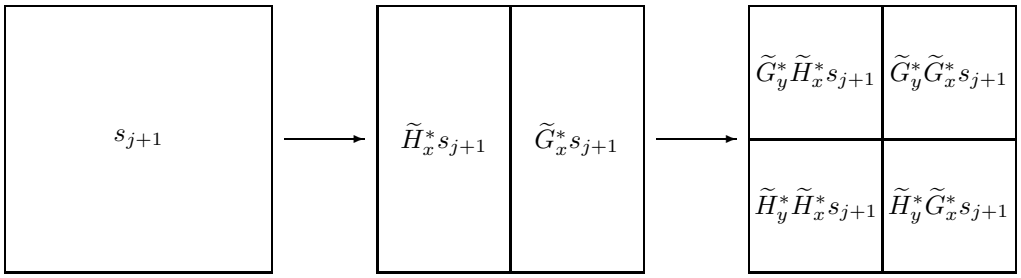

Figure 5.4: Separable filters.

ysis filter bank on the rows of s_{j+1} to get $\widetilde{H}_x^* s_{j+1}$ and $\widetilde{G}_x^* s_{j+1}$. We then apply the analysis filter bank along the columns of these, see Figure 5.4.

To reconstruct s_{j+1} from the scaling- and wavelet coefficients at the coarser scale, we reverse the whole process in Figure 5.4. Thus, we first apply the synthesis filters in the y-direction to recover $\widetilde{H}_x^* s_{j+1}$ and $\widetilde{G}_x^* s_{j+1}$:

$$\begin{aligned} \widetilde{H}_x^* s_{j+1} &= H_y s_j + G_y w_j^H, \\ \widetilde{G}_x^* s_{j+1} &= H_y w_j^V + G_y w_j^D. \end{aligned}$$

We then apply the synthesis filter bank in the x-direction to get

$$\begin{aligned} s_{j+1} &= H_x \widetilde{H}_x s_{j+1} + G_x \widetilde{G}_x s_{j+1} \\ &= H_x H_y s_j + H_x G_y w_j^H + G_x H_y w_j^V + G_x G_y w_j^D. \\ &= \mathbf{H} s_j + \mathbf{G}_H w_j^H + \mathbf{G}_V w_j^V + \mathbf{G}_D w_j^D. \end{aligned}$$

In the Forward Wavelet Transform, we recursively split the scaling coefficients s_j, starting from coefficients s_J at some finest scale. In Figure 5.5, two steps of the transform are shown, together with the typical organization of the wavelet coefficients. The Inverse Transform just reverses the whole

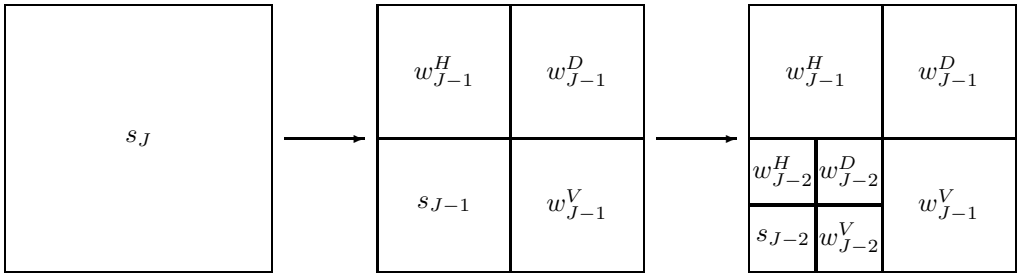

Figure 5.5: The separable wavelet transform.

process, using the synthesis filters.

We make a short remark here about the filtering process. The computation of, *e.g.*, s_j from s_{j+1} in (5.2) can be written out more explicitly as

$$s_{j,k} = 2 \sum_{l \in \mathbb{Z}^2} \widetilde{h}_{l_x - 2k_x} \widetilde{h}_{l_y - 2k_y} s_{j+1,l}.$$

This is a filtering with the two-dimensional lowpass filter $\widetilde{\mathbf{h}}_k = \widetilde{h}_{k_x} \widetilde{h}_{k_y}$, followed by a 'wo-dimensional downsampling'. This downsampling operator removes all coefficients with 'odd' multi-indices as follows.

$$(\downarrow 2)_2 s = \begin{pmatrix} & \vdots & \vdots & \vdots & \\ \cdots & s_{-2,2} & s_{0,2} & s_{2,2} & \cdots \\ \cdots & s_{-2,0} & s_{0,0} & s_{2,0} & \cdots \\ \cdots & s_{-2,-2} & s_{0,-2} & s_{2,-2} & \cdots \\ & \vdots & \vdots & \vdots & \end{pmatrix}.$$

The wavelet coefficients are computed analogously using the highpass filters $\widetilde{\mathbf{g}}_k^H = \widetilde{h}_{k_x} \widetilde{g}_{k_y}$, $\widetilde{\mathbf{g}}_k^V = \widetilde{g}_{k_x} \widetilde{h}_{k_y}$, and $\widetilde{\mathbf{g}}_k^D = \widetilde{g}_{k_x} \widetilde{g}_{k_y}$. Further, the reconstruction of s_{j+1} can be seen as applying two-dimensional filters after a two-dimensional upsampling. This upsampling operator inserts zeroes as follows.

$$(\uparrow 2)_2 s = \begin{pmatrix} & \vdots & \vdots & \vdots & \vdots & \vdots & \\ \cdots & s_{-1,1} & 0 & s_{0,1} & 0 & s_{1,1} & \cdots \\ \cdots & 0 & 0 & 0 & 0 & 0 & \cdots \\ \cdots & s_{-1,0} & 0 & s_{0,0} & 0 & s_{1,0} & \cdots \\ \cdots & 0 & 0 & 0 & 0 & 0 & \cdots \\ \cdots & s_{-1,-1} & 0 & s_{0,-1} & 0 & s_{1,-1} & \cdots \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \end{pmatrix}.$$

Separable Filters in the Frequency Domain

We have been referring to the filters $\mathbf{H}$ and $\mathbf{G}_\nu$ as low- and highpass without any further explanation than their 'averaging' and 'differencing'. However, it is straightforward to see that their filter functions are given by (without upsampling)

$$\begin{aligned} \mathbf{H}(\xi,\eta) &= H(\xi)H(\eta), \\ \mathbf{G}_H(\xi,\eta) &= G(\xi)H(\eta), \\ \mathbf{G}_V(\xi,\eta) &= H(\xi)G(\eta), \\ \mathbf{G}_D(\xi,\eta) &= G(\xi)G(\eta). \end{aligned} \tag{5.3}$$

Ideally, $H(\omega)$ equals 1 on $[0, \pi/2]$ and 0 on $[\pi/2, \pi]$, and $G(\omega)$ equals 1 on $[\pi/2, \pi]$ and 0 on $[0, \pi/2]$. Then, $\mathbf{H}(\xi, \eta)$ and $\mathbf{G}_\nu(\xi, \eta)$ will decompose the frequency domain as in Figure 5.6. For non-ideal filters, there is of course overlapping between the frequency regions.

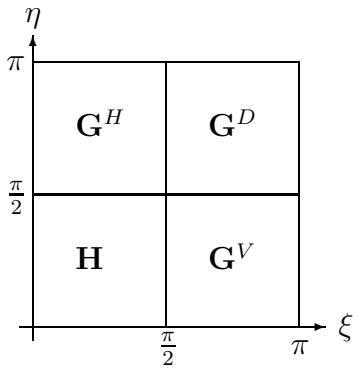

Figure 5.6: Frequency Support for Ideal Separable Filters.

Problems

5.1. Verify that the wavelet coefficients in the two-dimensional Haar transform are given by two one-dimensional filtering steps as described in Figure 5.3.

5.2. Work out the construction of separable filters in three dimensions. How many highpass filters will there be? (7)

5.3. Show that the two-dimensional downsampling operator can be written as $(\downarrow 2)_2 = (\downarrow 2)_x(\downarrow 2)_y$, where $(\downarrow 2)_x$ and $(\downarrow 2)_y$ are the one-dimemensional downsampling operators in the x- and y-directions. Also show that $(\uparrow 2)_2 = (\uparrow 2)_x(\uparrow 2)_y$.

5.2 Two-dimensional Wavelets

In one dimension, we started from the approximation and detail spaces, the scaling function, and the wavelet. From there, we derived filters and the fast wavelet transform. In the separable case, the natural starting point is the separable filters and the associated wavelet transform. We will now describe the scaling functions and wavelets associated with those filters, and the corresponding multiresolution analysis. The presentation is merely a sketch, and we refer to the exercises for the reader who wants to enter further into the details.

The Two-dimensional Haar Scaling Function and Wavelets

The two-dimensional Haar scaling function is defined by

$$\Phi(x,y) = \begin{cases} 1 & \text{when } 0 < x, y < 1, \\ 0 & \text{otherwise.} \end{cases}$$

With this function, we can write the piecewise constant function f_1 in the previous section as

$$f_1(x,y) = \sum_{k \in \mathbb{Z}^2} s_{1,k} \Phi(2x - k_x, 2y - k_y).$$

This is so, because $\Phi(2x - k_x, 2y - k_y)$ is one for $k_x/2 < x < (k_x + 1)/2$, $k_y/2 < y < (k_y + 1)/2$ and zero otherwise. The coarser approximation f_0 can similarly be written as

$$f_0(x,y) = \sum_{k} s_{0,k} \Phi(x - k_x, y - k_y).$$

In the one-dimensional Haar system, the difference $d_0 = f_1 - f_0$ can be written as a linear combination of wavelets that are piecewise constant with values ± 1. This is also the case in two dimensions, and, not surprisingly, we need three different wavelets Ψ^H, Ψ^V, and Ψ^D. They are defined by

$$\Psi^\nu(x,y) = \begin{cases} \pm 1 & \text{when } 0 < x, y < 1, \\ 0 & \text{otherwise.} \end{cases}$$

where ν is H, V, or D. They attain the values 1 and -1 according to the 'polarity' in Figure 5.2. With these wavelets, the 'detail image' d_0 can be written as

$$d_0(x,y) = \sum_{k} \sum_{\nu \in \{H,V,D\}} w^\nu_{0,k} \Psi^\nu(x - k_x, y - k_y). \tag{5.4}$$

Separable Scaling Functions and Wavelets

Just as for the filters, the Haar scaling function and wavelets can be expressed through their corresponding one-dimensional scaling function and wavelets. It is easy so see that

$$\begin{aligned} \Phi(x,y) &= \varphi(x)\varphi(y), \\ \Psi^H(x,y) &= \varphi(x)\psi(y), \\ \Psi^V(x,y) &= \psi(x)\varphi(y), \\ \Psi^D(x,y) &= \psi(x)\psi(y). \end{aligned}$$

This is also how wavelets and scaling functions are defined for general separable wavelet bases. We define dilated, translated, and normalized scaling functions by

$$\Phi_{j,k}(x,y) = 2^j\,\Phi(2^j x - k_x, 2^j y - k_y), \text{ where } j \in \mathbb{Z},\ k \in \mathbb{Z}^2$$

and similarly for the wavelets. Note that we need 2^j as the normalizing factor in two dimensions. We define approximation spaces $\mathbf{V}_j$ as the set of all functions of the form

$$f_j(x,y) = \sum_k s_{j,k}\,\Phi_{j,k}(x,y).$$

Detail spaces $\mathbf{W}_j^H$, $\mathbf{W}_j^V$, and $\mathbf{W}_j^D$ are defined analogously. The scaling function satisfies the scaling equation

$$\Phi(x) = 4\sum_k \mathbf{h}_k\,\Phi(2x - k_x, 2y - k_y).$$

We also have similar equations for the wavelets,

$$\Psi^\nu(x) = 4\sum_k \mathbf{g}_k^\nu\,\Phi(2x - k_x, 2y - k_y).$$

This implies that $\mathbf{V}_j \subset \mathbf{V}_{j+1}$ and $\mathbf{W}_j^\nu \subset \mathbf{V}_{j+1}$. Each $f_{j+1} \in \mathbf{V}_{j+1}$ can be decomposed as

$$f_{j+1} = f_j + d_j^H + d_j^V + d_j^D, \tag{5.5}$$

where $f_j \in \mathbf{V}_j$ and $d_j^\nu \in \mathbf{W}_j^\nu$. The function f_{j+1} can be written as

$$f_{j+1}(x,y) = \sum_k s_{j+1,k}\,\Phi_{j+1,k}(x,y), \tag{5.6}$$

and

$$f_j(x,y) = \sum_k s_{j,k}\,\Phi_{j,k}(x,y) \quad \text{and} \quad d_j^\nu(x,y) = \sum_k w_{j,k}^\nu\,\Psi_{j,k}^\nu(x,y). \tag{5.7}$$

To switch between (5.6) and (5.7) we use the analysis and synthesis filter banks (5.2) and (5.1).

Finally, we also have dual scaling functions, wavelets, approximation and detail spaces with the same properties described above. Together with the primal scaling functions and wavelets they satisfy biorthogonality conditions

similar to those in one dimensions. The biorthogonality conditions can be transposed to the filter domain. If we define

$$\mathbf{M}(\xi,\eta) = \begin{bmatrix} \mathbf{H}(\xi,\eta) & \mathbf{H}(\xi+\pi,\eta) & \mathbf{H}(\xi,\eta+\pi) & \mathbf{H}(\xi+\pi,\eta+\pi) \\ \mathbf{G}^H(\xi,\eta) & \mathbf{G}^H(\xi+\pi,\eta) & \mathbf{G}^H(\xi,\eta+\pi) & \mathbf{G}^H(\xi+\pi,\eta+\pi) \\ \mathbf{G}^V(\xi,\eta) & \mathbf{G}^V(\xi+\pi,\eta) & \mathbf{G}^V(\xi,\eta+\pi) & \mathbf{G}^V(\xi+\pi,\eta+\pi) \\ \mathbf{G}^D(\xi,\eta) & \mathbf{G}^D(\xi+\pi,\eta) & \mathbf{G}^D(\xi,\eta+\pi) & \mathbf{G}^D(\xi+\pi,\eta+\pi) \end{bmatrix},$$

and $\widetilde{\mathbf{M}}(\xi,\eta)$ similarly, we get

$$\overline{\mathbf{M}(\xi,\eta)}\widetilde{\mathbf{M}}(\xi,\eta)^{\mathrm{T}} = I. \tag{5.8}$$

It is easy to verify that the separable filters satisfy these equations. However, it is also possible to use these equations as the starting point for the construction of two-dimensional, non-separable wavelets. We will not explore this topic further, instead we take a look at two other, entirely different constructions of non-separable wavelets in the next section.

Separable Wavelets in the Frequency Domain

The Fourier transforms of the scaling function and the wavelets are given by

$$\begin{aligned} \widehat{\Phi}(\xi,\eta) &= \widehat{\varphi}(\xi)\widehat{\varphi}(\eta), \\ \widehat{\Psi^H}(\xi,\eta) &= \widehat{\varphi}(\xi)\widehat{\psi}(\eta), \\ \widehat{\Psi^V}(\xi,\eta) &= \widehat{\psi}(\xi)\widehat{\varphi}(\eta), \\ \widehat{\Psi^D}(\xi,\eta) &= \widehat{\psi}(\xi)\widehat{\psi}(\eta). \end{aligned}$$

We know that $\widehat{\varphi}$ is essentially localized to the frequency band $[0,\pi]$ and thus $\widehat{\Phi}(\xi,\eta)$ is essentially localized to the square $0 < \xi,\eta < \pi$ in the frequency plane. A similar argument for the wavelets gives that the separable wavelet transform corresponds to decomposition of the frequency plane as in Figure 5.7.

Exercises 5.2

5.4. Verify Equation (5.4).

5.5. Show that $\Phi_{j,k}(x,y) = \varphi_{j,k_x}(x)\varphi_{j,k_y}(y)$.

5.6. Show that the scaling function satisties the scaling equation

$$\Phi(x) = 4\sum_k \mathbf{h}_k\, \Phi(2x-k_x, 2y-k_y), \quad \text{where} \quad \mathbf{h}_k = h_{k_x}h_{k_y}.$$

Hint: Use the one-dimensional scaling function.

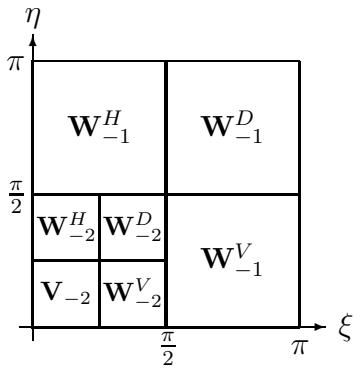

Figure 5.7: The frequency plane decomposition for the separable wavelet transform.

5.7. Show that $\mathbf{V}_j \subset \mathbf{V}_{j+1}$. Hint: Use the scaling equation to show that the scaling functions $\Phi_{j,k}$ belong to $\mathbf{V}_{j+1}$. From there, conclude that each finite sum

$$\sum_{k=-K}^{K} s_{j+1,k}\, \Phi_{j+1,k}(x,y)$$

belong to $\mathbf{V}_{j+1}$. Finally, since $\mathbf{V}_{j+1}$ is closed, it follows that all corresponding infinite sums, that is, elements of $\mathbf{V}_j$, belong to $\mathbf{V}_{j+1}$.

5.8. Show that we can decompose f_{j+1} as in (5.5). Hint: Show that each scaling function $\Phi_{j+1,k}$ can be decomposed in this way.

5.9. Show that separable filters satisfy the biortogonality conditions (5.8).

5.3 Non-separable Wavelets

A disadvantage with separable wavelets is their meagre rotational invariance. A strong emphasis is put on the coordinate axes. While variations in the coordinate directions are separated into the $\mathbf{W}_j^H$ and $\mathbf{W}_j^V$ frequency channels, variations in the two diagonal directions are mixed into the channels $\mathbf{W}_j^D$. A 45° rotation of an image can thus change the wavelet coefficients dramatically. We will now give two examples of more isotropic wavelets in two dimensions. We switch here to the notation $f(x)$, $x \in \mathbb{R}^2$, for convenience.

Quincunx Wavelets

For separable wavelets, the dilated, translated, and normalized scaling functions can be written as

$$\Phi_{j,k}(x) = 2^j \Phi(D^j x - k); \quad j \in \mathbb{Z}, k \in \mathbb{Z}^2.$$

Here, D is the *dilation matrix*

$$D = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}.$$

For Quincunx wavelets, we instead have scaling functions

$$\varphi_{j,k}(x) = 2^{j/2} \varphi(D^j x - k),$$

with the dilation matrix

$$D = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \quad \text{or} \quad D = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

Note that, for the first D, $D = \sqrt{2}R$, where R is an orthogonal matrix, a 45^o rotation. For the latter D, the rotation is followed by a reflection in the x_1-axis.

To simplify our discussion we will assume that we have the first D. In this case, when going from one scale to the next coarser, the scaling function is rotated 45^o and scaled in x by a factor $\sqrt{2}$. Approximation spaces V_j are defined in the usual way with the requirement $V_j \subset V_{j+1}$.

Let us look at the two approximation spaces V_0 and V_{-1}. Functions in V_0 can be written as linear combinations of the scaling functions $\varphi(x-k)$. Assuming that $\varphi(x-k)$ is localized in a region around $x = 0$, $\varphi(x-k)$ will be localized around $x = k$, and we associate this function with the point $k \in \mathbb{Z}^2$. The scaling functions $\varphi(D^{-1}x - k)$ in V_{-1} are in the same way associated with the points Dk, $k \in \mathbb{Z}^2$. We denote the set of these points $D\mathbb{Z}^2$. We sometimes refer to $\mathbb{Z}^2$ as the *sampling lattice* and $D\mathbb{Z}^2$ as the *subsampling lattice*.

In Figure 5.8, we have plotted the sampling and subsampling lattice for the Quincunx and the separable case. We see that there are 'twice as many' scaling functions in V_0 as in V_{-1} in the Quincunx case. Thus it seems plausible that the difference between V_0 and V_{-1} must be spanned by $\psi(D^{-1}x-k)$ for *one* wavelet ψ. This somewhat intuitive argument can in fact be made rigorous, but we will not do this here. Instead we observe that in the separable case there are 'four times as many' scaling functions in V_0 as in V_{-1},

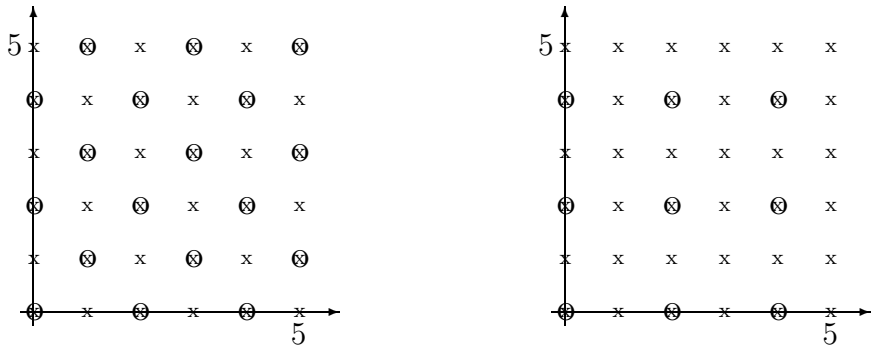

Figure 5.8: Sampling lattice (x's) and subsampling lattice (o's) for the Quincunx and the separable case.

and that we need *three* different wavelets. Dilated and translated wavelets can now be defined by

$$\psi_{j,k}(x) = 2^{j/2}\psi(D^j x - k); \quad j \in \mathbb{Z}, k \in \mathbb{Z}^2.$$

Their associated detail spaces are denoted W_j. The requirements $V_j \subset V_{j+1}$ and $W_j \subset V_{j+1}$ induces scaling and wavelet equations

$$\varphi(x) = \sum_k h_k \varphi(Dx - k) \quad \text{and} \quad \psi(x) = \sum_k g_k \varphi(Dx - k).$$

Again, the scaling function and wavelet are completely determined by the low and highpass filter coefficients (h_k) and (g_k). In the biorthogonal case, we also have dual scaling functions and wavelets, satisfying dual scaling and wavelet equations with dual filters $(\widetilde{h}_k)$ and $(\widetilde{g}_k)$. Biorthogonality conditions on wavelets and scaling functions can be transformed into conditions on the filters. In the frequency domain they become

$$\begin{array}{rcl}
\widetilde{H}(\xi_1,\xi_2)\overline{H(\xi_1,\xi_2)} + \widetilde{H}(\xi_1+\pi,\xi_2+\pi)\overline{H(\xi_1+\pi,\xi_2+\pi)} &=& 1, \\
\widetilde{G}(\xi_1,\xi_2)\overline{G(\xi_1,\xi_2)} + \widetilde{G}(\xi_1+\pi,\xi_2+\pi)\overline{G(\xi_1+\pi,\xi_2+\pi)} &=& 1, \\
\widetilde{G}(\xi_1,\xi_2)\overline{H(\xi_1,\xi_2)} + \widetilde{G}(\xi_1+\pi,\xi_2+\pi)\overline{H(\xi_1+\pi,\xi_2+\pi)} &=& 0, \\
\widetilde{H}(\xi_1,\xi_2)\overline{G(\xi_1,\xi_2)} + \widetilde{H}(\xi_1+\pi,\xi_2+\pi)\overline{G(\xi_1+\pi,\xi_2+\pi)} &=& 0.
\end{array} \tag{5.9}$$

The construction of biorthogonal Quincunx wavelets comes down to constructing filters satisfying these equations. One possible method is to start from a set of symmetric, biorthogonal filters in one dimension. They are all polynomials in $\cos\omega$, and replacing $\cos\omega$ with $\frac{1}{2}(\cos\xi_1 + \cos\xi_2)$ gives two-dimensional filters satisfying (5.9).

In the Forward Wavelet Transform, we start at a finest scale with $f_J \in V_j$, and recursively make the decomposition

$$\sum_k s_{j+1,k}\varphi_{j+1,k}(x) = f_{j+1}(x)$$
$$= f_j(x) + d_j(x) = \sum_k s_{j,k}\varphi_{j,k}(x) + \sum_k w_{j,k}\psi_{j,k}(x).$$

We compute s_j and w_j from s_{j+1} via the analysis filters

$$s_j = (\downarrow 2)_D \widetilde{H}^* s_{j+1} \quad \text{and} \quad w_j = (\downarrow 2)_D \widetilde{G}^* s_{j+1}$$

The downsampling operator $(\downarrow 2)_D$ removes all coefficients but those with indices $k \in D\mathbb{Z}^2$. These are sometimes referred to as *even indices* and all other indices are called *odd*. In the Inverse Wavelet Transform we successively recover s_{j+1} from s_j and w_j, using the synthesis filters,

$$s_{j+1} = H(\uparrow 2)_D s_j + G(\uparrow 2)_D w_j,$$

where the upsampling operator $(\uparrow 2)_D$ interleaves zeros at the odd indices.

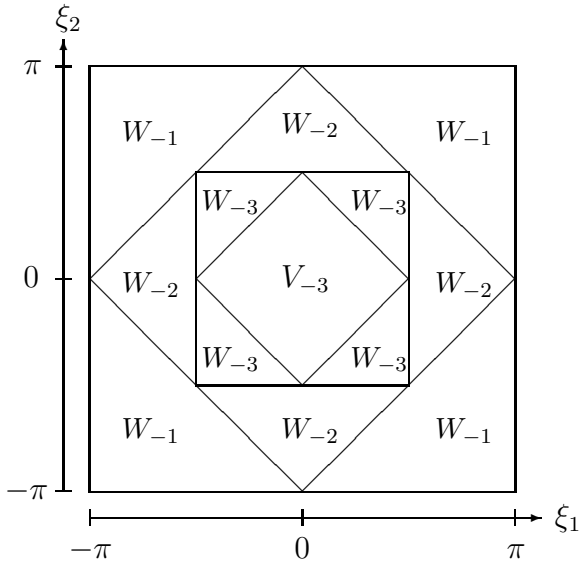

Figure 5.9: The ideal frequency plane decomposition for Quincunx wavelets.

Through their construction, Quincunx wavelets treat coordinate and diagonal directions equally, and we have achieved more rotational invariance compared to separable wavelets. A 45° rotation will (approximately) just move the frecuency channels one step: W_j becomes W_{j+1}. This is in sharp

contrast with the separable case, where the same rotation divides the frequency channel $\mathbf{W}^D$ into $\mathbf{W}^H$ and $\mathbf{W}^V$, and the two latter are mixed into $\mathbf{W}^D$. In Figure 5.9, we have plotted the frequency plane decomposition for Quincunx wavelets.

Hexagonal Wavelets

Quincunx wavelet achieved more rotational invariance by including a rotation in the dilation matrix. Another method is to use a sampling lattice different from $\mathbb{Z}^2$. One alternative is to use the *hexagonal lattice* $\mathbf{\Gamma}$ in Figure 5.10.

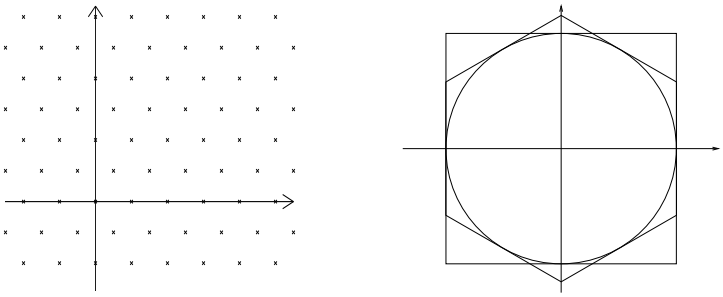

Figure 5.10: The Hexagonal Lattice (left) and the sampling frequency regions (right) for rectangular and hexagonal sampling.

An interesting feature with the hexagonal lattice is that it provides an optimal sampling density for band-limited functions with no directional preference. To be more precise, let f be circularly band-limited, $\widehat{f}(\xi) = 0$ for $|\xi| > \pi$. Sampling on the lattice $\mathbb{Z}^2$ is associated with the square $-\pi \leq \xi_1, \xi_2 \leq \pi$ in the frequency domain; a function that is band-limited in this frequency region can be exactly reconstructed from its sample values on the lattice $\mathbb{Z}^2$. Equivalently, a function sampled on the hexagonal lattice can be reconstructed from the sample values if it is band-limited to the hexagonal region in Figure 5.10. We see from this figure that, in the hexagonal case, we need to cover a smaller (13.4% less) area in the frequency domain compared to rectangular sampling, in order to recover f from its sample values. Those areas are proportional to the *sampling densities* associated to the lattices, *i.e.*, the number of sample points per unit area. Thus, using the hexagonal lattice we need 13.4% fewer sample values of the circularly band-limited function f compared to the usual rectangular sampling lattice.

The approximation space V_0 is spanned by scaling functions $\varphi(x - \gamma)$ where $\gamma \in \mathbf{\Gamma}$. We are thus translating φ along the hexagonal lattice. There will be one scaling function centred around each point in the hexagonal lat-

tice, assuming that φ is centred around the origin. The dilated, translated, and normalized scaling functions are defined as

$$\varphi_{j,\gamma}(x) = 2^j \varphi(2^j x - \gamma), \quad j \in \mathbb{Z}, \gamma \in \mathbf{\Gamma}.$$

It is also possible to have a dilation matrix D that includes a rotation, but we will only consider a pure scaling by 2 here. In this case, three wavelets ψ^1, ψ^2, and ψ^3 are needed to span the difference between V_1 and V_0.

Properly constructed, the scaling function will be invariant under 60° rotations and each wavelet will be associated with one of the directions in Figure 5.11. More precisely, ψ^1 will be oscillating in the x_1-direction, and $\psi^2(x) = \psi^1(R_{60°}x)$, $\psi^3(x) = \psi^1(R_{120°}x)$, where $R_{60°}$ and $R_{120°}$ denotes clockwise rotations 60° and 120°. In Figure 5.11 we have also sketched the frequency plane decomposition associated with those wavelets.

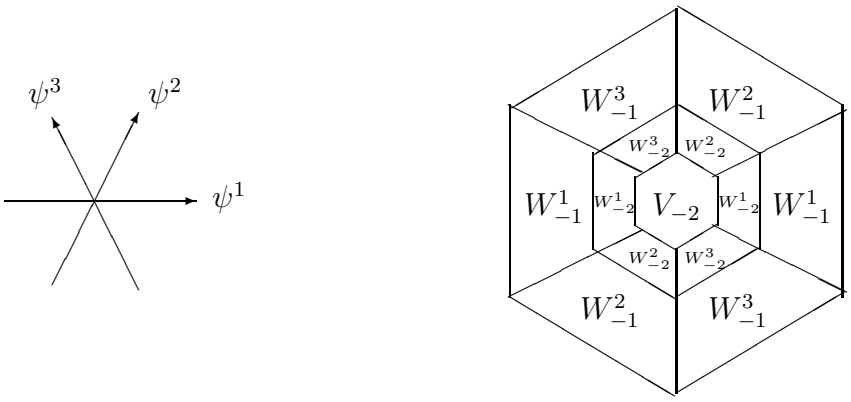

Figure 5.11: Hexagonal wavelets and their frequency plane decomposition.

The construction of such scaling functions and wavelets is based upon the construction of appropriate low- and highpass filters. The connection between these filters and scaling functions and wavelets is through scaling and wavelet equations. The filters have to satisfy certain biorthogonality (perfect reconstruction) conditions. We do not go into details here, and just point out that the filter design becomes considerably more difficult in the hexagonal lattice setting.

5.4 Notes

In the paper [9] by Cohen and Daubechies, several Quincunx wavelet bases are constructed and their regularity is investigated. An interesting fact is that for D being a rotation, orthogonal wavelets can at most be continuous, while

arbitrarily regular orthogonal wavelets are possible when D also includes a reflection. For biorthogonal wavelets, arbitrary regularity can be achieved for both choices of D. A few wavelets on the hexagonal lattice are constructed by Cohen and Schlenker in [10].

Finally, there exists a theory for the construction of scaling functions and wavelets for more general sampling lattices Γ and dilation matrices D. In the article by Strichartz [28] this theory is presented, along with a general construction of orthogonal wavelets. These wavelets can be made arbitrarily smooth, but they do not have compact support. In fact, they can be seen as a generalization of the Battle-Lemarié wavelets presented in section 8.4.

Chapter 6

The Lifting Scheme

Here we will consider the question of characterizing all compactly supported biorthogonal wavelet bases (*cf.* Chapter 3) sharing the same lowpass filter, say.[1] This will be done by using the Euclidean division algorithm on the decomposition of the filter 'polynomial' into even and odd terms.

Moreover, we will describe how any compactly supported biorthogonal filter bank can be built by successive 'lifting steps' starting from the trivial subsampling of elements with even and odd indices. This construction will follow from the factorization produced by the Euclidean algorithm just mentioned. This procedure has been applied to the construction of integer to integer MRA filter banks, which then allow lossless coding.

The successive lifting steps can also be used to tailor the filters to the signal by adopting a prediction scheme for the design of the highpass filter.

6.1 The Basic Idea

A Lifting Step

The basic idea behind the lifting scheme is to build larger filters from very simple ones through a sequence of *lifting steps*. In Figure 6.1 we show a lifting step. We feed s_{j+1} through the analysis part of a filter bank with filters h and g (down-sampling included) to obtain s_j and w_j. We then form 'new' scaling and wavelet coefficients

$$s_j^n = s_j \quad \text{and} \quad w_j^n = w_j - ps_j,$$

[1] For notational convience, we will let h and $\tilde{h}$ *etc.* denote the analysis and the synthesis filter functions, respectively, in this chapter. Elsewhere, the corresponding notation is $\tilde{H}$ and H, that is, the reversed one, in capital letters.

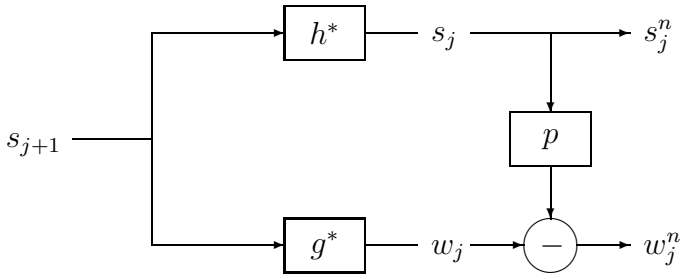

Figure 6.1: A lifting step.

where p is some filter. Notice that

$$w_j^n = g^* s_{j+1} - p h^* s_{j+1} = (g^* - p h^*) s_{j+1} = (g - h p^*)^* s_{j+1},$$

so s_j^n and w_j^n are the result of applying 'new' filters to s_{j+1},

$$s_j^n = h^{n*} s_{j+1} \quad \text{and} \quad w_j^n = g^{n*} s_{j+1},$$

where

$$h^n = h \quad \text{and} \quad g^n = g - h p^*.$$

To recover s_{j+1} from s_j^n and w_j^n we simply proceed as in Figure 6.2. This amounts to applying new (lifted) synthesis filters

$$\widetilde{h}^n = \widetilde{h} + \widetilde{g} p \quad \text{and} \quad \widetilde{g}^n = \widetilde{g},$$

and computing

$$s_{j+1} = \widetilde{h}^n s_j^n + \widetilde{g}^n w_j^n.$$

The connection between the original and lifted filters in the Fourier domain can be written out in matrix form

$$\text{(6.1a)} \qquad \begin{bmatrix} \widetilde{h}^n(\omega) \\ \widetilde{g}^n(\omega) \end{bmatrix} = \begin{bmatrix} 1 & s(\omega) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \widetilde{h}(\omega) \\ \widetilde{g}(\omega) \end{bmatrix},$$

$$\text{(6.1b)} \qquad \begin{bmatrix} \widetilde{h}^n(\omega) \\ g^n(\omega) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -s(\omega) & 1 \end{bmatrix} \begin{bmatrix} h(\omega) \\ g(\omega) \end{bmatrix},$$

where $s(\omega)$ is the filter function for p.

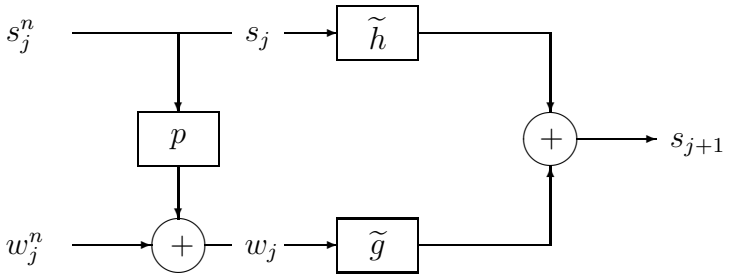

Figure 6.2: Inverting the lifted filter bank.

To motivate the lifting step, let us consider a simple example, where the initial filters are the lazy filters, that is,

$$s_{j,k} = s_{j+1,2k} \quad \text{and} \quad w_{j,k}^n = s_{j+1,2k+1}.$$

From a compression point of view, this is not a useful filter pair, since there are no reasons to expect many wavelet coefficients to be small. This is where the *prediction filter* p enters the scene. We try to predict the odd-indexed scaling coefficients from the even-indexed via linear interpolation:

$$\widehat{s}_{j+1,2k+1} = \frac{1}{2}(s_{j+1,2k} + s_{j+1,2k+2}), \text{or} \quad \widehat{w}_{j,k} = ps_{j,k} = \frac{1}{2}(s_{j,k} + s_{j,k+1}).$$

The new wavelet coefficients then become the prediction errors

$$\begin{aligned} w_{j,k}^n &= w_{j,k} - \widehat{w}_{j,k} \\ &= s_{j+1,2k+1} - \frac{1}{2}(s_{j+1,2k} + s_{j+1,2k+2}) \\ &= -\frac{1}{2}s_{j+1,2k} + s_{j+1,2k+1} - \frac{1}{2}s_{j+1,2k+2}. \end{aligned}$$

We see that the new highpass filter is $g_0^n = g_2^n = -1/2$, $g_1^n = 1$, all other $g_k^n = 0$.

In regions where the signal is smooth, the prediction can be expected to be accurate and thus the corresponding wavelet coefficients will be small. A more detailed analysis shows that the lifting step increases the number of vanishing moments, from zero in the lazy wavelet transform to two vanishing moments (linear polynomials will have zero wavelet coefficients).

Dual Lifting

After the lifting step, the wavelet coefficients changed, but the scaling coefficients were left unaltered. They can be updated with a *dual lifting step* as

shown in Figure 6.3. We apply an 'updating' filter u to the lifted wavelet coefficients, and get new scaling coefficients

$$s_j^n = s_j + uw_j^n.$$

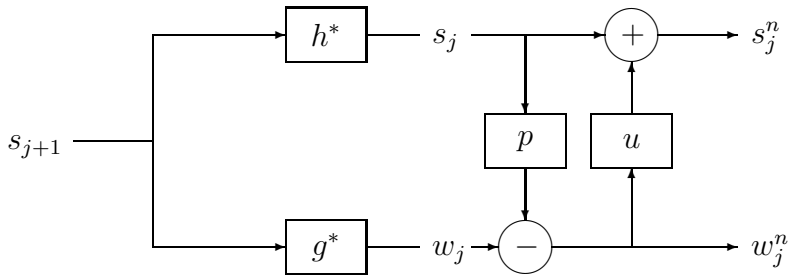

Figure 6.3: Dual lifting.

The reason for updating the scaling coefficents is to avoid aliasing. In our case, with the lazy filters, all signals with the same even-indexed scaling coefficients $s_{j+1,2k}$ will have the same scaling coefficients at the next level. To (partly) avoid this, we can force the mean value of the scaling coefficients to be constant, that is

$$\sum_k s_{j,k}^n = \sum_k s_{j+1,k}.$$

This is in fact equivalent to having one dual vanishing moment. We will not motivate this any further, instead we go on and make the *Ansatz*

$$\begin{aligned} s_{j,k}^n &= s_{j,k} + uw_{j,k}^n \\ &= s_{j,k} + Aw_{j,k}^n + Bw_{j,k-1}^n \\ &= -B/2\, s_{j+1,2k-2} + Bs_{j+1,2k-1} + (1 - A/2 - B/2)s_{j+1,2k} \\ &\quad + As_{j+1,2k+1} - A/2\, s_{j+1,2k+2}. \end{aligned}$$

It is not hard to show that $A = B = 1/4$ gives the constant mean value property. In this case, the new lowpass synthesis filter become $h_{-2}^n = h_2^n = -1/8$, $h_{-1}^n = h_1^n = 1/4$, $h_0^n = 3/4$, all other $h_k^n = 0$. The new filters (h_k^n) and (g_k^n) are the analysis filters associated with the hat scaling function and the wavelet in Figure 4.7.

Dual lifting has the following effect on the filters:

$$\begin{bmatrix} \widetilde{h}^n(\omega) \\ \widetilde{g}^n(\omega) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -t(\omega) & 1 \end{bmatrix} \begin{bmatrix} \widetilde{h}(\omega) \\ \widetilde{g}(\omega) \end{bmatrix}, \tag{6.2a}$$

$$\begin{bmatrix} h^n(\omega) \\ g^n(\omega) \end{bmatrix} = \begin{bmatrix} 1 & t(\omega) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} h(\omega) \\ g(\omega) \end{bmatrix}, \tag{6.2b}$$

where $t(\omega)$ is the filter function for u.

6.2 Factorizations

As we have seen in Chapter 3, to any biorthogonal wavelet basis with finite support there corresponds 'polynomials', uniquely defined up to shifts and multiplication with scalars. Such a 'polynomial' may be written

$$h(z) = \sum_{k=0}^{L-1} h_k z^{-k}$$

(A shift produces $\sum_{k=0}^{L-1} h_k z^{-k-N} = \sum_{k=-N}^{L-1-N} h_{k+N} z^{-k}$.)

As we also noted in Chapter 3, there are three other polynomials $\tilde{h}, g, \tilde{g}$, such that

$$\begin{cases} h(z)\tilde{h}(z^{-1}) + g(z)\tilde{g}(z^{-1}) = 2 \\ h(z)\tilde{h}(-z^{-1}) + g(z)\tilde{g}(-z^{-1}) = 0 \end{cases}$$

These are the conditions for perfect reconstruction from the analysis. We may write them in modulation matrix notation as follows (with redundancy):

$$\begin{bmatrix} h(z^{-1}) & g(z^{-1}) \\ h(-z^{-1}) & g(-z^{-1}) \end{bmatrix} \begin{bmatrix} \tilde{h}(z) & \tilde{h}(-z) \\ \tilde{g}(z) & \tilde{g}(-z) \end{bmatrix} = 2 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Dividing each polynomial into even and odd parts as

$$h(z) = h_e(z^2) + z^{-1} h_o(z^2)$$

where

$$h_e(z^2) := (h(z) + h(-z))/2, \; h_o(z^2) := (h(z) - h(-z))/(2z^{-1})$$

we define the *polyphase matrix* P (and analogously $\tilde{P}$), through

$$P(z^2) := \begin{bmatrix} h_e(z^2) & g_e(z^2) \\ h_o(z^2) & g_o(z^2) \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1 & 1 \\ z & -z \end{bmatrix} \begin{bmatrix} h(z) & g(z) \\ h(-z) & g(-z) \end{bmatrix}$$

and the matrix equation can now be written

$$\tilde{P}(z^{-1})^t P(z) = I$$

where I is the identity matrix. We now shift and scale g and $\tilde{g}$ so that $\det P(z) \equiv 1$.[2]

Note that the basis is orthogonal precisely when $P = \tilde{P}$, *i.e.*, $h = \tilde{h}$ and $g = \tilde{g}$.

Looking at $\tilde{P}(z^{-1})^t P(z) = I$, we note that P (*i.e.*, h and g) determines $\tilde{P}$ (*i.e.*, $\tilde{h}$ and $\tilde{g}$) since the matrix $\tilde{P}(z^{-1})^t$ is the inverse of the matrix $P(z)$ and $\det P(z) = 1$.

We will even see that, given h such that h_e and h_o have no common zeroes (except 0 and ∞), such a P (and thus $\tilde{P}$) can be constructed, using the Euclidean division algorithm on the given h_e and h_o.

The Euclidean division algorithm for integers is now demonstrated in a a specific case, by which the general principle becomes obvious.

Example 6.1. We wish to find the greatest common divisor of 85 and 34 and proceed as follows

$$\begin{aligned} 85 &= 2 \cdot 34 + 17 \\ 34 &= 2 \cdot 17 + 0 \end{aligned}$$

using $2 \cdot 34 \leq 85 < 3 \cdot 34$. Now, clearly, 17 divides both 34 and 85, and is the greatest common divisor. We now proceed with 83 and 34 instead.

$$\begin{aligned} 83 &= 2 \cdot 34 + 15 \\ 34 &= 2 \cdot 15 + 4 \\ 15 &= 3 \cdot 4 + 3 \\ 4 &= 1 \cdot 3 + 1 \\ 3 &= 3 \cdot 1 + 0 \end{aligned}$$

This means that 1 is the greatest common divisor of 83 and 34, and they are thus relatively prime. In matrix notation this is

$$\begin{bmatrix} 83 \\ 34 \end{bmatrix} = \begin{bmatrix} 2 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 2 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 3 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 3 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

□

[2] This is possible, since $\det P(z)$ must have length 1 if the inverse $\tilde{P}(z^{-1})^t$ is to contain polynomials only. Scaling and shifts are thus made in unison.

For the polynomials in the polyphase matrix we may proceed analogously, but the sequence of division steps is no longer unique, as will be seen in the following examples. We are only interested in the case when the polynomials to be divided are 'relatively prime', in other words, have no common zeroes (except 0 and ∞).

Example 6.2. Set $h(z) = 1 + z^{-1} + 4z^{-2} + z^{-3} + z^{-4}$ and thus $h_e(z) = 1 + 4z^{-1} + z^{-2}, h_o(z) = 1 + z^{-1}$. We divide h_e by h_o by requiring that the length of the remainder is to be strictly less than the length of the h_o. This can be done in three ways:

$$\begin{aligned}
1 + 4z^{-1} + z^{-2} &= (1 + z^{-1})(1 + z^{-1}) + 2z^{-1} \\
1 + 4z^{-1} + z^{-2} &= (1 + 3z^{-1})(1 + z^{-1}) - 2z^{-2} \\
1 + 4z^{-1} + z^{-2} &= (3 + z^{-1})(1 + z^{-1}) - 2
\end{aligned}$$

This division is thus not unique. Let us choose the first division and continue the algorithm:

$$1 + z^{-1} = 2z^{-1}(1/2\, z + 1/2) + 0.$$

Here the algorithm stops when the remainder is 0. We may represent the steps of the algorithm as

$$\begin{bmatrix} 1 + 4z^{-1} + z^{-2} \\ 1 + z^{-1} \end{bmatrix} = \begin{bmatrix} 1 + z^{-1} & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1/2z + 1/2 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 2z^{-1} \\ 0 \end{bmatrix}$$

Note that the number of steps will be bounded by the length of the division, here 2. □

When the common factor has length 1, the two polynomials h_e and h_o have no zeroes in common, except possibly 0 and ∞.

Example 6.3. The Haar case may be represented by $h(z) = 1 + z^{-1}$ with $h_e(z) = h_o(z) \equiv 1$. The algorithm may then be represented by

$$\begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

which is just one step. □

We will eventually arrive at the following factorization below: (6.3).

$$\begin{bmatrix} h_e(z) \\ h_o(z) \end{bmatrix} = \prod_{i=1}^{k/2} \begin{bmatrix} 1 & s_i(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ t_i(z) & 1 \end{bmatrix} \begin{bmatrix} C \\ 0 \end{bmatrix}$$

We now present an argument leading to the factorization (6.3). From the examples we get the scheme

$$\begin{bmatrix} h_e(z) \\ h_o(z) \end{bmatrix} = \begin{bmatrix} q_1(z) & 1 \\ 1 & 0 \end{bmatrix} \cdots \begin{bmatrix} q_k(z) & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} r_k(z) \\ 0 \end{bmatrix}$$

where $r_k(z)$ is a greatest common factor in $h_e(z)$ and $h_o(z)$. The zeroes of $r_k(z)$ one precisely the common zeroes of $h_e(z)$ and $h_o(z)$, excepting 0 and ∞.

In the case that $h_e(z)$ and $h_o(z)$ appear in a column of a polyphase matrix $P(z)$, for which $\det P(z) = 1$, clearly any $r_k(z)$ must have length 1, *i.e.*, $r_k(z) = Cz^M$ for some integer M and some non-zero constant C. ($r_k(z)$ will always factor out of $\det P(z)$, since it is a factor in the first column.)

We now consider a polyphase matrix

$$P(z) = \begin{bmatrix} h_e(z) & g_e(z) \\ h_o(z) & g_o(z) \end{bmatrix}$$

with $\det P(z) = 1$. We note that

$$\begin{bmatrix} q_i(z) & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ q_i(z) & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & q_i(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

and that, if the length of h_e is strictly less than the length of h_o, then $q_2(z) \equiv 0$. Therefore, we may write

$$\begin{bmatrix} h_e(z) \\ h_o(z) \end{bmatrix} = \begin{bmatrix} 1 & q_1(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ q_2(z) & 1 \end{bmatrix} \cdots \begin{bmatrix} 1 & 0 \\ q_k(z) & 1 \end{bmatrix} \begin{bmatrix} r_k(z) \\ 0 \end{bmatrix}$$

when k is even, since $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}^2 = I$. If k is odd, we may add a factor $\begin{bmatrix} 1 & 0 \\ q_{k+1} & 1 \end{bmatrix}$ with $q_{k+1} = 0$.

Now the product has an even number of factors and we have the desired factorization (6.3):

$$\begin{bmatrix} h_e(z) \\ h_o(z) \end{bmatrix} = \prod_{i=1}^{k/2} \begin{bmatrix} 1 & s_i(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ t_i(z) & 1 \end{bmatrix} \begin{bmatrix} C \\ 0 \end{bmatrix} \tag{6.3}$$

if we shift both h_e and h_o with the factor z^M from the algorithm. (This means that g_e and g_o has to be shifted with z^{-M} to preserve $\det P(z) = 1$.)

6.3 Lifting

The factorization (6.3) gives a polyphase matrix $P^n(z)$ through

$$P^n(z) := \begin{bmatrix} h_e(z) & g_e^n(z) \\ h_o(z) & g_o^n(z) \end{bmatrix} := \prod_{i=1}^{k/2} \begin{bmatrix} 1 & s_i(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ t_i(z) & 1 \end{bmatrix} \begin{bmatrix} C & 0 \\ 0 & 1/C \end{bmatrix}$$

where the last scaling $1/C$ is chosen to give $\det P^n(z) = 1$. Here the superscript n indicate that g_e^n and g_o^n may not come from the same highpass filter g, which we started from in P. All we did was to use the Euclidean algorithm on h_e and h_0 without any regard to g.

Moreover, given a polyphase matrix $P(z)$, any $P^n(z)$ with the same $h(z)$, *i.e.*, identical first columns and $\det P^n(z) = \det P(z) = 1$, is thus related to $P(z)$ through

$$P^n(z) = P(z) \begin{bmatrix} 1 & s(z) \\ 0 & 1 \end{bmatrix}$$

for some polynomial $s(z)$. In the same way, any $P^n(z)$ with the same $g(z)$ as $P(z)$ and with $\det P^n(z) = 1$ satisfies

$$P^n(z) = P(z) \begin{bmatrix} 1 & 0 \\ t(z) & 1 \end{bmatrix}$$

for some polynomial $t(z)$. In these two cases P^n is said to be obtained from P by *lifting* and *dual lifting*, respectively.

In this terminology, we can now conclude that any polyphase matrix can be obtained from the trivial subsampling of even and odd indexed elements (with the trivial polyphase matrix I, *i.e.*, $h(z) = 1$ and $g(z) = z^{-1}$) by successive lifting and dual lifting steps and a scaling.

6.4 Implementations

We will now turn to how the factorization is implemented. The polyphase matrix $P(z)^t$ performs the analysis part of the transform. *E.g.*, with $x(z) = x_e(z^2) + z^{-1}x_o(z^2)$ as before,

$$P(z)^t \begin{bmatrix} x_e(z) \\ x_o(z) \end{bmatrix} = \begin{bmatrix} h_e(z)x_e(z) + h_o(z)x_o(z) \\ g_e(z)x_e(z) + g_o(z)x_o(z) \end{bmatrix}$$

represents the even numbered entries of the outputs $h(z)x(z)$ and $g(z)x(z)$ after subsampling.

Specifically, we will now discuss the Haar case with $h(z) = 1 + z^{-1}$.

Example 6.4. Using the algorithm on the Haar lowpass filter, $h(z) = 1 + z^{-1}$, we have $h_e(z) = h_o(z) = 1$, and obtain

$$\begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix}.$$

This gives

$$P^n(z) = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

i.e., $g^n(z) = z^{-1}$, and

$$\tilde{P}^n(z^{-1}) = (P^n(z)^t)^{-1} = \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix}$$

which means $\tilde{h}^n(z) = 1$ and $\tilde{g}^n(z) = -1 + z$. □

If we consider the usual Haar polyphase matrix

$$P(z) = \begin{bmatrix} 1 & -1/2 \\ 1 & 1/2 \end{bmatrix}$$

i.e., $h(z) = 1 + z^{-1}$ and $g(z) = -1/2 + 1/2z^{-1}$, then we have (identical first columns)

$$P^n(z) = P(z) \begin{bmatrix} 1 & s(z) \\ 0 & 1 \end{bmatrix}$$

where now $s(z) = 1/2$.

Equivalently,

$$P(z) = P^n(z) \begin{bmatrix} 1 & -1/2 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & -1/2 \\ 0 & 1 \end{bmatrix}.$$

Correspondingly, we get

$$\tilde{P}(z^{-1}) = \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1/2 & 1 \end{bmatrix}$$

This factorization into lifting and dual lifting steps is implemented as follows.

Denote the signal to be analyzed by $\{x_k\}_k$,[3] and let the successive low- and highpass components of it be $\{v_k^{(j)}\}_k$ and $\{u_k^{(j)}\}_k$ after, respectively, stage $j = 1, 2, \ldots$. In our example, this becomes for the analysis,

$$\begin{cases} v_k^{(1)} = x_{2k} \\ u_k^{(1)} = x_{2k+1} \end{cases}$$

$$\begin{cases} v_k^{(2)} = v_k^{(1)} + u_k^{(1)} \\ u_k^{(2)} = u_k^{(1)} \end{cases}$$

$$\begin{cases} v_k = v_k^{(2)} \\ u_k = -1/2 v_k^{(2)} + u_k^{(2)} \end{cases}$$

where each step corresponds to a matrix factor.

For the reconstruction, the steps are reversed,

$$\begin{cases} v_k^{(2)} = v_k \\ u_k^{(2)} = 1/2 v_k + u_k \end{cases}$$

$$\begin{cases} v_k^{(1)} = v_k^{(2)} - u_k^{(2)} \\ u_k^{(1)} = u_k^{(2)} \end{cases}$$

$$\begin{cases} x_{2k} = v_k^{(1)} \\ x_{2k+1} = u_k^{(1)} \end{cases}$$

This is an example of an integer-to-integer transform. Note also that such a factorization represents a reduction in the number of operations needed to execute the transform.

Exercises 6.4

6.1. Determine the polyphase matrix $P(z)$ for the lazy filter bank: $h(z) = \tilde{h}(z) = 1$ and $g(z) = \tilde{g}(z) = z^{-1}$. (identity)

6.2. Apply the Euclidean division algorithm to the polynomials

$$(1 + z^{-1})(1 + 2z^{-1}) = 1 + 3z^{-1} + 2z^{-2} \quad \text{and } 1 + z^{-1}.$$

[3] This was denoted by the letter s instead of x in Section 6.1.

6.3. Determine two distinct polyphase and dual polyphase matrices sharing the polynomial $h(z)$ in Example 6.2.

6.4. Consider what happens in Example 6.4 if $h(z) = 1 + z^{-1}$ is scaled to $1/\sqrt{2} + 1/\sqrt{2}\, z^{-1}$.

6.5 Notes

This chapter is based on the paper Daubechies & Sweldens [12]. For information about the construction of integer-to-integer transforms, see the paper by Calderbank & Daubechies & Sweldens & Yeo [6]. A practical overview can be found in an article by Sweldens & Schröder [29]. All these three papers contain many further references.

Chapter 7

The Continuous Wavelet Transform

The continuous wavelet transform is the prototype for the wavelet techniques, and it has its place among 'reproducing kernel' type theories.

In comparison to the discrete transform, the continuous one allows more freedom in the choice of the analyzing wavelet. In a way, the discrete wavelet transform in Chapter 4 is an answer to the question of when the dyadic sampling of the continuous transform does not entail loss of information.

This chapter may be a little less elementary, but the arguments are quite straightforward and uncomplicated.

7.1 Some Basic Facts

The continuous wavelet transform is defined by the following expression

$$W_\psi f(a,b) = \int_{-\infty}^{\infty} f(t)\psi\left(\frac{t-b}{a}\right)a^{-1/2}\,dt$$

Here we may take $\psi \in L^2, a > 0$, and ψ is real-valued, and absolutely integrable, for simplicity. The variable a gives a continuous set of scales (dilations), and b a continuous set of positions (translations). In the discrete transforms both these sets were discrete.

Note that this may be seen as a convolution. With the notation $\psi_a(t) = a^{-1/2}\psi(-t/a)$ we have

$$Wf(a,b) := W_\psi f(a,b) = f * \psi_a(b)$$

or, after a Fourier transformation in the variable b,

$$\mathcal{F}_b Wf(a,\beta) = \widehat{f}(\beta)\widehat{\psi_a}(\beta)$$

If the last relation is multiplied by a factor and integrated over a, we might obtain just a constant factor times $\widehat{f}(\beta)$. We may then perform an inverse Fourier transform, and so get f reproduced from Wf.

Let us multiply by the complex conjugate of $\widehat{\psi_a}$, and integrate using that ψ is real-valued:

$$\begin{aligned} &\int_0^\infty \mathcal{F}_b W f(a,\beta)\, \overline{\widehat{\psi_a}(\beta)}\, da/a^2 = \\ &= f(\beta) \int_0^\infty |\widehat{\psi_a}(\beta)|^2\, da/a^2 \\ &= f(\beta) \int_0^\infty |\widehat{\psi}(\omega)|^2\, d\omega/\omega := C_\psi f(\beta) \end{aligned} \tag{7.1}$$

where we now assume that

$$C_\psi = \int_0^\infty |\widehat{\psi}(\omega)|^2\, d\omega/\omega$$

is positive and finite. This is called *the admissibility condition*. It implies that $\widehat{\psi}(0) = 0$, since absolute integrability of ψ implies that $\widehat{\psi}$ is a continuous function. The condition can be interpreted to say that ψ must have a *cancellation property*: $\int \psi(t)\, dt = \widehat{\psi}(0) = 0$.

Thus (we disregard convergence issues here) by an inverse Fourier transform, we get the inversion formula

$$f(t) = C_\psi^{-1} \int_{-\infty}^\infty \int_0^\infty W f(a,b)\, a^{-1/2} \psi\left(\frac{t-b}{a}\right) da/a^2 db$$

Obviously, there are many real-valued functions $\psi \in L^1 \cap L^2$ with $0 < C_\psi < \infty$. It is when we require that already the sampled values of $W_\psi f(a,b)$, *e.g.*, the values $W_\psi f(2^j, k2^{-j})$, $j,k \in \mathbb{Z}$, should suffice, that more restringent conditions have to be made on ψ. Clearly, these sample values suffice when $\{\psi(2^j t - k)\}_{j,k}$ is a basis, since we are then back in Chapter 4.

Using the Parseval formula, we also note that

$$\int_{-\infty}^\infty \int_0^\infty |W f(a,b)|^2\, da/a^2\, db = C_\psi \int_{-\infty}^\infty |f(t)|^2\, dt \tag{7.2}$$

Example 7.1. Consider the wavelet $\psi(t) = te^{-t^2}$. Take $f(t) = H(t - t_0) - H(t - t_1)$ where $t_0 < t_1$, and $H(t)$ is the Heaviside unit jump function at $t = 0$.

Further

$$\begin{aligned} Wf(a,b) &= \int_{t_0}^{t_1} a^{-1/2}\psi\left(\frac{t-b}{a}\right)dt = \int_{t_0}^{t_1} a^{-1/2}\frac{t-b}{a}e^{-\left(\frac{t-b}{a}\right)^2}dt \\ &= \left[\frac{1}{2}a^{1/2}e^{-\left(\frac{t-b}{a}\right)^2}\right]_{t_0}^{t_1} = \\ &= \frac{1}{2}a^{1/2}\left(e^{-\left(\frac{t_1-b}{a}\right)^2} - e^{-\left(\frac{t_0-b}{a}\right)^2}\right) \end{aligned}$$

and thus $Wf(a,b)$ decays rapidly in b away from $b = t_0$ and $b = t_1$. Moreover, this effect is more marked for small values of a, *i.e.*, for small scales in t. The reader is asked to draw a picture of this.

Exercises 7.1

7.1. Work out what happens if the multiplication is done by the complex conjugate of $\widehat{\Psi_a}$ (a different function) instead of by the complex conjugate of $\widehat{\psi_a}$ in Equation (7.1).

7.2. Make the modifications to the definition of the continuous wavelet transform and to the reconstruction formula needed if ψ is allowed to be complex-valued.

7.3. Verify the Parseval type formula (7.2).

7.4. Putting $\psi(t) = te^{-t^2}$, check that $0 < C_\psi < \infty$, and let $f(t) = e^{-\epsilon(t-t_0)^2}$. Compute the transform $Wf(a,b)$ and consider how it depends on $\epsilon > 0$ and t_0.

7.5. Consider now the function $\psi(t) = t/(1+t^2)$, and check that it is an admissible wavelet. Let now $f(t) = H(t-t_0) - H(t-t_1)$ as in Example 7.1, compute $Wf(a,b)$, and compare with the result in the example.

7.2 Global Regularity

We will now describe how certain differentiability properties are reflected in properties of the continuous wavelet transform.

The formula (assuming $0 < C_\psi < \infty$)

$$\int_0^\infty \mathcal{F}_b Wf(a,\beta)\, a^{1/2}\widehat{\psi}(a\beta)\, da/a^2 = C_\psi f(\beta)$$

can also be used to characterize certain differentiability properties of f in terms of properties of its wavelet transform $Wf(a,b)$.

First, we consider the spaces H^s for $s \geq 0$.

Definition 7.1. The space H^s consists of 'all' functions f such that

$$(1 + |\cdot|^2)^{s/2}\widehat{f} \in L^2$$

with norm

$$\|f\|_{H^s} = \left(\int_{-\infty}^{\infty} |(1+|\omega|^2)^{s/2}\widehat{f}(\omega)|^2\, d\omega\right)^{1/2}$$

□

Since $\mathcal{F}(D^k f)(\omega) = (i\omega)^k \mathcal{F}f(\omega)$, it is clear that H^s consists of f with $D^k f \in L^2$ for all $k \leq s$, when s is an even positive integer.[1] For our purposes, it is enough to think of such f, which are infinitely differentiable and vanish outside a bounded interval.

Multiplying $\widehat{f}(\beta)$ by $|\beta|^s$ in the formula (7.1), we have

$$\begin{aligned} a^{-s}\mathcal{F}_b Wf(a,\beta) &= f(\beta)a^{-s}\widehat{\psi_a}(\beta) = \\ &= |\beta|^s f(\beta)\,|a\beta|^{-s}\widehat{\psi_a}(\beta). \end{aligned}$$

Squaring the absolute values, integrating, and using the Parseval formula, we obtain

$$\begin{aligned} &\int_{-\infty}^{\infty}\int_0^{\infty} |a^{-s}W_f(a,b)|^2\, da/a^2 db = \\ &= \int_{-\infty}^{\infty} |\mathcal{F}^{-1}(|\cdot|^s \mathcal{F}f)(t)|^2\, dt \int_0^{\infty} |\omega|^{-2s}|\widehat{\psi}(\omega)|^2\, d\omega/\omega \end{aligned}$$

if

$$0 < \int_0^{\infty} |\omega|^{-2s}|\widehat{\psi}(\omega)|^2\, d\omega/\omega < \infty.$$

This requirement on $\widehat{\psi}$ means essentially that $\widehat{\psi}(\omega) = o(|\omega|^s)$ as $\omega \to 0$ or

$$\int_{-\infty}^{\infty} D^k\psi(t)\, dt = 0 \quad \text{for } 0 \leq k \leq s.$$

[1] The definition of H^s needs to be made more precise, but we will only note that infinitely differentiable functions f which vanish outside a bounded set is dense in H^s with its standard definition.

Additional cancellation is thus required of ψ now, in proportion to the number of derivatives of the function to be analyzed.

With the aid of the arguments in Exercise 7.6, we may now conclude that there exist positive constants A and B such that

$$A \|f\|_{H^s}^2 \;\leq\; \int_{-\infty}^{\infty}\int_{0}^{\infty} |a^{-s} W f(a,b)|^2 \, da/a^2 db \;\leq\; B \|f\|_{H^s}^2$$

This means that having s derivatives in L^2 corresponds exactly to having the integral

$$\int_{-\infty}^{\infty}\int_{0}^{\infty} |a^{-s} W f(a,b)|^2 \, da/a^2 db$$

finite. This may be seen a growth condition on the transform $Wf(a,b)$ in the scale variable a. The transform has to be small on small scales (small a) exactly tuned with the factor a^{-s}.

Exercises 7.2

7.6. Show that positive constants C_1 and C_2 exist for which

$$C_1 \|f\|_{H^s} \;\leq\; \|f\| + \|\mathcal{F}^{-1}\{|\cdot|^s \mathcal{F} f\}\| \leq\; C_2 \|f\|_{H^s}$$

using the inequalities

$$\max(1, |\omega|^s) \leq (1 + |\omega|^2)^{s/2} \leq 2^{s/2} \max(1, |\omega|^s).$$

7.3 Local Regularity

We will now make a local regularity statement in contrast to the global one involving the Fourier transform and H^s. Let us assume that for a fixed t_0

$$|f(t) - f(t_0)| \leq C|t - t_0|^s$$

where $0 < s < 1$, say. This is usually called a *local Lipschitz condition* and s is a measure of the regularity. The adjective *local* refers to the condition being tied to the point t_0, and thus it is not global or uniform.

What does this local regularity condition on the function f imply for the wavelet transform $W_\psi f(a,b)$?

We note first that the cancellation property $\int \psi(t)\,dt = 0$ makes it possible to insert $f(t_0)$:

$$Wf(a,b) = \int_{-\infty}^{\infty} (f(t) - f(t_0))\, a^{-1/2} \psi\left(\frac{t-b}{a}\right) dt.$$

Thus

$$\begin{aligned} |Wf(a,t_0)| &\le C \int_{-\infty}^{\infty} |t-t_0|^s a^{-1/2} \left|\psi\left(\frac{t-t_0}{a}\right)\right| dt = \\ &= C a^{\frac{1}{2}+s} \end{aligned}$$

and

$$\begin{aligned} |Wf(a,b)| &\le \int_{-\infty}^{\infty} |f(t) - f(t_0)| a^{-1/2} \left|\psi\left(\frac{t-b}{a}\right)\right| dt + \\ &\quad + \int_{-\infty}^{\infty} |f(t_0) - f(b)| a^{-1/2} \left|\psi\left(\frac{t-b}{a}\right)\right| dt \\ &\le C a^{1/2} (a^s + |b-t_0|^s). \end{aligned}$$

Thus the local Lipschitz regularity condition implies a growth condition on the wavelet transform.

Conversely, going from a growth condition on the wavelet transform, there is the following local regularity result. Note that a global condition on f is also made, and that an extra logarithmic factor appears in the regularity estimate.

If f is, say, bounded and

$$|Wf(a,b)| \le C a^{1/2} (a^s + |b-t_0|^s)$$

it follows that

$$|f(t) - f(t_0)| \le C |t-t_0|^s \log |t-t_0|^{-1}$$

holds for all t close enough to t_0. We omit the argument leading to this result.

Exercises 7.3

7.7. Determine the exponent in the Lipschitz condition for the function $f(t) = t^s$ for $t > 0$ and $f(t) = 0$ otherwise. Calculate also the transform, using the Haar wavelet $\psi(t) = t/|t|$ for $0 < |t| < 1/2$ and $\psi(t) = 0$ elsewhere.

7.4 Notes

Further material may be found in, *e.g.*, the books by Holschneider [18], Kahane & Lemarié [21], and Meyer [23].

The connection between differentiability properties and discrete wavelet representations is described in Chapter 12.

Part II

Applications

Chapter 8

Wavelet Bases: Examples

So far we have only seen a few examples of wavelet bases. There is in fact a large number of different wavelet bases, and in a practical application it is not always an easy task to choose the right one. In this chapter, we will describe the most frequently used wavelet bases. We will also try to give some general advice on how to choose the wavelet basis in certain applications.

8.1 Regularity and Vanishing Moments

Most of the wavelets we will discuss in this chapter have compact support, and the corresponding filters are FIR. We will always use the alternating flip construction

$$G(\omega) = -e^{-i\omega}\overline{\widetilde{H}(\omega+\pi)} \quad \text{and} \quad \widetilde{G}(\omega) = -e^{-i\omega}\overline{H(\omega+\pi)},$$

or, expressed in the filter coefficients,

$$g_k = (-1)^k \widetilde{h}_{1-k} \quad \text{and} \quad \widetilde{g}_k = (-1)^k h_{1-k}.$$

The lowpass filters are factorized as

$$H(\omega) = \left(\frac{1+e^{-i\omega}}{2}\right)^N Q(\omega) \quad \text{and} \quad \widetilde{H}(\omega) = \left(\frac{1+e^{-i\omega}}{2}\right)^{\widetilde{N}} \widetilde{Q}(\omega).$$

Different wavelet families differ in the choice of the trigonometric polynomials Q and $\widetilde{Q}$. The parameters N and $\widetilde{N}$ can be varied within each family, and control the number of vanishing moments. The choice of these parameters is important. We saw earlier that the number N of vanishing moments for the analysis (dual) wavelet determines the rate of decay of the wavelet

coefficients where the signal is smooth. The vanishing moments thus produce the compression ability, and are also connected to the smoothness of the synthesis wavelet. We usually want the synthesis wavelets to have some smoothness, which is particularly important in compression applications. This is so because compression is achieved by, roughly speaking, leaving out terms in the sum

$$f(t) = \sum_{j,k} w_{j,k}\psi_{j,k}(t),$$

corresponding to 'small' wavelet coefficients $w_{j,k}$.

The human eye tends to detect this cancellation more easily if the synthesis wavelets are irregular. One the other hand, the more regularity and vanishing moments we want, the longer filter we have to use. This makes the synthesis wavelets more spread out in time, which may cause 'ringing' artifacts around discontinuities and edges.

Thus there is a trade-off between regularity/vanishing moments and filter length. We again remark that the number $\widetilde{N}$ of vanishing moments for the synthesis wavelets, is not as important as N. This is because we take scalar products with the dual wavelets, $w_{j,k} = \langle f, \widetilde{\psi}_{j,k} \rangle$, and not with the synthesis wavelets. For the same reason, the smoothness of the dual wavelets, which is connected to $\widetilde{N}$, is not so important. As a consequence, we usually assign more vanishing moments to the analysis wavelets. In a practical application, there is no obvious choice of N and $\widetilde{N}$.

8.2 Orthogonal Bases

The first choice that has to be made is whether to use an orthogonal basis or not. Orthogonality allows interpretation of the wavelet coefficients as an energy distribution. The energy of a signal f can be expressed as

$$\int_{-\infty}^{\infty} |f(t)|^2 \, dt = \sum_{j,k} |w_{j,k}|^2 .$$

The squared modulus of the wavelet coefficient $|w_{j,k}|^2$ can be viewed as a measure of f's energy content roughly in the time interval $(2^{-j}k, 2^{-j}(k+1))$ and frequency interval $(2^j\pi, 2^{j+1}\pi)$. Summing these energies over the whole time-frequency plane gives the total energy.

In the best basis algorithm in Chapter 9, it will be orthogonality that makes the entropy cost function additive.

Another application, in which orthogonality appears, is noise reduction. It is so, because white noise in the sample values will then be transformed into white noise in the wavelet coefficients. As a consequence, the noise will statistically be equally spread over all the wavelet coefficients, while the energy of the signal will (hopefully) be concentrated in a few large wavelet coefficients. This makes it possible to extract the signal by keeping large coefficients, and at the same time remove most of the noise by setting small coefficients to zero. We will return to this 'wavelet shrinkage denoising' in Chapter 10. Finally, orthogonality guarantees numerical stability in the wavelet transform, since the condition number of the wavelet transform then is one. Thus relative errors will not grow during the forward and inverse wavelet transform.

Orthogonal bases corresponds to $N = \widetilde{N}$ and $Q = \widetilde{Q}$. We are left with designing the trigonometric polynomial Q in the factorization

$$H(\omega) = \left(\frac{1+e^{-i\omega}}{2}\right)^N Q(\omega),$$

such that

$$|H(\omega)|^2 + |H(\omega+\pi)|^2 = 2. \tag{8.1}$$

Since $|Q(\omega)|^2$ is real-valued, it is a polynomial in $\cos\omega$ (Exercise 8.1). We write it as a polynomial in $\sin^2\omega/2 = (1-\cos\omega)/2$ instead,

$$|Q(\omega)|^2 = P\left(\sin^2\frac{\omega}{2}\right).$$

With $y = \sin^2\omega/2$, (8.1) then becomes

$$(1-y)^N P(y) + y^N P(1-y) = 2, \quad \text{for} \quad 0 \le y \le 1. \tag{8.2}$$

One solution to this equation is the $(N-1)$:th order Taylor polynomial of $(1-y)^{-N}$

$$P_N(y) = \sum_{k=0}^{N-1} \binom{N-1+k}{k} y^k = (1-y)^{-N} + O(y^N).$$

The product filter $H(z)H(z^{-1})$ obtained from P_N is usually referred to as the Daubechies-2N Product Filter. To realize that $P_N(y)$ is a solution, we observe that

$$(1-y)^N P_N(y) + y^N P_N(1-y) = 2 + (1-y)^N O(y^N) + y^N O((1-y)^N).$$

Consider the term $(1-y)^N O(y^N)+y^N O((1-y)^N)$. This must be a polynomial of degree at most $2N-1$. On the other hand, it has zeros of order N at both $y=0$ and $y=1$. Therefore, it must be identically 0, and so P_N is a solution to (8.2). It is possible to show that every other solution must be of the form

$$P(y) = P_N(y) + y^N R(1-2y), \tag{8.3}$$

where R is any odd polynomial, such that $P(y) \geq 0$ for $0 \leq y \leq 1$.

After choosing a suitable R, we can easily compute the polynomial $S(z) = Q(z)Q(z^{-1})$. From there, we can extract Q by *spectral factorization*.

We briefly review the spectral factorization algorithm here. The zeros of the polynomial $S(z)$ come in pairs z_i and z_i^{-1}. One of these zeros belongs to $Q(z)$, and the other one to $Q(z^{-1})$. Thus, the factor $(z-z_i)$ goes into one polynomial and $(z-z_i^{-1})$ goes into the other. Since there are two choices for each such pair, the spectral factorization is not unique. As we will see, different choices can give quite different wavelets.

Daubechies Orthogonal Wavelets

In the Daubechies orthogonal wavelet family, the lowpass filter is constructed by first choosing $R=0$ in (8.3). In the spectral factorization, zeros inside the unit circle always go into the polynomial $Q(z)$. This minimizes the variation of the phase of $Q(z)$ as z goes around the unit circle, and this is why these filters are called *minimum-phase*. The filter length is $2N$, and the mother wavelet and scaling function are zero outside $[0, 2N-1]$. The smoothness increases with N and the number of continous derivatives grows asymptotically as $0.2075N$. A disadvantage with the Daubechies wavelets is that they are very asymmetric, except for the Haar wavelet which corresponds to $N=1$. This can be seen in Figures 8.1 - 8.2, where we have plotted Daubechies wavelets for $N = 2, \ldots, 9$.

Symmlets

A way to make orthogonal wavelets less asymmetric is to do the spectral factorization without requiring the minimum-phase property. Instead of always choosing zeros inside the unit circle, we may choose zeros to make the phase as nearly linear as possible. The corresponding family of orthogonal wavelets is usually referred to as *least asymmetric wavelets* or *Symmlets*. They are clearly more symmetric than Daubechies wavelets, as can be seen in Figure 8.3. The price we have to pay is that the Symmlet with N vanishing moment is less regular than the corresponding Daubechies wavelet.

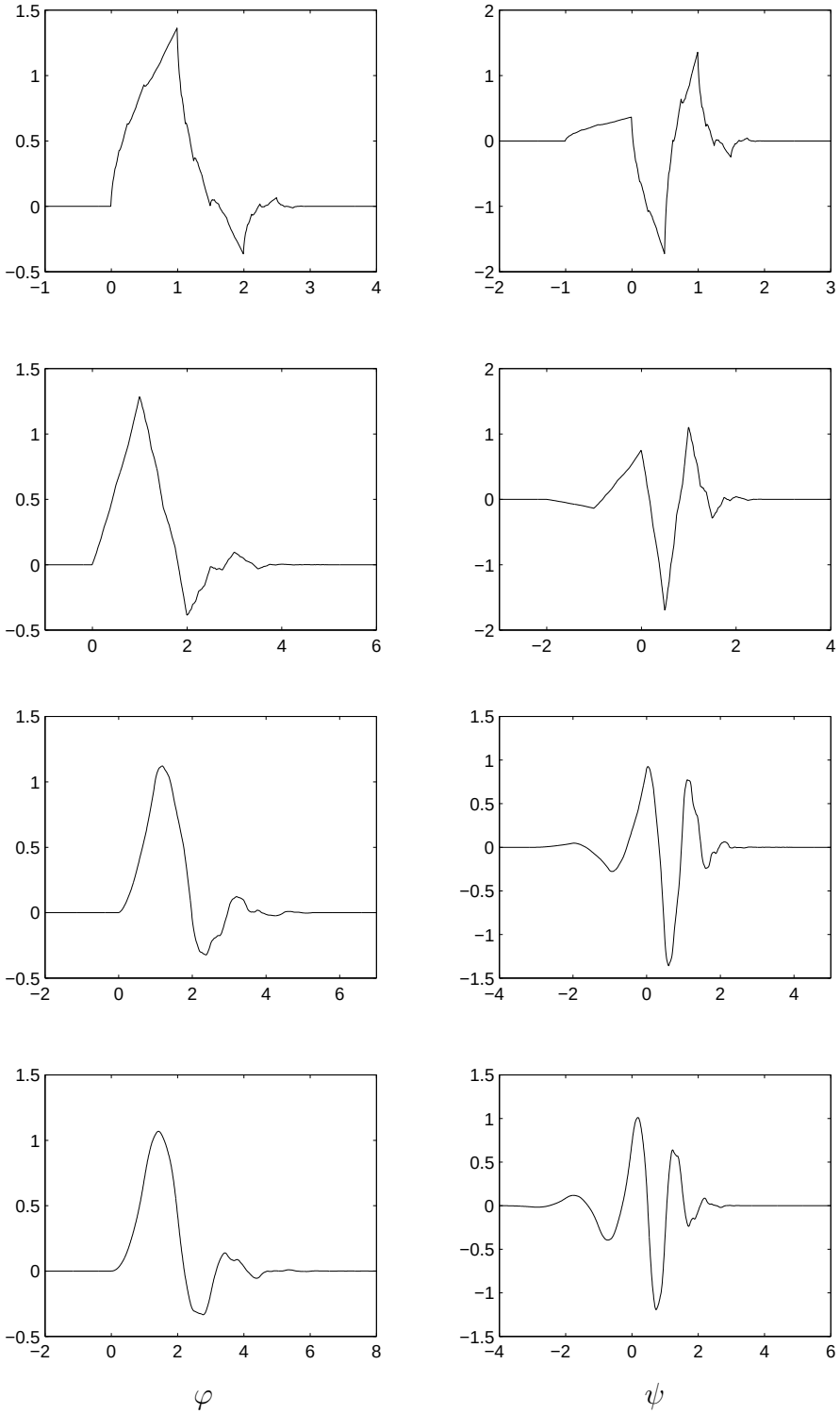

Figure 8.1: Daubechies orthogonal wavelets and scaling functions for $N = 2, 3, 4, 5$.

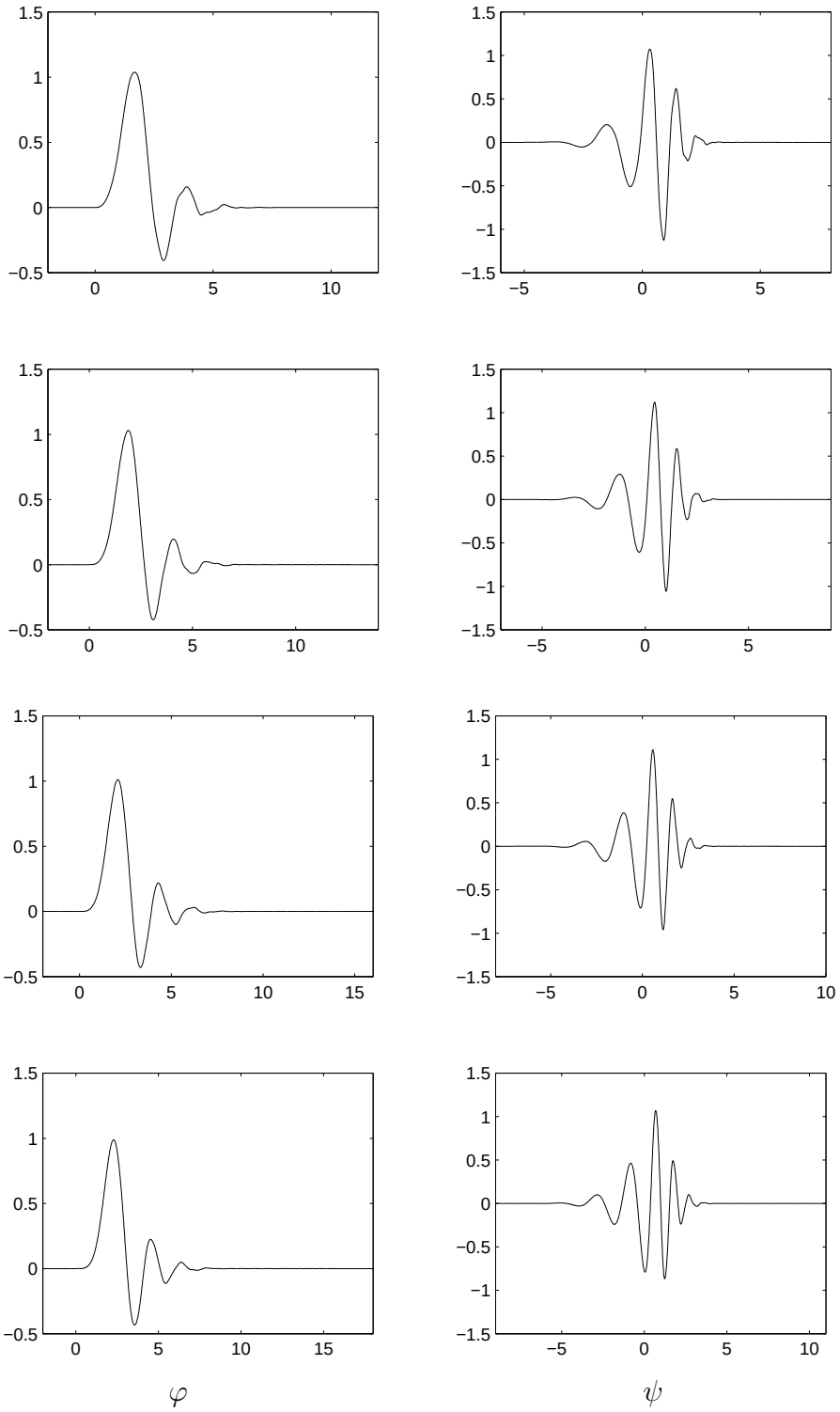

Figure 8.2: Daubechies orthogonal wavelets and scaling functions for $N = 6, 7, 8, 9$.

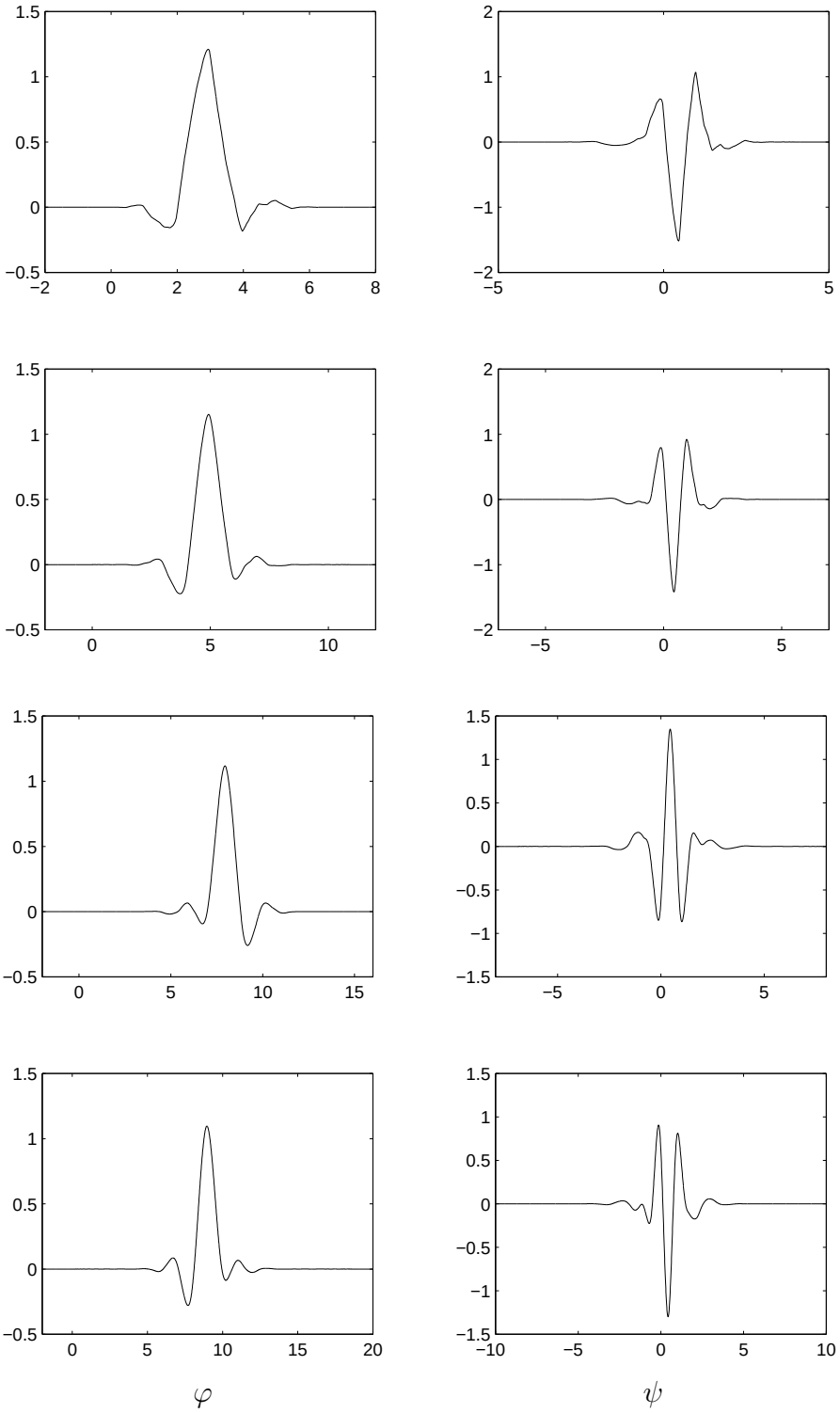

Figure 8.3: Symmlets and their scaling functions for $N = 2, 3, 4$, and 5.

Coiflets

Another family of orthogonal wavelets was designed to give also the scaling functions vanishing moments,

$$\int_{-\infty}^{\infty} t^n \varphi(t)\, dt = 0, \quad n = 1, \ldots, N-1.$$

This can be a useful property since, using a Taylor expansion, $\langle f, \varphi_{J,k} \rangle = 2^{J/2} f(2^{-J}k) + O(2^{-Jp})$ in regions where f has p continuous derivatives. For smooth signals, the fine-scale scaling coefficients can thus be well approximated by the sample values, and pre-filtering may be dispensed with. These wavelets are referred to as *Coiflets*. They correspond to a special choice of the polynomial R in (8.3). Their support width is $3N-1$ and the filter length is $3N$. In Figure 8.4, we have plotted the first four Coiflets together with their scaling functions. We see that the Coiflets are even more symmetric than the Symmlets. This is, of course, obtained to the price of the increased filter length.

The Cascade Algorithm

Before proceeding to biorthogonal wavelets, we describe how the plots in Figures 8.1 - 8.4 were obtained. This was carried out using the *cascade algorithm*. It is based on the fact that, for a continuous function f, $2^{j/2}\langle f, \varphi_{j,k} \rangle \to f(t_0)$, as $j \to \infty$, if $t_0 = 2^{-j}k$ is kept fixed. This is outlined in Exercise 8.4

If we want to compute sample values of a (continous) scaling function $\varphi_{j_0,l}$ we can thus compute its scaling coefficients at some fine scale $J > j_0$.

Algorithm. (**The Cascade Algorithm**)

1. Start with scaling coefficients of $\varphi_{j_0,l}$ at scale j_0 given by

$$s_{j_0,k} = \langle \varphi_{j_0,l}, \varphi_{j_0,k} \rangle = \delta_{k,l},$$
$$w_{j_0,k} = \langle \varphi_{j_0,l}, \psi_{j_0,k} \rangle = 0.$$

2. Compute scaling coefficients $s_{j,k} = \langle \varphi_{j_0,l}, \varphi_{j,k} \rangle$ and wavelet coefficients $w_{j,k} = \langle \varphi_{j_0,l}, \psi_{j,k} \rangle$ with the Fast Inverse Wavelet Transform.

3. Stop at some finest scale J and use $2^{J/2} s_{J,k}$ as approximations of $\varphi_{j_0,l}(2^{-J}k)$.

□

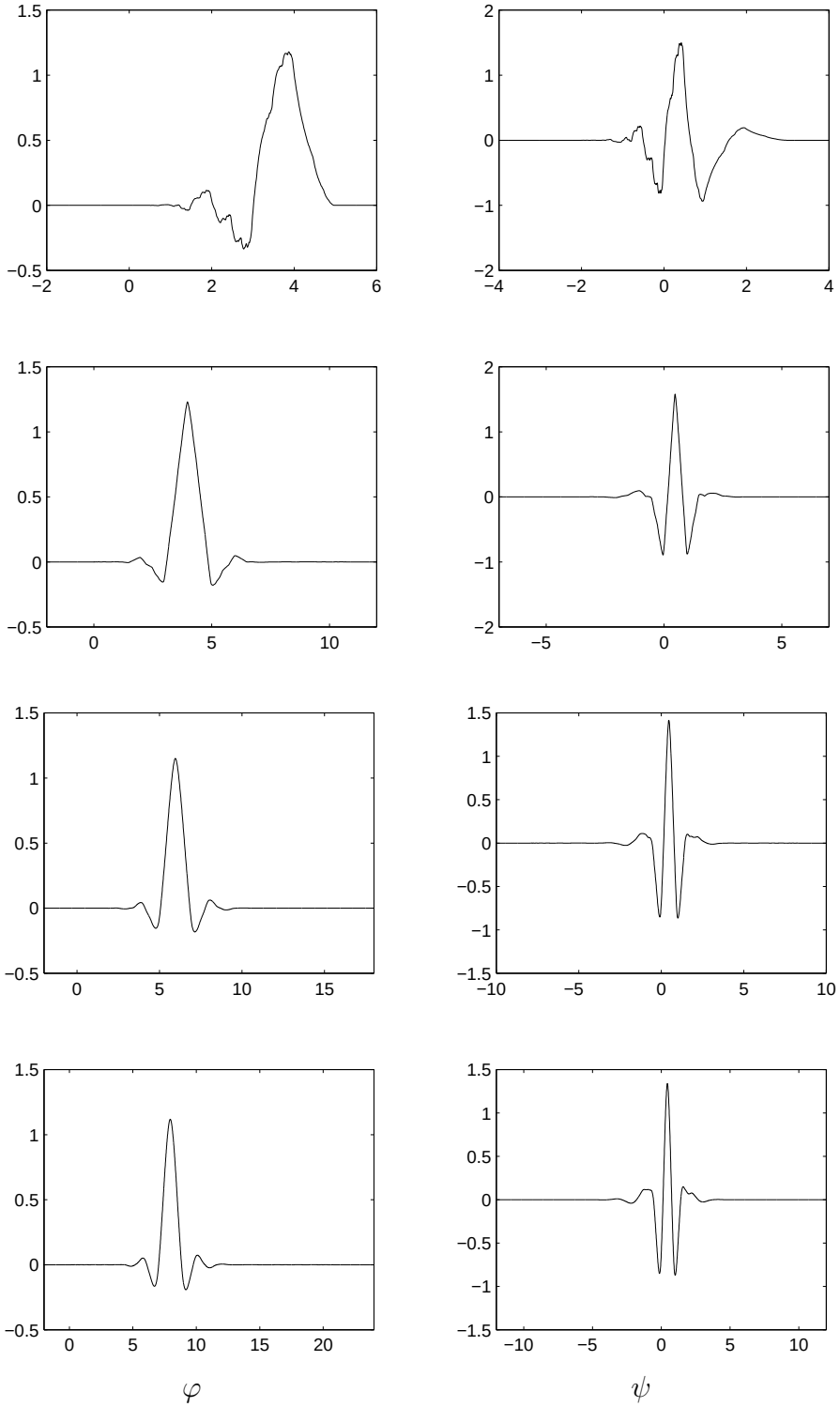

Figure 8.4: The first four Coiflets with scaling functions for $2, 4, 6$, and 8 vanishing moments.

Exercises 8.2

8.1. Show that $|Q(\omega)|^2$ is a polynomial in $\cos\omega$. Hint: It is a trigonometric polynomial,

$$|Q(\omega)|^2 = \sum_{k=-K}^{K} c_k e^{ik\omega} = \sum_{k=-K}^{K} c_k \cos k\omega + i \sum_{k=-K}^{K} c_k \sin k\omega.$$

The coefficients c_k are real (why?) and thus

$$0 = \operatorname{Im} |Q(\omega)|^2 = \sum_{k=-K}^{K} c_k \sin k\omega.$$

Finally, show that each $\cos k\omega$ is a polynomial in $\cos\omega$.

8.2. Verify that (8.1) is transformed into (8.2) with the substitutions

$$|Q(\omega)|^2 = P\left(\sin^2\frac{\omega}{2}\right) \quad \text{and} \quad y = \sin^2\frac{\omega}{2}.$$

8.3. Show that all solutions to (8.2) must be of the form (8.3). Hint: Assume that $P(y)$ is any solution. Show that we then must have

$$(1-y)^N[P(y) - P_N(y)] + y^N[P(1-y) - P_N(1-y)] = 0,$$

and that this implies that

$$P(y) - P_N(y) = y^N \widetilde{P}(y),$$

for some polynomial $\widetilde{P}$. Further, verify that $\widetilde{P}$ satisfies

$$\widetilde{P}(y) + \widetilde{P}(1-y) = 0.$$

Finally, conclude that $\widetilde{P}(y)$ must be anti-symmetric around $y = 1/2$ and thus can be written as $\widetilde{P}(y) = R(1-2y)$ for some odd polynomial R.

8.4. Show that if $t_0 = 2^{-j}k$ is held constant, and f is continuous, $2^{j/2}\langle f, \varphi_{j,k}\rangle \to f(t_0)$, as $j \to \infty$. Hint: Verify that

$$\begin{aligned} 2^{j/2}\langle f, \varphi_{j,k}\rangle &= 2^{j/2} \int_{-\infty}^{\infty} f(t)\varphi_{j,k}(t)\,dt \\ &= 2^{j/2} f(t_0) \int_{-\infty}^{\infty} \varphi_{j,k}(t)\,dt + 2^{j/2} \int_{-\infty}^{\infty} [f(t) - f(t_0)]\,\varphi_{j,k}(t)\,dt \\ &= f(t_0) + \int_{-\infty}^{\infty} \big[f(t_0 + 2^{-j}k) - f(t_0)\big]\,\varphi(t)\,dt. \end{aligned}$$

Since f is continuous, $f(t_0 + 2^{-j}k) - f(t_0) \to 0$ as $j \to \infty$, for each t. Therefore, the last integral goes to zero as well. (The latter statement is justified by the so called *Dominated Convergence Theorem*, but we believe that most readers will accept it anyway.)

8.3 Biorthogonal Bases

As mentioned above, it is impossible to have symmetry and orthogonality at the same time. However, using biorthogonal wavelets, symmetry is possible. In image processing, and in particular image compression, symmetry is important for the same reason as smoothness: leaving out a term in the sum

$$\sum_{j,k} w_{j,k}\psi_{j,k}(t)$$

is more easily detected by the human eye, if the synthesis wavelets $\psi_{j,k}$ are asymmetric. Also, symmetry is equivalent to filters having linear phase, *cf.* Section 2.4.

In general, we have more flexibility in the biorthogonal case, since we have two filters to design instead of one. Therefore, a general guideline could be to always use biorthogonal bases, unless orthogonality is important for the application at hand.

The filter design is also easier in the biorthogonal case. There are several different ways to design the filters. One method is to choose an arbitrary synthesis lowpass filter $H(z)$. For the analysis lowpass filter, we have to solve for $\widetilde{H}(z)$ in

$$H(z)\widetilde{H}(z^{-1}) + H(-z)\widetilde{H}(-z^{-1}) = 1. \tag{8.4}$$

It is possible to show that solutions exist, if, *e.g.*, $H(z)$ is symmetric and $H(z)$ and $H(-z)$ have no common zeros. These solutions can be found by solving linear systems of equations for the coefficients of $\widetilde{H}(z)$.

Another method is to apply spectral factorization to different product filters. If we want to design symmetric filters, both Q and $\widetilde{Q}$ must be symmetric. A detailed analysis shows that $N + \widetilde{N}$ must be even, and that we can write

$$Q(\omega) = q(\cos\omega) \quad \text{and} \quad \widetilde{Q}(\omega) = \widetilde{q}(\cos\omega)$$

for some polynomials q and $\widetilde{q}$. This is when Q and $\widetilde{Q}$ both have odd length. They can also both have even length, and then a factor $e^{-i\omega/2}$ must be included. In any case, if we define $L = (N + \widetilde{N})/2$ and the polynomial P through

$$P\left(\sin^2\frac{\omega}{2}\right) = q(\cos\omega)\widetilde{q}(\cos\omega),$$

the equation (8.4) transforms into

$$(1-y)^L P(y) + y^L P(1-y) = 2,$$

The solutions to this equation are known from the previous section. After choosing a solution, that is, choosing R in (8.3), Q and $\widetilde{Q}$ are computed using spectral factorization.

A third method to construct biorthogonal filters, is to use lifting, as described in Chapter 6.

Biorthogonal Spline Wavelets.

This is a family of biorthogonal wavelet bases, where the scaling function is a B-spline function of order N, *i.e.*, a convolution of the Haar scaling function with itself N times. The corresponding analysis lowpass filter is

$$H(\omega) = \left(\frac{1+e^{-i\omega}}{2}\right)^N.$$

The synthesis lowpass filter is

$$\widetilde{H}(\omega) = \left(\frac{1+e^{-i\omega}}{2}\right)^{\widetilde{N}} \sum_{k=0}^{L-1} \begin{pmatrix} L-1+k \\ k \end{pmatrix} \left(\sin^2\frac{\omega}{2}\right)^k.$$

The lowpass filters are based on a particular factorization of Daubechies' product filter ($R = 0$), namely $Q(z) \equiv 1$ and $\widetilde{Q}(z) = P_L(z)$.

These spline wavelets have many attractive features, in addition to being symmetric. The filter coefficients are *dyadic rational*, that is, of the form $2^j k$ where $j, k \in \mathbb{Z}$. This means that they can be exactly represented in a computer. Moreover, multiplication with these coefficients becomes a very fast operation, which is an advantage in computer implementations. Another nice property is that we have analytic expressions for φ and ψ, something that is not available in general.

A major disadvantage, however, is the large difference in filter length between the lowpass filters. This occurs because we put the whole polynomial P_L into the analysis lowpass filter $\widetilde{H}(\omega)$. This factor also destroys the regularity of the dual functions. The primal scaling function derived from the synthesis filter has $N-2$ continuous derivatives, and so does the wavelet. As seen in Figure (8.5) - (8.7), the dual scaling function and wavelet have much less regularity for $\widetilde{N} = N$, which is due to the factor $P_L(\sin^2\omega/2)$ present in $\widetilde{H}(\omega)$.

Other Biorthogonal Wavelets

Here we will describe other biorthogonal wavelets, based on the Daubechies product filter.

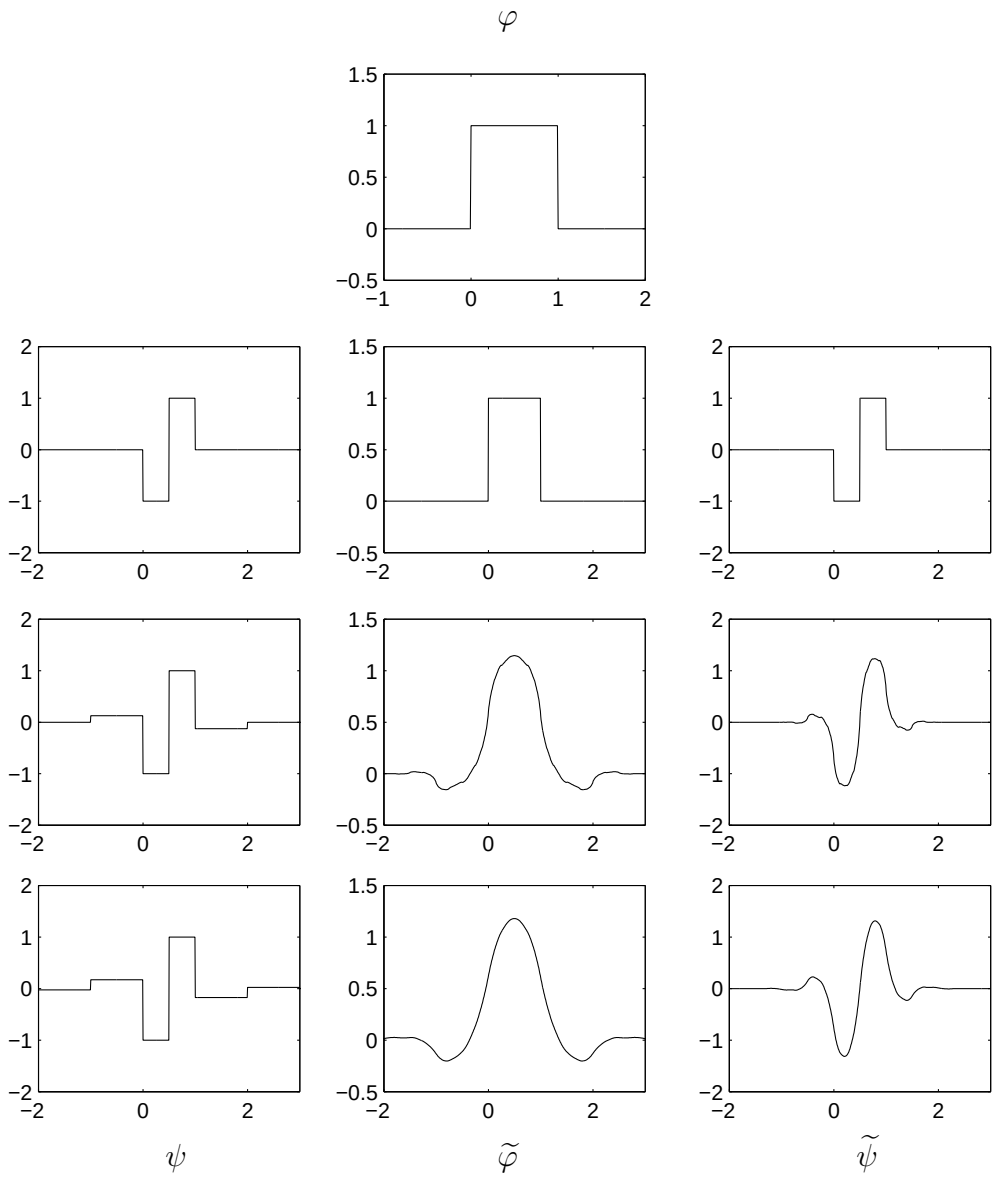

Figure 8.5: Biorthogonal spline wavelets with $N = 1$ and $\widetilde{N} = 1, 2$, and 3.

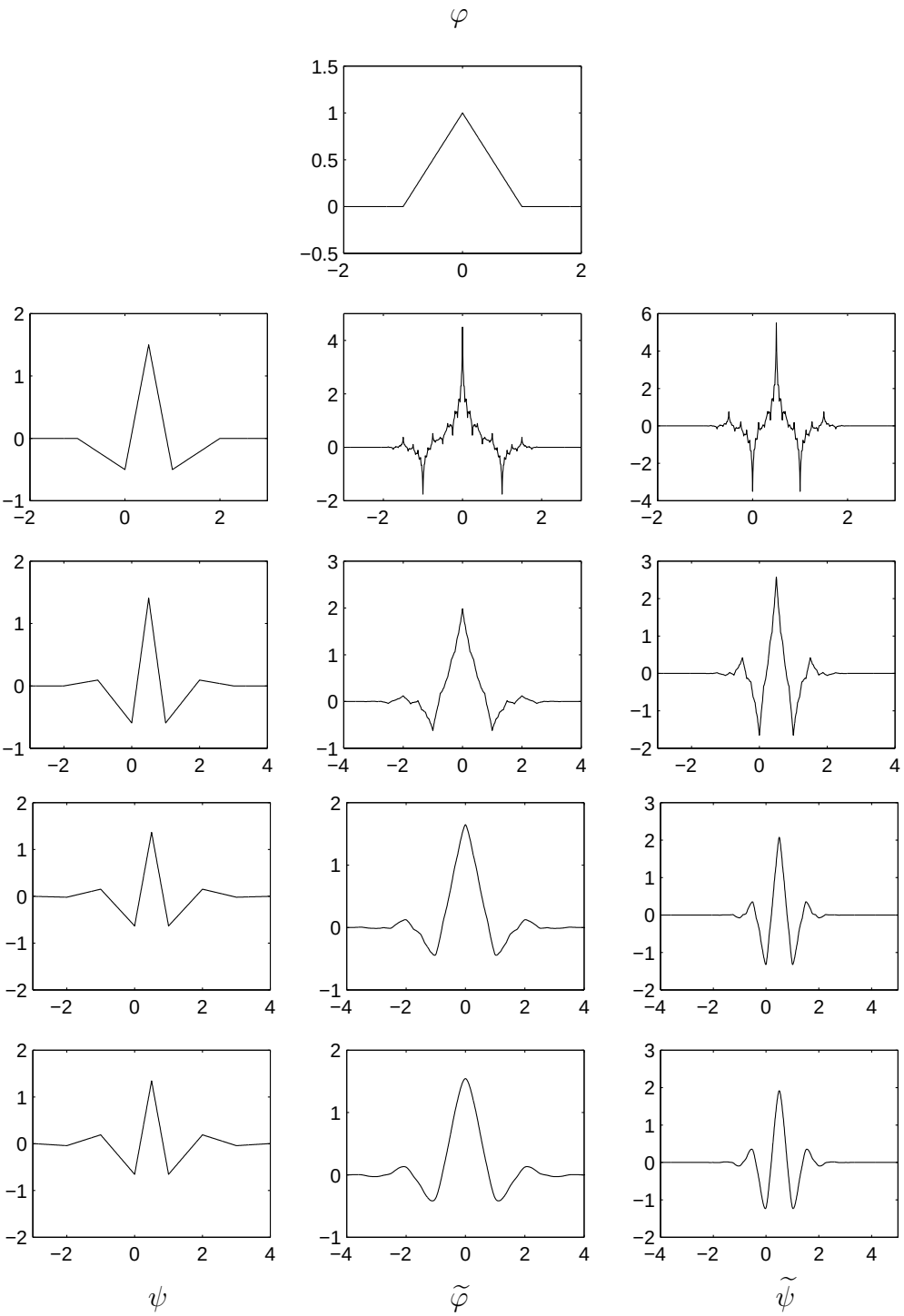

Figure 8.6: Biorthogonal spline wavelets with $N = 2$ and $\widetilde{N} = 2, 4, 6$, and 8.

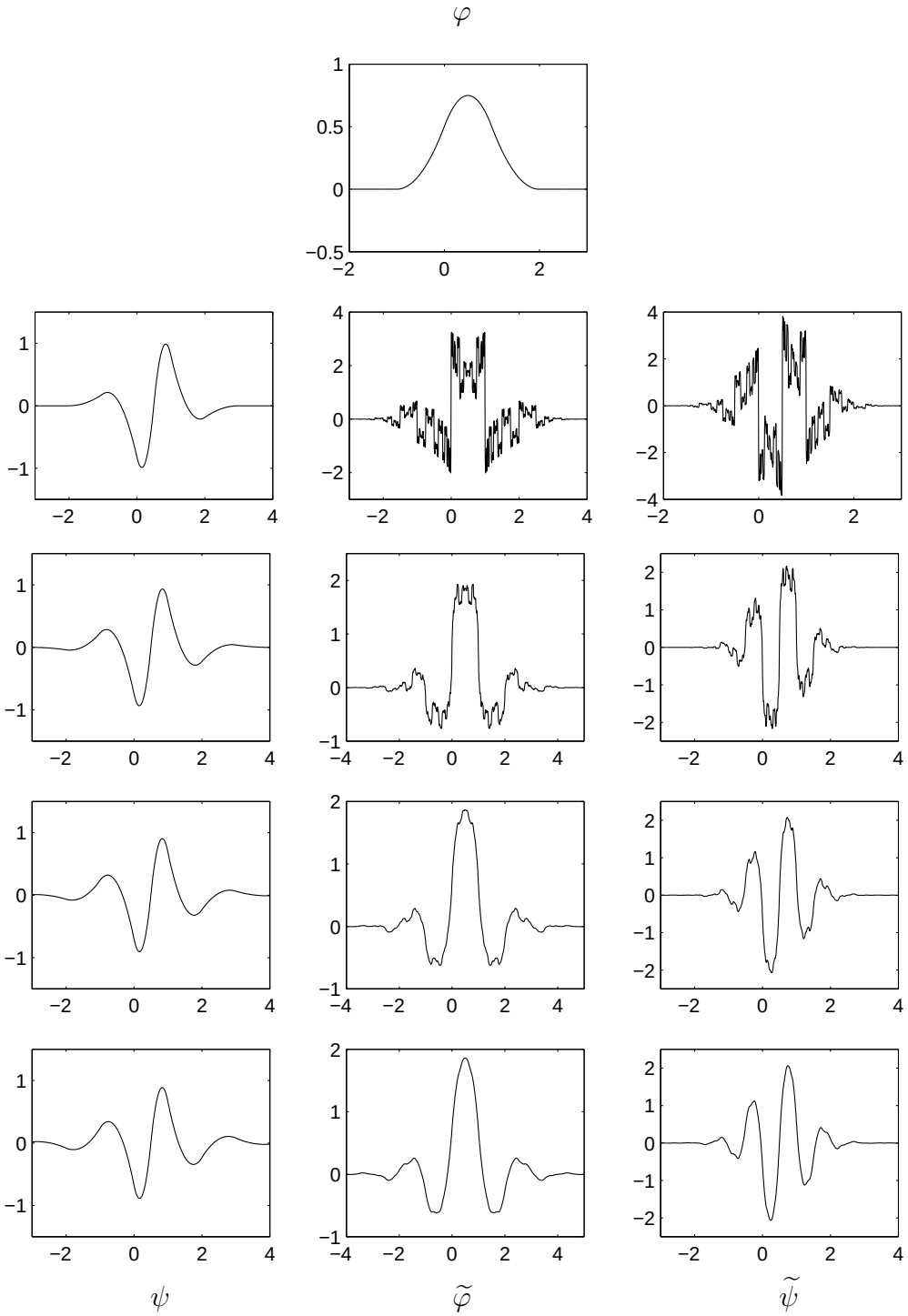

Figure 8.7: Biorthogonal spline wavelets with $N = 3$ and $\widetilde{N} = 3, 5, 7$, and 9.

To make the filters have more similar lengths, we again invoke the spectral factorization method to factorize $P_L(z)$ into $Q(z)$ and $\widetilde{Q}(z)$. To maintain symmetry, we always put zeros z_i and z_i^{-1} together. Also, z_i and $\overline{z_i}$ must always come together in order to have real-valued filter coefficients. This limits the number of possible factorizations, but, for fixed N and $\widetilde{N}$, there are still several possibilities. A disadvantage with these wavelets is that the filter coefficients no longer are dyadic rational, or even rational.

We have plotted scaling functions and wavelets for some for some of the most common filters in Figures 8.8 - 8.9.

The first filter pair is the biorthogonal filters used in the FBI fingerprint standard. They are constructed using a particular factorization of the Daubechies-8 product filter. Both the primal and the dual wavelet have 4 vanishing moments. The filters have lengths 7 and 9, and this is the reason for the notation 9/7 (the first number being the length of the dual lowpass filter).

The second 6/10 pair is obtained by moving one vanishing moment (a zero at $z = -1$) from the primal to the dual wavelet, and also interchanging some other zeros. The primal scaling function and wavelet become somewhat smoother. This filter pair is also very good for image compression.

The two last 9/11 pairs are based on two different factorizations of the Daubechies-10 product filter.

Almost Orthogonal Systems

Another construction comes from the observation that Coiflets are very close to being symmetric. This suggests that it might be possible to construct symmetric, biorthogonal wavelets that are close to the Coiflets, and thus close to being orthogonal. One possible way is to choose, roughly speaking, $H(\omega)$ as a rational approximation of the Coiflet low-pass filter. The dual filter $\widetilde{H}$ is then found by solving linear equations. This gives us a family of biorthogonal, symmetric, almost orthogonal wavelets with rational coefficients.

8.4 Wavelets without Compact Support

We will end this chapter by briefly mentioning some other wavelet bases, where not all of the wavelets and scaling functions have compact support. They are perhaps not so interesting in applications, but be include them

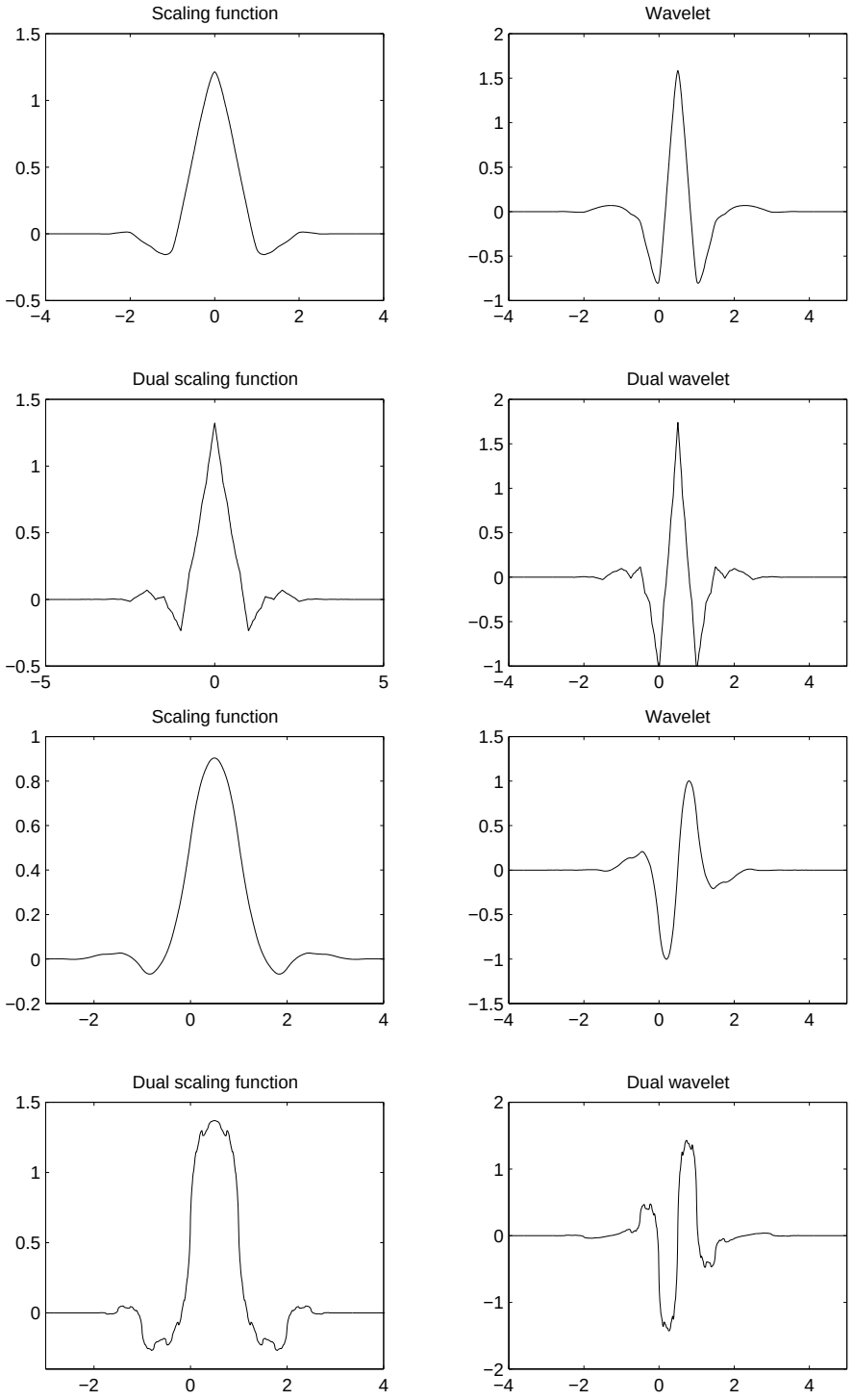

Figure 8.8: The 6/10 and the FBI 9/7 biorthogonal filter pair.

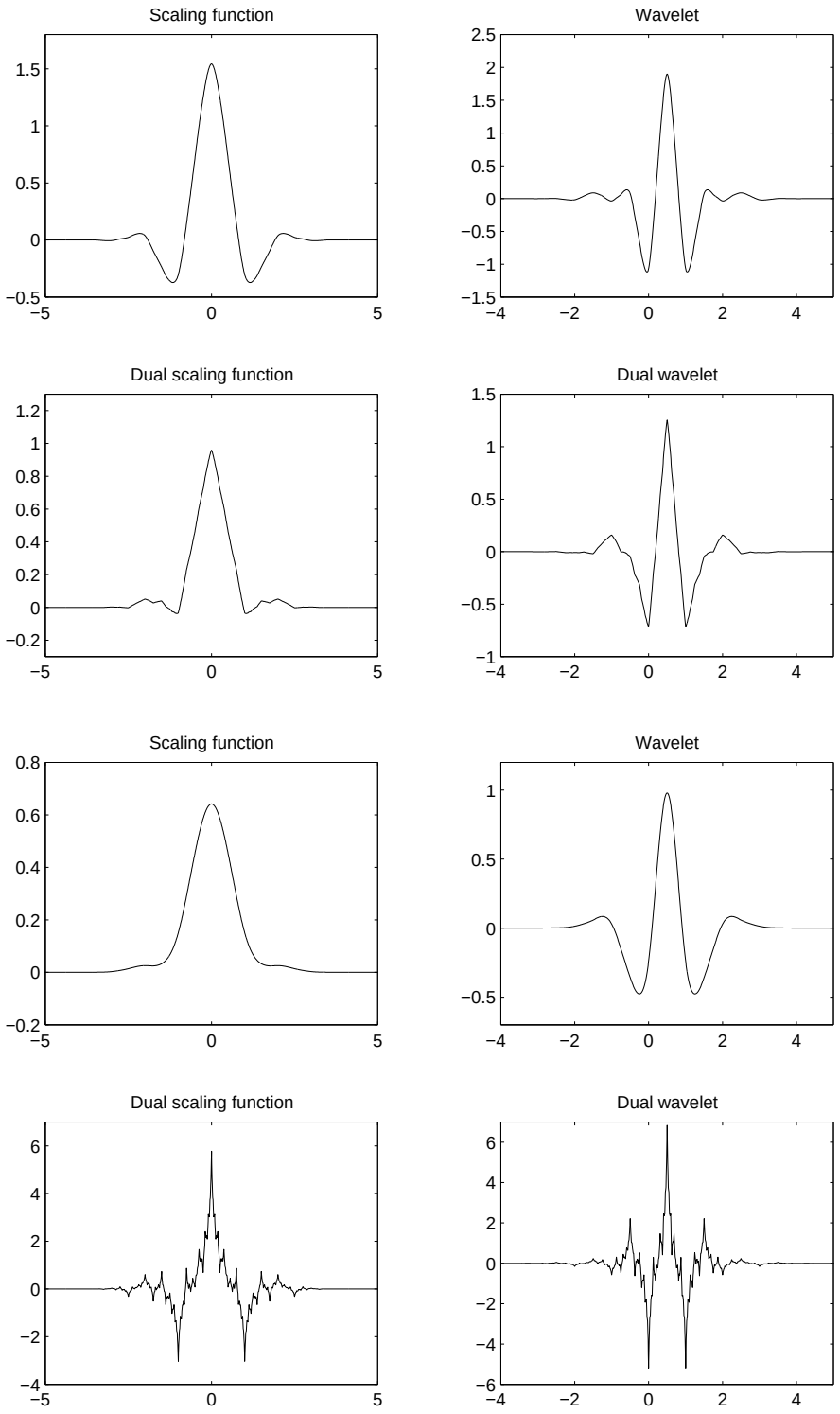

Figure 8.9: Scaling functions and wavelets for two 9/11 biorthogonal filter pair.

anayway, both for historical reasons, and because of their interesting theoretical properties.

Meyer Wavelets

The Meyer wavelets are explicitly described: the Fourier transform of the wavelet is defined as

$$\widehat{\psi}(\omega) = \begin{cases} \frac{1}{2\pi} e^{-i\omega/2} \sin\left(\frac{\pi}{2}\nu(3\,|\omega|-1)\right) & \text{if } \frac{2\pi}{3} < |\omega| \le \frac{4\pi}{3}, \\ \frac{1}{2\pi} e^{-i\omega/2} \cos\left(\frac{\pi}{2}\nu(\frac{3}{2}\,|\omega|-1)\right) & \text{if } \frac{4\pi}{3} < |\omega| \le \frac{8\pi}{3} \\ 0, & \text{otherwise.} \end{cases}$$

Here, ν is any smooth function such that

$$\nu(x) = \begin{cases} 0 & \text{if } x \le 0, \\ 1 & \text{if } x \ge 1, \end{cases}$$

and $\nu(x) + \nu(1-x) = 1$. These conditions on ν are enough to ensure that $\{\psi_{j,k}\}$ is an orthogonal basis. Using MRA, a proof of this is possible by identifying the scaling function

$$\widehat{\varphi}(\omega) = \begin{cases} \frac{1}{2\pi} & \text{if } |\omega| \le \frac{2\pi}{3}, \\ \frac{1}{2\pi} \cos\left(\frac{\pi}{2}\nu(3\,|\omega|-1)\right) & \text{if } \frac{2\pi}{3} < |\omega| \le \frac{4\pi}{3}, \\ 0 & \text{otherwise.} \end{cases}$$

The Meyer wavelets have a number of interesting properties. Since $\widehat{\psi}$ is zero in an interval around the origin, we have $\widehat{\psi}^{(n)}(0) = 0$ for all $n \in \mathbb{Z}$. Therefore the Meyer wavelets have an infinite number of vanishing moments. Both wavelets and scaling functions are symmetric. They are both infinitely differentiable (C^∞), since they have compact support in the frequency domain. They cannot have compact support, though, but decay fast. The rate of decay is determined by the smoothness of ν.

Battle-Lemarié Wavelets

The construction of the Battle-Lemarié orthogonal family starts from the B-spline scaling functions. These are not orthogonal to their integer translates, since they are positive and overlapping. Remember the orthogonality condition for the scaling function from Section 4.4.

$$\sum_l |\widehat{\varphi}(\omega + 2\pi l)|^2 = 1. \tag{8.5}$$

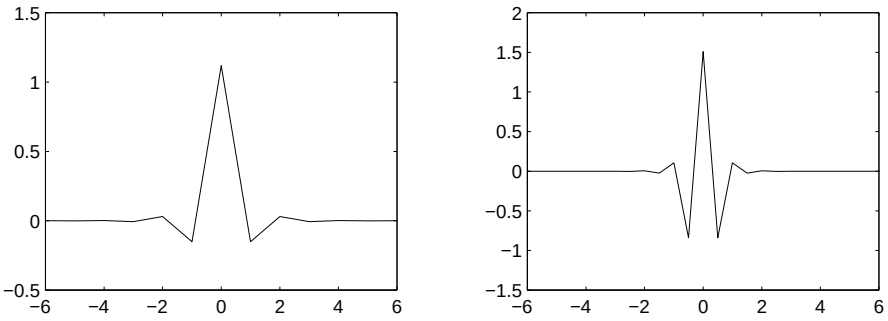

Figure 8.10: The piecewise linear Battle-Lemarié scaling function and wavelet.

Instead, the B-spline scaling functions satisfy the inequalities

$$A \leq \sum_l |\widehat{\varphi}(\omega + 2\pi l)|^2 \leq B,$$

for all ω, where A and B are some positive constants. This is actually the condition on $\{\varphi(t-k)\}$ to be a Riesz basis, transformed to the Fourier domain. Now it is possible to define the Battle-Lemarié scaling function

$$\widehat{\varphi}^{\#}(\omega) = \frac{\widehat{\varphi}(\omega)}{\sqrt{\sum_l |\widehat{\varphi}(\omega + 2\pi l)|^2}}.$$

This scaling function satisfies (8.5), and thus it generates an orthogonal multiresolution analysis. The wavelet is defined by the alternating flip construction.

One can show that this 'orthogonalized' spline scaling function spans the same V_j spaces as the original spline scaling function. However, it will no longer have compact support, and this also holds for the wavelet. They both have exponential decay, though, and are symmetrical. In Figure 8.10, we have plotted the Battle-Lemarié scaling function and wavelet correponding to piecewise linear splines. Note that they are both piecewise linear, and that the decay is indeed very fast.

Chui Semiorthogonal Spline Wavelets

Semiorthogonal bases is a special case of biorthogonal bases, where we require $V_j = \widetilde{V}_j$ and $W_j = \widetilde{W}_j$. We still have two scaling functions φ, $\widetilde{\varphi}$, and wavelets ψ, $\widetilde{\psi}$, but they must now generate the same V_j and W_j spaces. For orthogonality, we need the extra condition $\varphi = \widetilde{\varphi}$ and $\psi = \widetilde{\psi}$. The usual

biorthogonality conditions immediately implies that $V_j \perp W_j$ for each j. At each scale, all scaling functions must then be orthogonal to all wavelets. Further, the wavelets $\psi_{0,k}$ belong to all approximation spaces V_j for $j > 0$. They must therefore be orthogonal to all wavelets with $j > 0$. We can thus conclude that wavelets at different scales are always orthogonal. At a fixed scale j, the wavelets $\psi_{j,k}$ are not orthogonal among themselves, but rather to the dual wavelets $\widetilde{\psi}_{j,k}$,

$$\langle \psi_{j,k}, \widetilde{\psi}_{j,l} \rangle = \delta_{k,l}.$$

The semiorthogonal spline wavelets of Chui & Wang use B-splines of order N as scaling functions. The wavelets become spline functions with support width $[0, 2N-1]$. The dual scaling functions and wavelets are also splines but without compact support. They decay very fast, though. All scaling functions and wavelets are symmetric, and all filter coefficients are rational. We also have analytic expressions for all scaling functions and wavelets.

8.5 Notes

One of the first families of wavelets to appear was the Meyer wavelets, which were constructed by Yves Meyer in 1985. This was before the connection between wavelets and filter banks was discovered, and thus the notion of multiresolution analysis was not yet available. Instead, Meyer defined his wavelets through explicit constructions in the Fourier domain.

Ingrid Daubechies constructed her family of orthogonal wavelets in 1988. It was the first construction within the framework of MRA and filter banks. Shortly thereafter followed the construction of Symmlets, Coiflets (which were asked for by Ronald Coifman in 1989, therefore the name), and later also various biorthogonal bases.

For more details about the construction of different wavelet bases, we refer to Daubechies' book [11], where also tables with filter coefficients can be found. Filter coefficients are also available in WaveLab.

Chapter 9

Adaptive Bases

In this chapter, we describe two constructions allowing an adaption of the analysis to the signal at hand. These are Wavelet Packets and Local Trigonometric Bases. In a sense, they are dual to each other: wavelet packets provide flexibility in the frequency band decomposition, whereas local trigonometric bases, which are closely related to windowed Fourier analysis, allow an adaption of the time interval decomposition.

Wavelet packets and local trigonometric bases are special cases of the more general concept of time-frequency decompositions, which we discuss briefly at first.

9.1 Time-Frequency Decompositions

A time-frequency decomposition of a signal f is an expansion

$$f(t) = \sum_k c_k \, b_k(t),$$

where each basis function b_k is well localized in both time and frequency. By the latter we mean that both $f(t)$ and $\widehat{f}(\omega)$ decay fast as $|t|, |\omega| \to \infty$. Such basis functions are sometimes referred to as *time-frequency atoms*. The wavelet decomposition

$$f(t) = \sum_{j,k} \langle f, \widetilde{\psi}_{j,k} \rangle \, \psi_{j,k}(t)$$

is an example of a time-frequency decomposition. We assume that the mother wavelet ψ has its energy mainly in the time interval $(0, 1)$ and frequency band $(\pi, 2\pi)$. By this we mean that

$$\int_0^1 |\psi(t)|^2 \, dt \quad \text{and} \quad \int_\pi^{2\pi} |\widehat{\psi}(\omega)|^2 \, d\omega$$

both are close to the total energy of the signal. (*Cf.* the inequality 1.1 in Chapter 1, which puts a limit to simultaneous localization in time and in frequency.) By scaling, the basis functions $\psi_{j,k}(t)$ then have their energy essentially concentrated to the time interval $(2^{-j}k, 2^{-j}(k+1))$ and the frequency interval $(2^j\pi, 2^{j+1}\pi)$. We associate $\psi_{j,k}$ with the rectangle $(2^{-j}k, 2^{-j}(k+1)) \times (2^j\pi, 2^{j+1}\pi)$ in the *time-frequency plane* (see Figure 9.1). These rectangles are sometimes referred to as *Heisenberg boxes*. The Heisen-

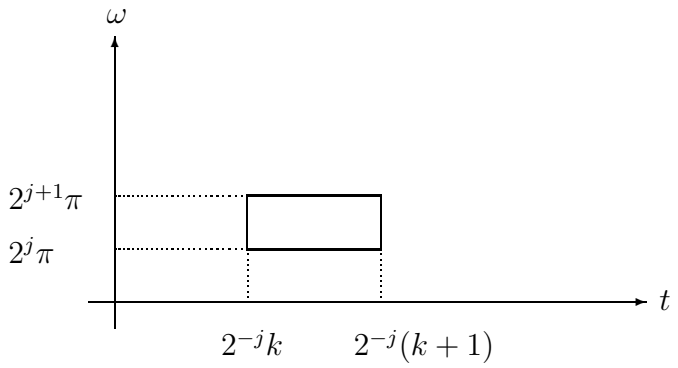

Figure 9.1: The Heisenberg box for $\psi_{j,k}$.

berg boxes for the wavelets $\psi_{j,k}$ gives a tiling of the time-frequency plane as shown in Figure (9.2). The lowest rectangle corresponds to the scaling function $\varphi_{j_0,0}$ at the coarsest scale. If we have an orthonormal wavelet basis,

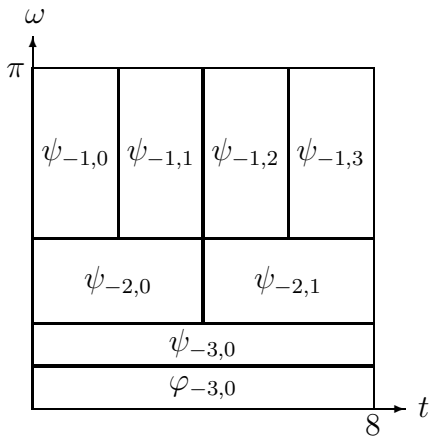

Figure 9.2: Tiling of the time-frequency plane for a wavelet basis.

the energy of a signal can be expressed in terms of the wavelet coefficients

$w_{j,k} = \langle f, \psi_{j,k} \rangle$ as

$$\|f\|^2 = \sum_{j,k} |w_{j,k}|^2 .$$

Hence, the squared modulus of wavelet coefficients $|w_{j,k}|^2$ can be interpreted as an energy distribution in the time-frequency plane. Each Heisenberg box contains a certain amount $|w_{j,k}|^2$ of the signal's total energy.

Two extreme cases of time-frequency decompositions are shown in Figure 9.3. The first one has perfect time resolution but no frequency resolution. The corresponding basis is associated with the sample values, so the basis functions could be, *e.g.*, sinc functions. More generally, they could be scaling functions at the finest scale. The other time-frequency decomposition is the discrete Fourier transform, which has no time resolution but perfect frequency resolution.

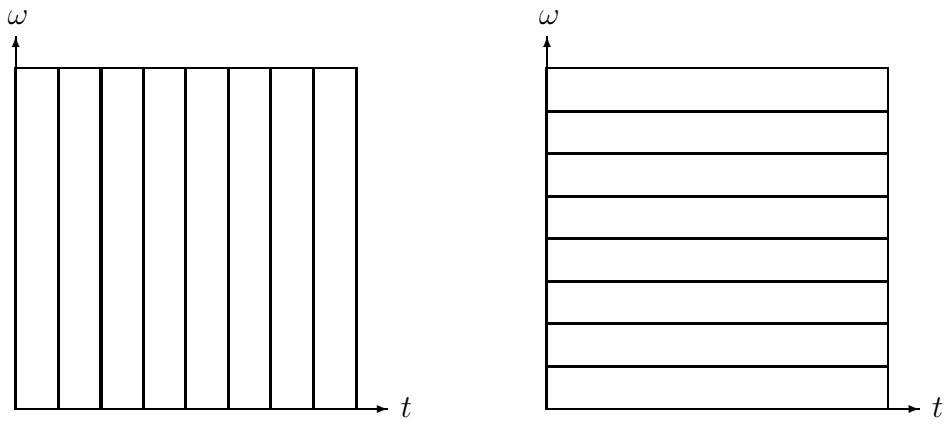

Figure 9.3: Time-frequency plane for the basis associated with sampling and the Fourier basis.

Another time-frequency decomposition is the windowed Fourier transform, where the signal is divided into segments $(kT, (k+1)T)$ of length T, and a discrete Fourier transform is applied to each segment. The decomposition into segments can be thought of as multiplying the signal with *window functions* that equal 1 on $(kT, (k+1)T)$, and 0 otherwise. These windows are called *ideal* windows. To make the transition between segments smoother, one can use smoother versions of the ideal window. One then multiplies the signal with $w(t - kT)$, where w is a smooth approximation of the ideal window for $(0, T)$. The tiling of the time-frequency plane for the windowed Fourier transform is shown in Figure (9.4) for two different *window sizes* T.

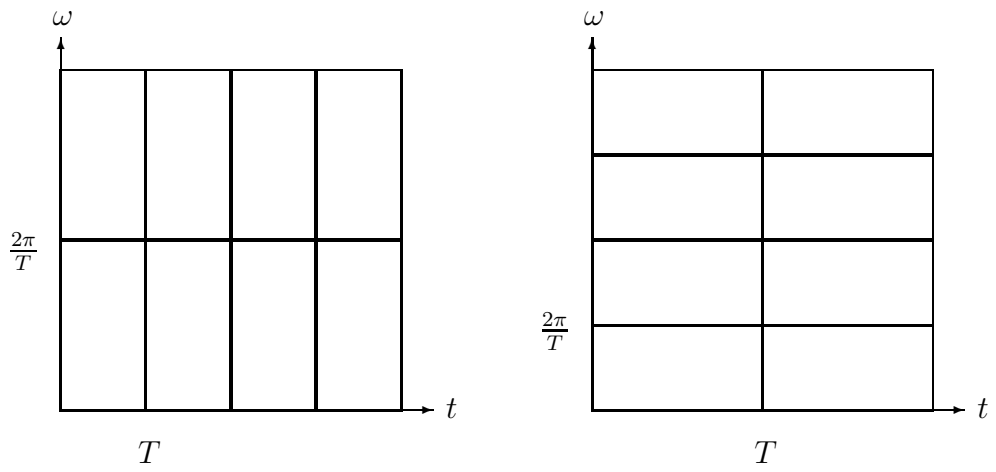

Figure 9.4: Tiling of the time-frequency plane for windowed Fourier transform with two window sizes

How to choose a proper window size is the major problem with the windowed Fourier transform. If the window size is T, each segment will be analyzed at frequencies, which are integer multiples of $2\pi/T$. Choosing a narrow window will give us good time resolution, but bad frequency resolution, and *vice versa*.

One solution to this problem would be to adapt the window size to the frequency, and use narrow windows at high frequencies and wider windows for low frequencies. This is basically done by the wavelet transform, even though there is no explicit window involved. For high frequencies, the wavelets are well localized in time, and the Heisenberg boxes are narrow and high. At lower frequencies, the wavelets are more spread out, and we get boxes with large width and small height. This is useful, since many high frequency phenomena are short-lived, *e.g.*, edges and transients, while lower frequency components usually have a longer duration in time.

Exercises 9.1

9.1. Sketch for yourself how the decomposition of the time-frequency plane changes as we go through the filtering steps in the Forward Wavelet Transform, starting at the finest scaling coefficients.

9.2 Wavelet Packets

As mentioned above, the wavelet transform is suitable for signals with short-lived high-frequency and long-lived low-frequency components. However, some signals having high-frequency components with a long duration.[1] This is the case for images with *textures*, *i.e.*, parts of the image having a special structure. An example of textures is given by the fingerprint image in Figure 1.12 in Chapter 1. Such parts of the image often have a well defined frequency content at high frequencies. Many fine-scale wavelets will therefore be needed to approximate this region of the image. In this case, it could be preferable to have more spread out fine-scale wavelets.

The Wavelet Packet Transforms

The wavelet transform decomposes signals into the frequency bands as in Figure 4.8. We would like to be able to split the signal into more general frequency bands. One way to accomplish this flexibility is to allow also the wavelet coefficients to be split with the analysis filter bank. Starting with the scaling coefficients s_J, we compute scaling coefficients at a coarser scale, s_{J-1}, and wavelet coefficients w_{J-1}, using the analysis filter bank. Both can be further split with the analysis filters. At the next level we can have up to four sequences of coefficients, and they are all allowed to be further split. This gives us a tree-structure of coefficients at different levels as depicted in Figure 9.5. This tree is called the *wavelet packet tree*. At each node in the wavelet packet tree, we have a set of coefficients that we can choose to further split or not. Every sequence of such choices gives a particular *wavelet packet transform*. Each such transform corresponds to a subtree of the wavelet packet tree with all nodes having either two children or none.

Wavelet Packet Basis Functions

Every wavelet packet transform gives us the coefficients in a particular *wavelet packet basis*. The scaling coefficients $(s_{J-1,k})$ and wavelet coefficients $(w_{J-1,k})$ are coefficients in the bases $\{\varphi_{J-1,k}\}$ and $\{\psi_{J-1,k}\}$ of V_{J-1} and W_{J-1}.

When we split the space V_{J-1} further, we get coefficients $(s_{J-2,k})$ and $(w_{J-2,k})$ for the basis functions $\varphi_{J-2,k}$ and $\psi_{J-2,k}$, that together span V_{J-1}. When we split W_{J-1} with the analysis filters, we get coefficients for certain

[1] With space variables, the terms are large wave-number components and long extension, respectively. We will, however, follow established practice and use the term frequency instead of wave-number.

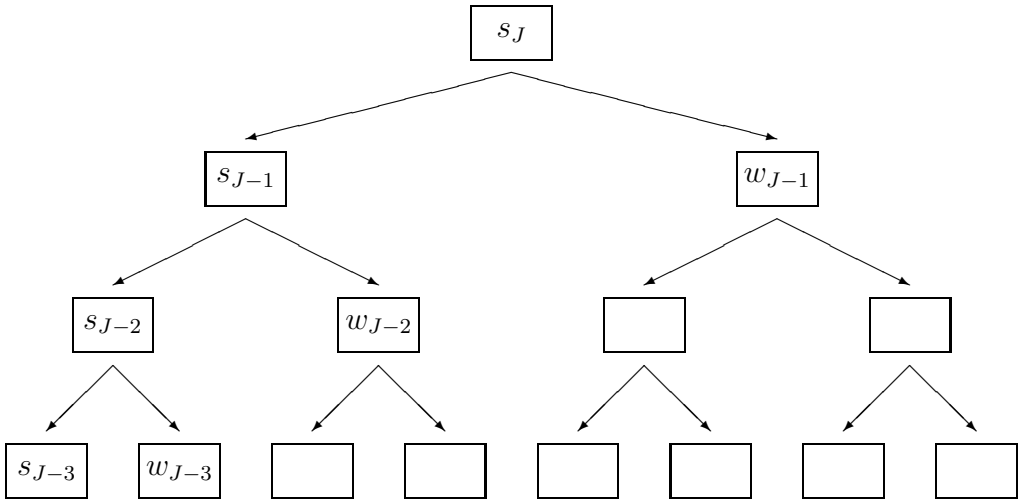

Figure 9.5: The wavelet packet tree.

wavelet packet basis functions. Before we define these basis functions, we introduce a convenient notation:

$$m_0(\omega) = H(\omega) \quad \text{and} \quad m_1(\omega) = G(\omega).$$

We now define wavelet packets at level $J-2$ as

$$\psi^{(1,0)}_{J-2,k}(t) = \psi^{(1,0)}_{J-2}(t - 2^{J-2}k) \quad \text{and} \quad \psi^{(1,1)}_{J-2,k}(t) = \psi^{(1,1)}_{J-2}(t - 2^{J-2}k),$$

where

$$\widehat{\psi^{(1,0)}_{J-2}}(\omega) = \frac{1}{\sqrt{2}} m_0(2^{1-J}\omega)\widehat{\psi}_{J-1,0}(\omega),$$

$$\widehat{\psi^{(1,1)}_{J-2}}(\omega) = \frac{1}{\sqrt{2}} m_1(2^{1-J}\omega)\widehat{\psi}_{J-1,0}(\omega). \tag{9.1}$$

The wavelet packets $\psi^{(1,0)}_{J-2,k}$ and $\psi^{(1,1)}_{J-2,k}$ together constitute a Riesz basis for W_{J-1}. This is a consequence of the *splitting trick*:

Theorem 9.1. Assume that $\{\chi(t-k)\}$ is a Riesz basis for a subspace V. Then the functions

$$\chi^0_k(t) = \sqrt{2}\chi^0(t-2k) \quad \text{and} \quad \chi^1_k(t) = \sqrt{2}\chi^1(t-2k)$$

together consitute a Riesz basis for V, where

$$\widehat{\chi^0}(\omega) = m_0(\omega)\widehat{\chi}(\omega) \quad \text{and} \quad \widehat{\chi^1}(\omega) = m_1(\omega)\widehat{\chi}(\omega).$$

□

We now define general wavelet packets as

$$\psi^e_{j,k}(t) = \psi^e_j(t - 2^{-j}k),$$

where $e = (e_1, \ldots, e_L)$ is a sequence of 0's and 1's, and

$$\widehat{\psi^e_j}(\omega) = 2^{-j/2}\widehat{\varphi}(2^{-j-L}\omega) \prod_{i=0}^{L-1} m_{e_i}(2^{i-j-L}\omega).$$

In other words, the wavelet packet ψ^e_j is obtained by applying a sequence of low- and highpass filterings according to the e_i's, starting with the scaling function $\varphi_{j+L,0}$. The space spanned by the wavelet packets $\psi^e_{j,k}$ for fixed j and e is denoted W^e_j. In Figure 9.6, we show the ideal frequency bands associated with these spaces.

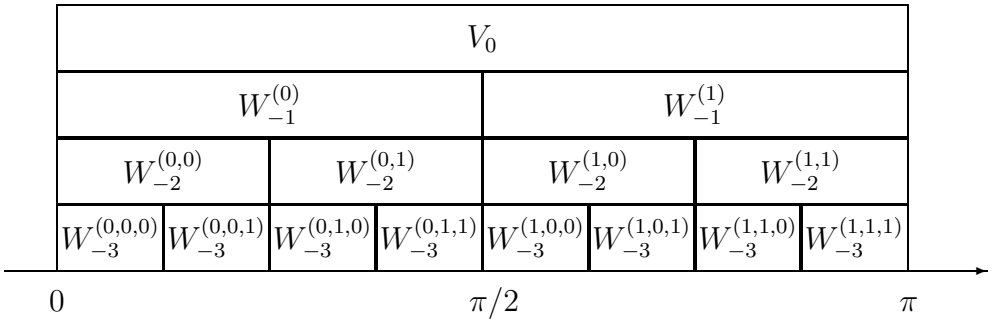

Figure 9.6: The ideal frequency bands for the wavelet packet spaces

In Figure (9.7) we show the tiling of the ideal time-frequency plane for two particular wavelet packet bases together with the corresponding subtrees of the wavelet packet tree. Note that we get two high-frequency basis functions with long duration in time with the first basis (the left figure).

Wavelet Packets in Two Dimensions

There are also wavelet packets for two-dimensional wavelet transforms. For separable wavelets, this amounts to allowing the detail spaces $\mathbf{W}^H$, $\mathbf{W}^V$, and $\mathbf{W}^D$ to be further split with the separable two-dimensional filters. This gives a more complicated wavelet packet tree, where each node has four children. In Figure 9.8, we show the frequency plane decomposition of separable wavelet packets after one and two splittings. The superscripts $0, 1, 2, 3$ denote filtering with $\mathbf{H}$, $\mathbf{G}_H$, $\mathbf{G}_V$, and $\mathbf{G}_D$ respectively.

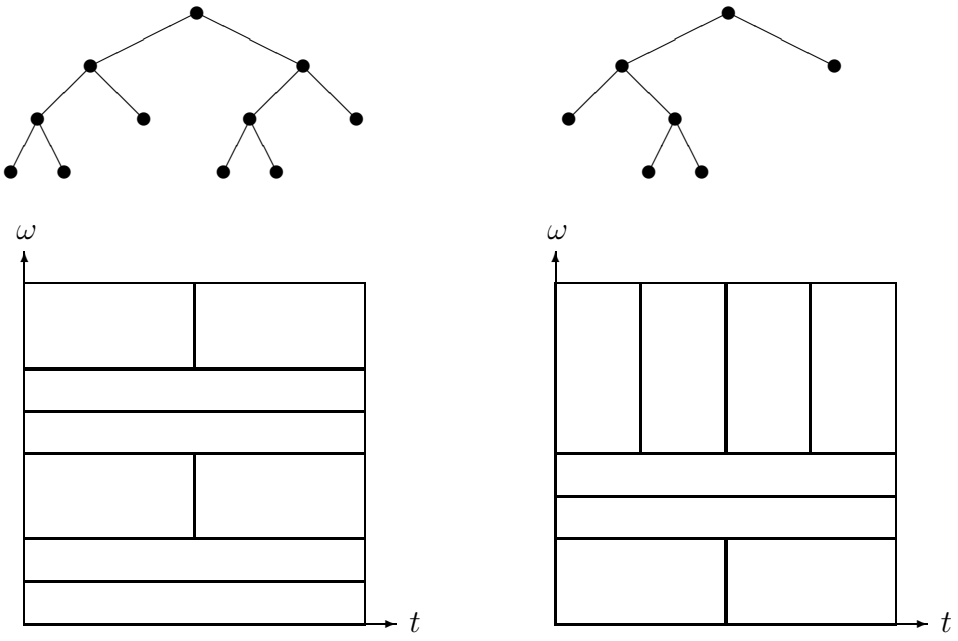

Figure 9.7: The tiling of the ideal time-frequency plane for two different wavelet packet bases

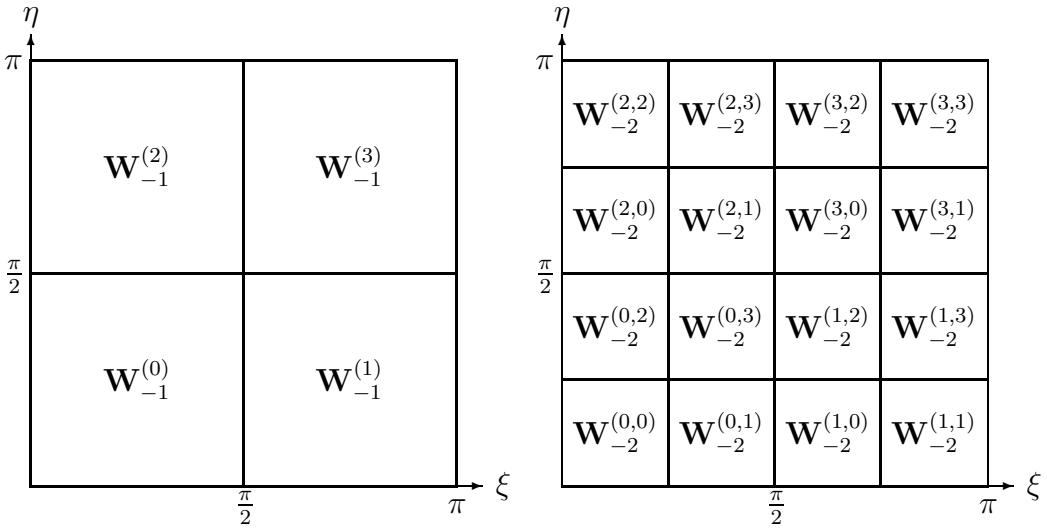

Figure 9.8: The ideal decomposition of the frequency plane for separable two-dimensional wavelet packet bases

Exercises 9.2

9.2. Sketch for yourself step by step how the ideal decomposition of the time-frequency plane changes when going from the representaion in the V_J space to the wavelet packets representations in Figure 9.7.

9.3 Entropy and Best Basis Selection

A natural question at this point is which of the many wavelet packets transforms to choose. If we have some *a priori* information about the signals under consideration, this can sometimes be used to select a proper wavelet packet basis. Such is the case with the FBI fingerprints images, where the ridges in the fingerprints are known to be repeated with certain frequencies. If no such information is available, we have to use the transform that is optimal for the given signal, with respect to some chosen criterion.

The Best Basis Algorithm

Suppose that we have a *cost function* Λ that somehow measures the performance of different transforms, that is, if c and $\widetilde{c}$ are the coefficients in two different bases, the first one is preferred if $\Lambda(c) < \Lambda(\widetilde{c})$. We further assume that Λ is *additive*, which means that if we merge two sequences c_1 and c_2 into one sequence c, which we write $c = [c_1\, c_2]$, we have $\Lambda(c) = \Lambda(c_1) + \Lambda(c_2)$. We will describe how to find the wavelet packet basis whose coefficients, for a given signal, minimize Λ. At first, this seems to be an overwhelming task, since the number of wavelet packet bases is proportional to 2^N, if we start with N scaling coefficients $s_{J,k}$. However, there is a depth-first search algorithm that finds the optimal basis in $O(N \log N)$ operations. This algorithm is called the *Best Basis Algorithm*.

To understand how the algorithm works, let us look at the splitting of the scaling coefficients s_J into s_{J-1} and w_{J-1}. We recursively assume that we know the optimal sequence of choices for splitting s_{J-1} and w_{J-1}, and denote the corresponding optimal wavelet packets coefficients s^{opt}_{J-1} and w^{opt}_{J-1}. The additivity now implies that the best wavelet packet coefficients we can get, if we decide to split s_J, is $c^{opt}_{J-1} = [s^{opt}_{J-1} w^{opt}_{J-1}]$, since for any other set of wavelet packet coefficients $c^{oth}_{J-1} = [s^{oth}_{J-1} w^{oth}_{J-1}]$, we have

$$\Lambda(c^{oth}_{J-1}) = \Lambda(s^{oth}_{J-1}) + \Lambda(w^{oth}_{J-1}) \geq \Lambda(s^{opt}_{J-1}) + \Lambda(w^{opt}_{J-1}) = \Lambda(c^{opt}_{J-1}).$$

Thus we should split s_J exactly when

$$\Lambda(s_J) > \Lambda(c^{opt}_{J-1}) = \Lambda(s^{opt}_{J-1}) + \Lambda(w^{opt}_{J-1}).$$

Entropy and Other Cost Functions

Entropy

In many applications, the goal is to extract the relevant information in a signal, using as few coefficients as possible. Thus, we want a few coefficients to be large and the remaining ones to be small. The *entropy* is a common way to measure this property. The entropy of the coefficients c is defined as

$$\mathcal{H}(c) := -\sum_k p_k \log p_k,$$

where

$$p_k = \frac{|c_k|^2}{\|c\|^2} \quad \text{and} \quad 0 \log 0 := 0.$$

A well-known fact about the entropy measure, which is outlined in Exercise 9.4, is the following:

Theorem 9.2. If c is a finite sequence, of length K say, then

$$0 \le \mathcal{H}(c) \le \log K.$$

Further, the minimum value is attained only when all but one c_k are 0, and the maximum is attained only when all $|c_k|$ equal $1/\sqrt{K}$. □

Of course, the conclusion also holds for sequences c with at most K non-zero coefficients. The number $d(c) = e^{\mathcal{H}(c)}$ is known in information theory as the *theoretical dimension* of c. It can be proved that the number of coefficents that are needed to approximate c with error less than ϵ is proportional to $d(c)/\epsilon$.

However, we may not use the entropy measure as a cost function directly, since it is not additive. But if we define the additive cost function

$$\Lambda(c) := -\sum_k |c_k|^2 \log |c_k|^2 ,$$

we have that

$$\mathcal{H}(c) = 2 \log \|c\| + \frac{1}{\|c\|^2} \Lambda(c). \tag{9.2}$$

Thus we see that minimizing Λ is equivalent to minimizing $\mathcal{H}$ for coefficient sequences with fixed norm. Therefore, if we want to use the best basis algorithm to minimize the entropy, we must ensure that the norm of s_J equals the norm of $[s_{J-1}\, w_{J-1}]$. In other words, we must use orthogonal filters.

Other cost functions

An alternative cost function is the ℓ^1-norm of c:

$$\Lambda(c) := \sum_k |c_k| \,.$$

For coefficient sequences with fixed $\ell^2(\mathbb{Z})$-norm, Λ is also minimized when all coefficients are 0 except one and maximized when all $|c_k|$ are equal. To get relevant comparisons between s_J and $[s_{J-1}\, w_{J-1}]$, we again need the filters to be orthogonal.

Finally, we mention two application-dependent cost functions. When using wavelet packets for reducing noise in signals, one would like to choose the basis that gives the smallest error between the denoised signal and the real signal. As a cost function one can use an estimate of this prediction error. One example of this, the SURE cost function, will be discussed in Section 10.2. In classification applications, cost functions that measure the capability of separating classes are used. They will be discussed in more detail in Chapter 14.

Exercises 9.3

9.3. Verify that the entropy measure is not an additive cost function, but that Λ is. Then verify the identity (9.2).

9.4. Prove theorem 9.2. Hint: Use that the exponential function is strictly convex, which means that

$$e^{\sum_k \lambda_k x_k} \leq \sum_k \lambda_k e^{x_k},$$

for $\lambda_k \geq 0$ and $\sum_k \lambda_k = 1$, and equality occurs only when all x_k are equal. Apply this with $\lambda_k = p_k$ and $x_k = \log p_k$ to obtain $e^{\mathcal{H}(c)} \leq K$. The left inequality should be obvious, since all $0 \leq p_k \leq 1$, which implies $\log p_k \leq 0$. The only way $\mathcal{H}(c) = 0$ could happen is if all p_k are either 0 or 1. But since they sum up to 1, this implies that one p_k equals 1 and the remaining are 0.

9.5. Show that for $c \in \mathbb{R}^K$

$$\|c\| \leq \sum_{k=1}^{K} |c_k| \leq \sqrt{K}\, \|c\| \,,$$

where the maximum value is attained when all $|c_k|$ are equal, and the minimum is attained when all c_k's but one are 0. Hint: For the right hand inequality, write

$$\sum_{k=1}^{K} |c_k| = \sum_{k=1}^{K} 1 \cdot |c_k|$$

and use the Cauchy-Schwartz inequality. For the left hand inequality, you may assume that $\|c\| = 1$ (why?). Show that this implies $|c_k| \leq 1$ for each k. Then $|c_k|^2 \leq |c_k|$, and the inequality follows.

9.4 Local Trigonometric Bases

In the previous section, we have described how wavelet packets provide time-frequency decompositions with adaptivity in frequency. When using these packets, we split frequency bands into two parts whenever it is to our advantage. Local trigonometric bases can be seen as the dual of wavelet packets, where now the adaptivity comes in the time variable instead. Moreover, the local trigonometric bases are defined explicitly, while the definition of wavelet packets is algorithmic.

The Local Trigonometric Basis Functions

We start with a partition of the real line into intervals I_k, $k \in \mathbb{Z}$. To keep things simple, we choose $I_k = [k, k+1]$, but the construction works without serious complications for non-uniform partitions. We consider window functions $w_k(t)$ associated with the intervals I_k, having the following properties:

$$0 \leq w_k(t) \leq 1 \quad \text{for all } t, \tag{9.3a}$$

$$w_k(t) = 0 \quad \text{for } t \leq k - \frac{1}{2} \text{ and } t \geq k + \frac{3}{2}, \tag{9.3b}$$

$$w_k(k+t)^2 + w_k(k-t)^2 = 1 \quad \text{for } |t| \leq \frac{1}{2}, \tag{9.3c}$$

$$w_k(k+1+t)^2 + w_k(k+1-t)^2 = 1 \quad \text{for } |t| \leq \frac{1}{2}, \tag{9.3d}$$

$$w_k(k-t) = w_{k-1}(k+t) \quad \text{for all } t. \tag{9.3e}$$

These somewhat complicated conditions are needed to ensure that the local trigonometric basis functions, which we will define soon, give an ON basis of $L^2(\mathbb{R})$.

Setting $t = 1/2$ in (9.3c), we see that $w_k(k+1/2) = 1$, since, from (9.3b), we have $w_k(k-1/2) = 0$. Sometimes, the window function w_k is required to be identically 1 on $[k, k+1]$ and 0 outside, except in small intervals around $t = k$ and $t = k+1$. In this case, $w_k(t)$ is very close to the ideal window. Since we have a uniform partition, we can get the window functions from one window function $w(t)$ by translation, $w_k(t) = w(t-k)$.

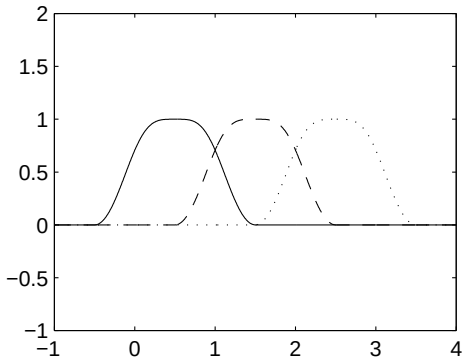

Figure 9.9: Window functions.

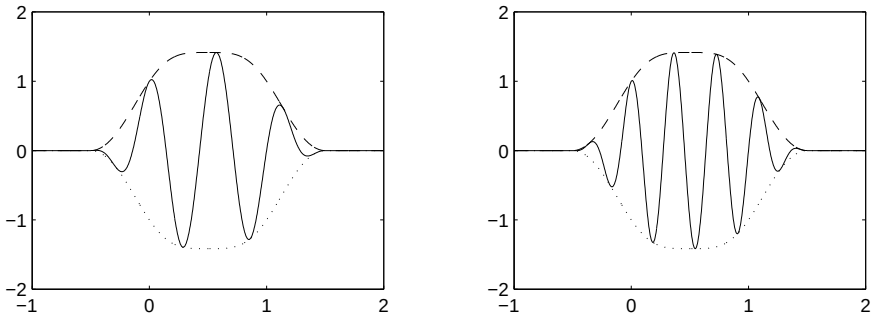

Figure 9.10: Two basis functions $b_{3,0}$ and $b_{5,0}$

The Local Trigonometric Basis Functions for the partitioning I_k are constructed by filling the windows $w_k(t)$ with cosine oscillations at frequencies $\pi(n+\frac{1}{2})$:

$$b_{n,k}(t) = \sqrt{2}\, w_k(t) \cos\left[\pi(n+\frac{1}{2})(t-k)\right].$$

It can be shown that the functions $b_{n,k}$ constitute an orthonormal basis for

$L^2(\mathbb{R})$. To compute the coefficients

$$c_{n,k} = \langle f, b_{n,k} \rangle = \int_{-\infty}^{\infty} f(t) b_{n,k}(t)\, dt$$
$$= \sqrt{2} \int_{k-\frac{1}{2}}^{k+\frac{3}{2}} f(t) w_k(t) \cos\left[\pi(n+\frac{1}{2})(t-k)\right]\, dt,$$

we observe that $\cos\left[\pi(n+\frac{1}{2})(t-k)\right]$ is symmetric around the left interval endpoint $t = k$, and anti-symmetric around the right interval endpoint $t = k+1$. Therefore, the parts of $f(t)w_k(t)$ outside the interval $[k, k+1]$ can be folded back into the interval as shown in Figure 9.11. This produces a folded version $\widetilde{f}$ of f, and the coefficients $c_{n,k}$ are then given by

$$c_{n,k} = \sqrt{2} \int_{k}^{k+1} \widetilde{f}(t) \cos\left[\pi(n+\frac{1}{2})(t-k)\right]\, dt.$$

When computing the coefficients $c_{n,k}$, after the folding, we can use the fast

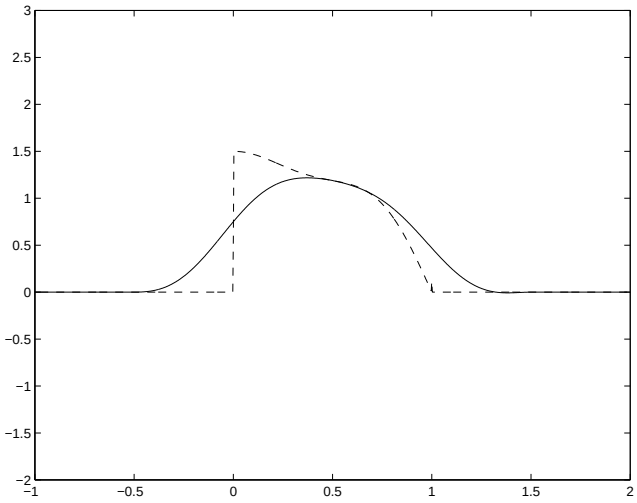

Figure 9.11: The windowed function $f(t)w(t)$ (solid line) and its folded version (dashed line).

algorithms associated with the Discrete Cosine Transform (DCT). For N sample values, the number of operations is then $O(N \log N)$. To recover the sample values of f from the coefficients $c_{n,k}$, we first apply an inverse cosine transform to get sample values of the folded function $\widetilde{f}$. This function can then be unfolded to produce f, see Exercise 9.6.

Adaptive Segmentation

As mentioned earlier, the construction of the basis functions $b_{n,k}$ works for an arbitrary partition of the real line. We seek an optimal partition, using, for instance, one of the cost functions described above. This can be done by adaptively merging intervals, starting from some initial partition. Let us consider merging the intervals $I_0 = [0, 1]$ and $I_1 = [1, 2]$ to $\widetilde{I}_0 = [0, 2]$. A window function for $\widetilde{I}_0$ is given by

$$\widetilde{w}_0(t) = \sqrt{w_0(t)^2 + w_1(t)^2},$$

which can be shown to satisfy the conditions (9.3a) -(9.3e), slightly modified. The basis functions associated with $\widetilde{I}_0$ are defined by

$$\widetilde{b}_{n,0}(t) = \frac{1}{\sqrt{2}}\, \widetilde{w}_0(t) \cos\left[\pi(n + \frac{1}{2})\frac{t}{2}\right].$$

Note that we analyze at frequencies $\frac{\pi}{2}(n + \frac{1}{2})$ after merging, so we get twice as good frequency resolution. At the same time, we lose time resolution. Denote the coefficients for the basis functions $b_{n,0}$, $b_{n,1}$, and $\widetilde{b}_{n,0}$ with $c_{n,0}$, $c_{n,1}$, and $\widetilde{c}_{n,0}$. The transform from $[c_0\, c_1]$ to $\widetilde{c}_0$ can be shown to be orthogonal. Therefore it pays off to merge I_0 and I_1 when $\Lambda(\widetilde{c}_0) < \Lambda(c_0) + \Lambda(c_1)$.

We can merge intervals $I_{2k} = [2k, 2k + 1]$, $I_{2k+1} = [2k + 1, 2k + 2]$ into $\widetilde{I}_k = [2k, 2k + 2]$ in the same way. These can then be merged further, which gives a tree structure similar to the wavelet packet tree. This tree can be searched with the same algorithm to find the optimal time segmentation with $O(N \log N)$ operations.

In Figure 9.12, the decomposition of the time-frequency plane is shown for a particular local trigonometric basis. We encourage the reader to figure out how the merging of intervals was done to obtain this basis, starting from no frequency resolution as in Figure 9.3.

Exercises 9.4

9.6. Show that the folded function $\widetilde{f}$ is given by

$$\widetilde{f}(k + t) = \begin{cases} w_k(k - t)f(k + t) - w_k(k + t)f(k - t), & \text{for } -\frac{1}{2} \le t \le 0, \\ w_k(k + t)f(k + t) + w_k(k - t)f(k - t), & \text{for } 0 \le t \le \frac{1}{2}. \end{cases}$$

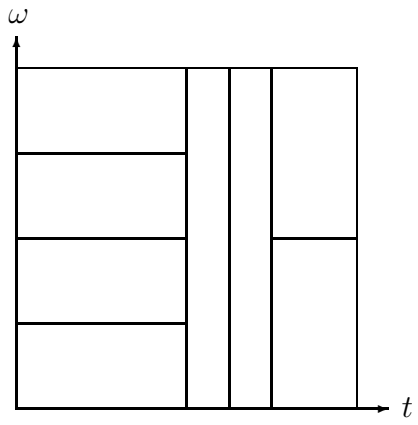

Figure 9.12: Tiling of the time-frequency plane with a Local Trigonometric basis.

Also show that f can be recovered from its folded version through

$$f(k+t) = \begin{cases} w_k(k+t)\widetilde{f}(k-t) + w_k(k-t)f(k+t), & \text{for } -\frac{1}{2} \leq t \leq 0, \\ w_k(k+t)f(k+t) - w_k(k-t)f(k-t), & \text{for } 0 \leq t \leq \frac{1}{2}. \end{cases}$$

9.7. Construct a tiling of the time-frequency plane that is impossible to obtain from wavelet packets or local trigonometric bases.

9.5 Notes

For further reading about wavelet packets and local trigonometric bases, and time-frequency decompositions in general, we refer to the paper *Lectures on Wavelet Packet Algorithms* [31] by Wickerhauser.

Wavelet packets can also be defined in the non-separable case. Wavelet packets for hexagonal wavelets give rise to some fascinating frequency plane decompositions, which can be found in the paper by Cohen and Schlenker [10].

Chapter 10

Compression and Noise Reduction

To date, perhaps the most successful application of wavelets is found in image compression. It is based on the observation that, for most images, a few large wavelet coefficients contain the 'relevant information' of the image, while the other coefficients are very small.

An analoguous consideration is the theoretical basis behind wavelet denoising. Since orthogonal wavelets transform white noise into white noise, noise will be uniformly spread over all wavelet coefficients. This makes it possible to extract the few large wavelet coefficients, and removing most of the noise by setting small wavelet coefficients to zero.

10.1 Image Compression

For simplicity, we will only work with grey-scale images $f(x, y)$, where $0 < x, y < 1$. We assume $0 \leq f(x, y) \leq 1$, where 0 is black and 1 is white. The extension to colour images is straightforward, since a colour image can be represented by three functions $f_r(x, y)$, $f_g(x, y)$, and $f_b(x, y)$, using the standard red-green-blue representation. A general image compression algo-

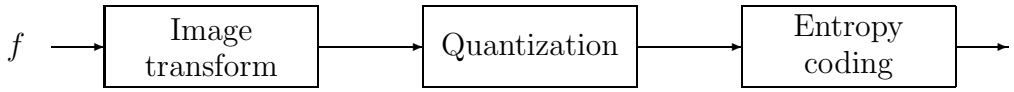

Figure 10.1: The general image compression algorithm

rithm consists of three parts as shown in figure 10.1, an *image transform*,

quantization, and *entropy coding*. The image transform is in general a linear, invertible transform that decorrelates the image to make compression possible. The quantization step maps the transformed coefficients into a smaller, finite set of values. This is the step in the compression algorithm, where some information is lost. Most of the compression is achieved in the entropy coding step. To recover the compressed image, we reverse the whole process as in Figure 10.2, yielding an approximation $\widehat{f}$ of the original image.

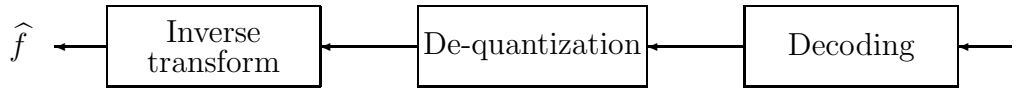

Figure 10.2: Decompression

Image Transform

Most images have spatial correlation, that is, neighbouring pixels tend to have similar grey-scale values. The purpose of the image transform is to exploit this redundancy to make compression possible. Remember the two-dimensional Haar transform, where groups of four pixel values are replaced by their mean value, and three wavelet coefficients or 'differences'. If those four pixel values are similar, the corresponding wavelet coefficients will essentially be 0. The averaging and differencing is repeated recursively on the mean values, to capture large-scale redundancy. For smooth regions of the image, most wavelet coefficients will be almost 0. Fine-scale wavelet coefficients are needed around edges, and in areas with rapid variations. A few large-scale coefficients will take care of slow variations in the image. For images without too much variation, a few wavelet coefficients contain the relevant information in the image. For images with *e.g.* texture, such as the fingerprint images, a wavelet packet transform might be more appropriate.

Most of the compression is achieved in the first filtering steps of the wavelet transform. Therefore, in wavelet image compression, the filter bank is usually iterated only a few times, say, 4 or 5. A smoother wavelet than the Haar wavelet is used, since compression with Haar wavelets leads to *blocking artifacts*; rectangular patterns appear in the reconstructed image. The choice of an optimal wavelet basis is an open problem, since there are many aspects to take into account.

First, we want synthesis scaling functions and wavelets to be smooth. At the same time, smoothness increases the filter length, and thus also the

support width of scaling functions and wavelets. Too long synthesis filters will give rise to ringing artifacts around edges. Further, we want all filters to be symmetric.

Another problem associated with wavelet image compression is *border artifacts*. The wavelet transform assumes that $f(x, y)$ is defined in the whole plane, and therefore the image need to be extended outside the borders. There are three extensions in practice: zero-padding, periodic extension, or symmetric extension. Zero-padding defines the image to be zero outside the borders. After compression, this will have a 'darkening' influence on the image near the border. Periodic extension assumes that the image extends periodically outside the borders. Unless the grey-scale values at the left border matches those at the right border *etc.*, periodic extension will induce discontinuities at the borders, which again lead to compression artifacts.

Generally, the best method is to use symmetric extension, which gives a continuous extension at the borders, and no compression artifacts appears. Symmetric extension requires symmetric filters. An alternative is to use so called boundary corrected wavelets. We will discuss the boundary problem further in Chapter 15.

Another possible image transform is the classical Fourier transform. If the signal is smooth, the Fourier coefficients decay very fast towards high frequencies, and the image can be represented using a fairly small number of low frequency coefficients. However, the presence of a single edge will cause the Fourier coefficients to decay very slowly, and compression no longer becomes possible. A way to get around this is to use a windowed Fourier transform instead. This is basically the JPEG algorithm, where the image is divided into blocks of 8×8 pixels, and a cosine transform is applied to each block. For blocks containing no edges, high frequency coefficients will be almost zero.

For high compression ratios, blocking artifacts appear in the JPEG algorithm, that is, the 8×8 blocks become visible in the reconstructed image. With properly choosen wavelets, wavelet image commpression works better. For moderate compression ratios, *e.g.* $1 : 10$, the performance of JPEG is comparable to wavelets.

Quantization and Entropy Coding

After the image transform, most coefficients will be almost zero. A crude compression algorithm would be to set all those to zero, and only store the few remaining significant coefficients. A more sophisticated algorithm applies a *quantization rule* to the transformed coefficients.

A quantization rule is a function $q : \mathbb{R} \to \{0, 1, \ldots, K\}$. A coefficient

value $x > 0$ is assigned the integer k if it belongs to an interval $(d_k, d_{k+1}]$. The numbers d_k are called the *decision points* of the quantization rule. A negative value of x will be assigned to a corresponding interval on the negative axis.

The *de-quantization rule* assigns a *reconstruction level* $r_k \in (d_k, d_{k+1}]$ to the integer k. Usually, r_k is chosen as the midpoint of the interval $[d_k, d_{k+1}]$. Quantization followed by de-quantization becomes a piecewise constant function that assigns all $d_k < x \leq d_{k+1}$ the same value r_k (Figure 10.3).

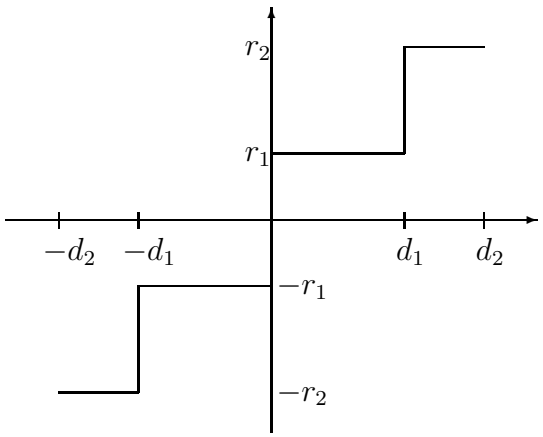

Figure 10.3: Quantization followed by de-quantization.

The design of the decision points d_k is important to obtain good compression. In wavelet image compression, different quantization rules are used for different scales. More bits, and thus more decision points are used in 'key' frequency bands. Usually, the highest frequency bands are assigned very few bits, and more and more bits are assigned to the lower frequency bands. Within each band, the decision points are decided from a statistical model of the wavelet/scaling coefficients.

After quantization, we are left with a set of integer coefficients, of which many are zero. To achieve significant compression, all these 0's need to be stored in an efficient way. One way to do this is to order the wavelet coefficients in a clever way to obtain long strings of 0's. These strings can then be represented by their length, using, for instance, run-length coding. This is an example of *entropy coding*. Other encoding schemes are also possible, *e.g.*, Huffman coding. Entropy coding is a lossless step, and thus invertible. The only lossy step in the compression algorithm is the quantization.

Video Compression

A video signal is a sequence of images $f_i(x, y)$. Each second contains approximately 30 images, so the amount of information is huge. In order to transfer video signals over the internet or telephone lines (video conferencing), massive compression is necessary. The simplest approach to video compression is to compress each image f_i seperately. However, this method does not exploit temporal correlation in the video signal: adjacent frames tend to be very similar. One way to do this is to treat the video signal as a 3D signal $f(x, y, t)$ and apply a three-dimensional wavelet transform.

Another method is to compute the difference images

$$\Delta f_i = f_{i+1} - f_i.$$

Together with an initial image f_0 these difference images contain the information necessary to reconstruct the video signal. The difference images contain the changes between adjacent frames. For parts of the images without movement, the difference images will be zero, and thus we have already achieved significant compression. Further compression is obtained by exploiting spatial redundancy in the difference images, and applying a two-dimensional wavelet transform W to all Δf_i and to f_0. The transformed images $W\Delta f_i$ and Wf_0 are quantized and encoded, and then transmitted/stored.

To reconstruct the video signal, inverse wavelet transforms are computed to get back approximate $\widehat{\Delta f}_i$ and $\widehat{f_0}$. From there, we can recover the video signal approximately by

$$\widehat{f}_{i+1} = \widehat{f}_i + \widehat{\Delta f}_i.$$

The sparse structure of Δf_i can be used to speed up the inverse wavelet transforms. For wavelets with compact support, each scaling and wavelet coefficient only affect a small region of the image. Thus we only need to compute pixels in Δf_i corresponding to non-zero scaling and wavelet coefficients.

The difference scheme just described is a special case of *motion estimation*, where we try to predict a frame f_i from M previous frames

$$\widehat{f}_i = P(f_{i-1}, \ldots, f_{i-M}),$$

and apply the wavelet transform to the prediction errors $\Delta f_i = \widehat{f}_i - f_i$. The predictor P tries to discover motions in the video to make an accurate guess about the next frame. The wavelet transform is well suited for this, since it contains local information about the images.

An Example of Wavelet Image Compression

We conclude our discussion about image compression by including a simple example of wavelet compression. All computations were done using the WaveLab package. As test image we used the famous 512×512 pixels *Lena* image, which is a standard test image for image compression. We used the crude wavelet compression algorithm, and kept the $p\,\%$ largest wavelet coefficients by amplitude. We used the FBI 9/7 filter pair and repeated the filter bank 5 times. We tried four different values of the 'compression parameter', $p = 100$ (no compression), $p = 10$, 4, and 1. Note that p does not exactly correpond to the compression ratio, since, in practice, also the zeroes need to be stored. In Figures 10.4 and 10.5 we show the decompressed images. For $p = 10$, there is hardly no difference to the original image. For $p = 4$, the image quality is still good, but there is some ringing around edges. For $p = 1$, the quality of the decompressed image is quite poor.

10.2 Denoising

Suppose that a signal $f(t)$ is sampled on the unit interval $[0, 1]$ at the points $t_k = 2^{-J}k$, $k = 1, \ldots, K = 2^J$. Denote the exact sample values by $f_k = f(t_k)$. Assume that we only have noisy mearuments of f_k, *i.e.*, we have data $y_k = f_k + \sigma z_k$. Here, z_k is assumed to be Gaussian white noise, *i.e.*, independent normally distributed random variables with mean 0 and variance 1. The parameter σ is the noise level, which is generally unknown, and has to be estimated from the data.

We want to recover f from the noisy data. Applying an orthogonal discrete wavelet transform W, yields

$$Wy = Wf + \sigma Wz$$

Orthogonal transforms turns Gaussian white noise into Gaussian white noise. If we denote the wavelet coefficients of y_k by $\gamma_{j,k}$, and f:s wavelet coefficients by $w_{j,k}$, we thus have

$$\gamma_{j,k} = w_{j,k} + \sigma \widetilde{z}_{j,k},$$

where $\widetilde{z}_{j,k}$ is Gaussian white noise.

In Figure 10.6 we have plotted the *HeaviSine* test function with and without noise. We have also included a 'spike plot' of both the exact and noisy wavelet coefficients. The fine-scale wavelet coefficients are at the bottom of the figure, and the coarse-scale coefficients at the top. From this figure it is clear that it is possible to remove most of the noise by removing 'small'

Figure 10.4: Original image and 1:10 compression.

p=4

p=1

Figure 10.5: Decompressed images with 1:25 and 1:100 compression.

wavelet coefficients $\gamma_{j,k}$ that mostly contain noise, and extract the test signal by keeping large coefficients. Note the similarity with the crude wavelet image compression algorithm.

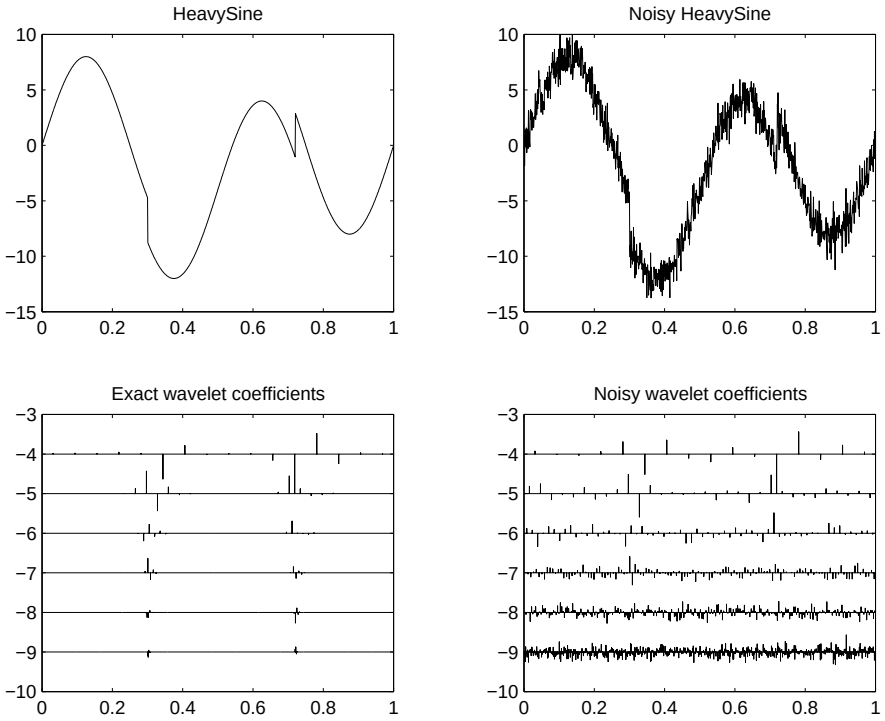

Figure 10.6: The HeavySine test function and its wavelet coefficients, with and without noise.

Hard and Soft Thresholding

The noise reduction algorithm suggested above can be seen as applying a *threshold function* $\eta_T(w)$ to the noisy wavelet coefficients. In this case, we are applying *hard thresholding*

$$\eta_T(w) = \begin{cases} w & \text{if } |w| \geq T, \\ 0 & \text{otherwise.} \end{cases} \tag{10.1}$$

Coefficients with absolute value less than some *threshold* T are shrunk to 0, and all other coefficients are left unaltered. The threshold need to be properly choosen, depending on the noise level σ. There is also a *soft thresholding*,

where the threshold function is defined by

$$\eta_T(w) = \begin{cases} w - T & \text{if } w \geq T, \\ w + T & \text{if } w \leq -T, \\ 0 & \text{otherwise.} \end{cases} \tag{10.2}$$

The difference compared to hard thresholding is that coefficients with absolute value greater than T are shrunk by an amount of T. In Figure 10.7, we plot those two threshold functions. Other threshold functions are also possible, *e.g.*, combinations of hard and soft thresholding.

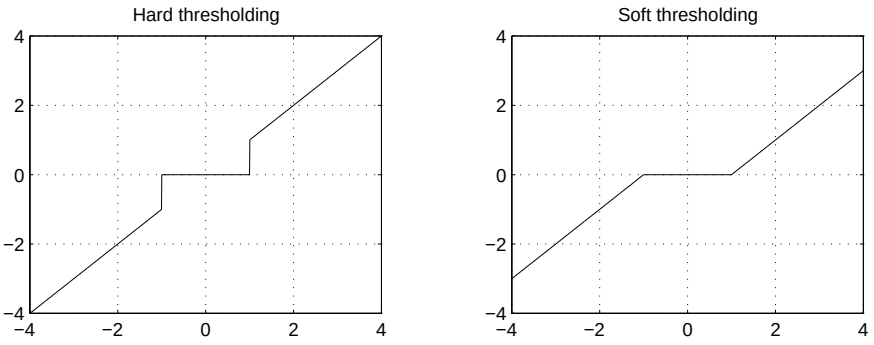

Figure 10.7: Hard and soft thresholding.

Algorithm. (**Wavelet Denoising**).

1. Compute noisy wavelet coefficients $\gamma_{j,k}$, $j = j_0, \ldots, J-1$, and noisy scaling coefficients $\lambda_{j_0,k}$.

2. Choose the threshold as $T = \sqrt{2 \log N}$.

3. Estimate scaling and wavelet coefficients by

$$\begin{aligned} \widehat{s}_{j_0,k} &= \lambda_{j_0,k}, \\ \widehat{w}_{j,k} &= \eta_T(\gamma_{j,k}). \end{aligned}$$

4. Obtain estimates $\widehat{f}_i$ by applying an inverse wavelet transform to $\widehat{s}_{j_0,k}$ and $\widehat{w}_{j,k}$.

□

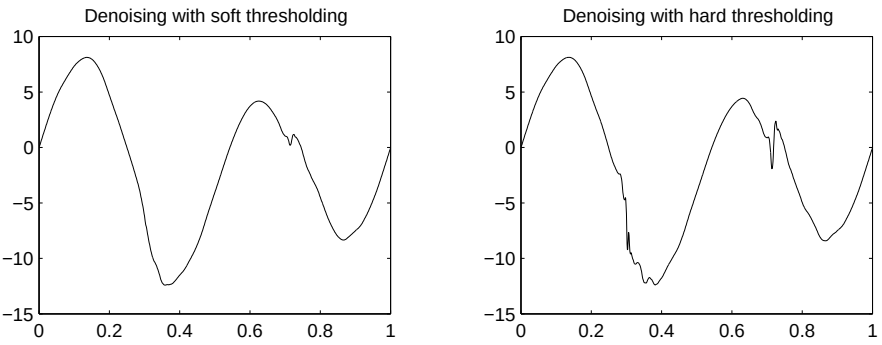

Figure 10.8: The de-noised signal.

In Figure 10.8, we show the result of applying wavelet denoising to the test signal in Figure 10.6.

We remark that the threshold choice above can be theoretically justified. With this threshold, the wavelet de-noising algorithm can be shown to be optimal in a certian sense. If $\widehat{f}$ is any estimate of f from y, we define the *risk* associated with this estimator as the expected mean squared error

$$R(\widehat{f}, f) = \frac{1}{K} E\left[\sum_{k=1}^{K} (\widehat{f}_k - f_k)^2\right].$$

Denoising with soft thresholding minimizes the risk under the constraint that $\widehat{f}$ should with high propability be at least as smooth as f. We will not make this more precise. Instead, we just mention that hard thresholding generally leads to smaller mean squared error than soft thresholding, but that the estimator $\widehat{f}$ is not so smooth.

In a practical application, the threshold has to be fine-tuned to the particular class of signals under consideration. Also, the noise level is in general unknown, and has to be estimated from the data. This is done using the wavelet coefficients on the finest scale, since the influence from the signal f is usually less there. The noise level is estimated as

$$\widehat{\sigma} = \text{Median}(|\gamma_{J-1,k}|)/.6745.$$

The reason for using a median estimate is to reduce the influence of outliers, *i.e.*, noisy wavelet coefficients with a large signal content. The same estimate can be used for noise reduction with coloured noise. In that case, scale-dependent thresholds are chosen as

$$T_j = \sqrt{2 \log K}\, \text{Median}(|\gamma_{j,k}|)/.6745.$$

The wavelet coefficients at each scale are then thresholded according to these thresholds.

Wavelet Packet Denoising

The performance of the denoising algorithm depends on how much the entropy of the signal is decreased in the wavelet transform, that is, to what extent most coefficients are small, and very few coefficients are large. This suggests that it might be worthwile to use the best basis algorithm of Chapter 9, and apply soft or hard thresholding to the optimal wavelet packets coefficients. However, we can not use, *e.g.*, the entropy measure since this requires knowledge of the exact wavelet coefficients, which we do not know. Computing the entropy of the noisy wavelet coefficients would be very misleading because of the white noise, which has a very high entropy.

Instead, one can use as cost function an estimate of the risk $R(f, \widehat{f})$, where $\widehat{f}$ is the estimator of f obtained from thresholding the noisy wavelet packet coefficients. For soft thresholding, such an estimate is given by *Stein's Unbiased Risk Estimator*

$$\text{SURE}(c) = \sigma^2(K - 2\sharp\{|c_k| \leq T\}) + \sum_{k=1}^{K} \min(c_i^2, T^2), \tag{10.3}$$

where c_k are the noisy wavelet packet coefficients in a particular wavelet packet basis. Using the best basis algorithm with the SURE cost function, can improve noise reduction in certain cases.

10.3 Notes

The overview article by Jawerth *et al.* [17] is a good starting point for further reading about wavelet image and video compression. The book by Nguyen & Strang [27] contains a more detailed discussion about various aspects of image compression, such as quantization, entropy coding, border extension, and filter design. Also, comparisons between wavelet image compression and the JPEG compression algorithm can be found.

The overview article [13] by Donoho contains a comprehensive description of wavelet denoising algorithms. It also includes several numerical examples, both for synthetic and real-world signals.

Chapter 11

Fast Numerical Linear Algebra

In this chapter we will study the numerical solution of linear equations. Typically, we are given a linear operator T and a function f, and we seek the solution u of the equation

$$Tu = f.$$

The linear operator T is either a differential or integral operator. Discretization of such an equation leads to a system of linear equations with a large number of unknowns. The discretization normally starts at some coarse scale, or grid, which is successively refined giving a sequence of linear equations denoted as

$$T_j u_j = f_j.$$

Here T_j is a matrix, or equivalently, an operator on a suitable finite-dimensional space. For differential equations this matrix is a sparse and ill-conditioned, and for integral equations the matrix is full and, depending on the operator, sometimes ill-conditioned. Today these equations are solved using various iterative methods. The most efficient methods are multilevel or multigrid methods. These take advantage of the fact that we have a sequence of scales, or operators, and are also quite simple. Recently it has been proposed to use wavelets for the solution of both linear and nonlinear equations. We will describe the so called non-standard form of an operator in a wavelet basis, and see how it relates to standard multilevel methods.

11.1 Model Problems

We will restrict our attention to differential and integral equations that are elliptic. By this we mean that the discretization matrices are symmetric and positive definite.

As an example of an elliptic differential equation we consider the Laplace equation in one dimension

$$\begin{aligned} -(a(x)u'(x))' &= f(x), \quad 0 < x < 1, \\ u(0) &= u(1) = 0. \end{aligned} \tag{11.1}$$

where $0 < \underline{a} < a(x) < \overline{a}$.

The following integral equation is a simple type of a boundary layer potential in one dimension

$$-\int_0^1 u(y) \log|x - y| \, dy = f(x), \quad 0 < x < 1. \tag{11.2}$$

The kernel is called the *logarithmic potential* and is more commonly defined on a curve in the plane.

11.2 Discretization

We will consider the Galerkin method for the discretization of linear operator equations. There exists several other discretization methods, such as the finite difference method for differential equations and the collocation method for integral equations. These can be seen as special cases of the Galerkin method, though, with certain choices of quadrature formulas and function spaces. In any case, wavelet and other multilevel methods work in the same way.

Now, assume that the operator $T : V \to V$, where V is some Hilbert space such as $L^2(\mathbb{R})$. The equation $Tu = f$ is then equivalent to finding a $u \in V$ such that

$$\langle Tu, v \rangle = \langle f, v \rangle \quad \text{for all } v \in V.$$

This is called the variational formulation of the operator equation. We now seek an approximation u_j of u in some finite-dimensional subspace V_j of V. The Galerkin method is a finite-dimensional analogue of the variational form: find $u_j \in V_j$ such that

$$\langle Tu_j, v_j \rangle = \langle f, v_j \rangle \quad \text{for all } v_j \in V_j. \tag{11.3}$$

Let us further assume that $(\varphi_k)_{k=1}^N$ is a basis of V_j, so that we can write

$$u_j = \sum_{k=1}^{N} a_k \varphi_k,$$

for some numbers (a_k). If we substitute this expression into equation (11.3), and use the fact that $(\varphi_k)_{k=1}^N$ is a basis of V_j, equation (11.3) is equivalent to

$$\sum_{k=1}^{N} a_k \langle T\varphi_k, \varphi_n \rangle = \langle f, \varphi_n \rangle, \quad \text{for } n = 1, \ldots, N.$$

But this a linear system of equations, which we write in matrix form as

$$T_j u_j = f_j,$$

where T_j is an $N \times N$ matrix with elements

$$(T_j)_{n,k} = \langle T\varphi_k, \varphi_n \rangle,$$

and where u_j and f_j are vectors with components a_n and $\langle f, \varphi_n \rangle$, respectively.

Throughout this chapter we will, with a slight abuse of notation, use the the same symbol T_j to denote both a matrix and an operator on V_j. Similarly, we use the notation u_j to denote both a vector with components a_k, and the corresponding function $u_j = \sum_{k=1}^N a_k \varphi_k$.

Example 11.1. For the differential equation (11.1) the elements of the matrix T_j are (by partial integration)

$$(T_j)_{n,k} = \int_0^1 \varphi_k'(x)\varphi_n'(x)\, dx,$$

and the components of the right-hand side vector f_j equal

$$(f_j)_n = \int_0^1 f(x)\varphi_n(x)\, dx.$$

Here, a natural choice of the finite-dimensional spaces V_j is the space of piecewise linear and continuous functions on a grid with $N_j = 2^j$ nodes. The basis functions spanning this space are the hat functions, with support on two intervals. The matrix T_j is then tri-diagonal and thus sparse. Unfortunately, the condition number of the matrix is proportional to N_j^2.

Example 11.2. For the integral equation (11.2) the elements of the matrix T_j are given by

$$(T_j)_{n,k} = -\int_0^1 \int_0^1 \varphi_k(x)\varphi_n(x) \log|x - y|\, dxdy,$$

and the components of the right-hand side vector f_j are given by

$$(f_j)_n = \int_0^1 f(x)\varphi_n(x)\,dx.$$

There are several natural choices of the finite dimensional spaces V_j for this equation, and the simplest is to let V_j be the space of piecewise constant functions on a grid with $N_j = 2^j$ intervals. The basis functions spanning this space are the box functions. In this case the matrix T_j is full because of the coupling factor $\log|x-y|$ in the integrand. The condition number of the matrix is also large (since the continuous operator T is compact).

11.3 The Non-Standard Form

First, assume we are given a matrix (or operator) T_J on some fine-scale subspace V_J of a multiresolution analysis. We then seek the solution u_J of the linear system

$$T_J u_J = f_J.$$

Recall that this linear system was equivalent to the set of equations

$$\sum_{k=1}^{N_J} u_{J,k}\langle T\varphi_{J,k}, \varphi_{J,n}\rangle = \langle f, \varphi_{J,n}\rangle, \quad \text{for } n = 1, \ldots, N_J,$$

where $u_J = \sum_{k=1}^{N_J} u_{J,k}\varphi_{J,k}$. From the subspace splitting $V_J = V_{J-1} \oplus W_{J-1}$ we can also write u_J as

$$u_J = \sum_{k=1}^{N_{J-1}} u_{J-1,k}\varphi_{J-1,k} + \sum_{k=1}^{N_{J-1}} w_{J-1,k}\psi_{J-1,k},$$

and this induces the following decomposition of the linear system

$$\sum_{k=1}^{N_{J-1}} u_{J-1,k}\langle T\varphi_{J-1,k}, \varphi_{J-1,n}\rangle + \sum_{k=1}^{N_{J-1}} w_{J-1,k}\langle T\psi_{J-1,k}, \varphi_{J-1,n}\rangle = \langle f, \varphi_{J-1,n}\rangle,$$
$$\sum_{k=1}^{N_{J-1}} u_{J-1,k}\langle T\varphi_{J-1,k}, \psi_{J-1,n}\rangle + \sum_{k=1}^{N_{J-1}} w_{J-1,k}\langle T\psi_{J-1,k}, \psi_{J-1,n}\rangle = \langle f, \psi_{J-1,n}\rangle,$$

for $n = 1, \ldots, N_{J-1}$. This follows since the scaling functions $(\varphi_{J-1,k})_{k=1}^{N_{J-1}}$ is a basis of V_{J-1} and the wavelets $(\psi_{J-1,k})_{k=1}^{N_{J-1}}$ is a basis of W_{J-1}. We write

this as the following block matrix system

$$\begin{pmatrix} T_{J-1} & C_{J-1} \\ B_{J-1} & A_{J-1} \end{pmatrix} \begin{pmatrix} u_{J-1} \\ w_{J-1} \end{pmatrix} = \begin{pmatrix} f_{J-1} \\ d_{J-1} \end{pmatrix}. \tag{11.4}$$

The non-standard form of an operator is obtained if we continue this splitting recursively on T_{J-1} until we reach some coarse-scale operator T_L, where $L < J$. The matrices, or operators, of the non-standard form are thus defined as

$$\begin{aligned} A_j &: W_j \to W_j, & (A_j)_{n,k} &= \langle T\psi_{j,k}, \psi_{j,n} \rangle, \\ B_j &: V_j \to W_j, & (B_j)_{n,k} &= \langle T\varphi_{j,k}, \psi_{j,n} \rangle, \\ C_j &: W_j \to V_j, & (C_j)_{n,k} &= \langle T\psi_{j,k}, \varphi_{j,n} \rangle, \\ T_j &: V_j \to V_j, & (T_j)_{n,k} &= \langle T\varphi_{j,k}, \varphi_{j,n} \rangle. \end{aligned}$$

We can therefore represent the operator T_J as the coarse scale operator T_L, plus the sequence of triplets $\{A_j, B_j, C_j\}_{j=L}^{J-1}$.

In a multiresolution context, the scale j refers to the subspace W_j and the functions therein. We then note that the operator A_j describe the interaction on the scale j, and the operators B_j and C_j describe the interaction between the scale j and all coarser scales. Also, the operator T_j is an averaged version of the operator T_{j+1}. These properties of the operators reveals a remarkable feature of the non-standard form. Namely, the decoupling of the interaction between the different scales.

It is also important to note that we can not represent the non-standard form as a single block matrix, that is, it is not a representation of the operator in any basis. The non-standard form is recursive in nature, and is based on the nesting property $\cdots \subset V_0 \subset V_1 \subset \cdots$ of the subspaces of a multiresolution analysis.

It is convenient to store the non-standard form (of a discretization matrix) as a block matrix as depicted to the left in Figure 11.1, though. This block matrix can also be seen as the result of taking the two-dimensional discrete wavelet transform of the discretization matrix T_J. Compare, Chapter 8 on wavelet bases in several dimensions and Chapter 10 on image compression.

We have in this section only derived the algebraic properties of the non-standard form. This means that it is also applicable to all other bases (besides wavelet bases) that have the nesting property, for example, to hierarchical bases.

11.4 The Standard Form

The standard form of an operator is just the representation, or discretization, of the operator in a wavelet basis. If V_J is a subspace of an MRA, we can

decompose this space as

$$V_J = V_L \oplus W_L \oplus \cdots \oplus W_{J-1}, \quad \text{where } L < J.$$

We know that for an MRA the scaling functions $(\varphi_{j,k})_{k=1}^{N_j}$ span the V_j-spaces and the wavelets $(\psi_{j,k})_{k=1}^{N_j}$ span the W_j-spaces. By a change of basis the equation $T_J u_J = f_J$ on V_J can be written as the following block matrix system

$$\begin{pmatrix} T_L & C_{L,L} & \dots & C_{L,J-1} \\ B_{L,L} & A_{L,L} & \dots & A_{L,J-1} \\ \vdots & \vdots & \ddots & \vdots \\ B_{J-1,L} & A_{J-1,L} & \dots & A_{J-1,J-1} \end{pmatrix} \begin{pmatrix} u_L^j \\ w_L^j \\ \vdots \\ w_{J-1}^j \end{pmatrix} = \begin{pmatrix} f_L \\ d_L \\ \vdots \\ d_{J-1} \end{pmatrix}.$$

The standard form does not fully utilize the hierarchical structure of a multiresolution analysis. Therefore we will not consider it any further in this chapter.

11.5 Compression

So far we have said nothing about the structure of the A_j, B_j, and C_j matrices. Since wavelets have vanishing moments it turns out that these matrices are sparse. More specifically, for one-dimensional problems they have a rapid off-diagonal decay. In the case the T_j matrices are ill-conditioned the A_j matrices are even well conditioned, making the non-standard form a suitable representation for iterative methods.

Let us now prove that integral operators produce sparse matrices in the non-standard form. Start with an integral operator with kernel $K(x, y)$

$$Tu(x) = \int K(x, y)u(y)\, dy.$$

For the moment we assume that the kernel $K(x, y)$ is smooth away from the diagonal $x = y$, where it is singular. Typical examples of such kernels are the following

$$K(x, y) = -\log|x - y|, \qquad \text{(logarithmic potential)}$$
$$K(x, y) = \frac{1}{x - y}. \qquad \text{(Hilbert transform)}$$

For simplicity we will use the Haar wavelet basis, which has one vanishing moment, that is,

$$\int \psi_{j,k}(x)\,dx = 0.$$

The support of the Haar wavelet $\psi_{j,k}$, as well as the scaling function $\varphi_{j,k}$, is the interval $I_{j,k} = [2^{-j}k, 2^{-j}(k+1)]$. Now, consider the elements of the matrix B_j (the A_j and C_j matrices are treated similarly)

$$(B_j)_{n,k} = \int_{I_{j,k}} \int_{I_{j,n}} K(x,y)\psi_{j,n}(x)\varphi_{j,k}(y)\,dxdy.$$

If we assume that $|k-n| > 1$, so that $K(x,y)$ is smooth on the domain of integration, we can make a Taylor expansion of $K(x,y)$ around the midpoint $x_0 = 2^{-j}(k+1/2)$ of the interval $I_{j,k}$

$$K(x,y) = K(x_0,y) + (x-x_0)\partial_x K(\xi,y), \quad \text{for some } \xi \in I_{j,k}.$$

Since the Haar wavelet has one vanishing moment it follows that

$$\int_{I_{j,n}} K(x,y)\psi_{j,n}(x)\,dx = \int_{I_{j,n}} \partial_x K(\xi,y)x\psi_{j,n}(x)\,dx,$$

for some $\xi \in I_{j,k}$. Therefore,

$$\left| \int_{I_{j,n}} K(x,y)\psi_{j,n}(x)\,dx \right| \leq C\,|I_{j,n}| \max_{x\in I_{j,n}} |\partial_x K(x,y)|\,,$$

where $C = \int x\psi_{j,n}(x)\,dx$ and $|I_{j,n}| = 2^{-j}$. This gives us an estimate for the size of the elements of B_j

$$\begin{aligned} |(B_j)_{n,k}| &\leq C2^{-j} \int_{I_{j,k}} \max_{x\in I_{j,n}} |\partial_x K(x,y)|\;dy \\ &\leq C2^{-2j} \max_{x\in I_{j,n}, y\in I_{j,k}} |\partial_x K(x,y)|\,. \end{aligned}$$

To proceed from here we need to know the off-diagonal decay of the kernel. For the logarithmic potential we have

$$|\partial_x K(x,y)| = |x-y|^{-1}\,,$$

and this implies that

$$|(B_j)_{n,k}| \leq C2^{-j}(|k-n|-1)^{-1}.$$

Exactly the same estimate will hold for the elements of C_j. For the matrix A_j we can also make a Taylor expansion of the kernel in the y-variable, giving an even faster off-diagonal decay of its elements

$$|(A_j)_{n,k}| \leq C2^{-j}(|k-n|-1)^{-2}.$$

For other integral operators similar estimates for the off-diagonal decay will hold. Also, the number of vanishing moments of the wavelet will increase the degree of the decay. As a matter of fact, there is a large class of integral operator, referred to as Calderón-Zygmund operators, for which one can prove a general estimate. A Calderón-Zygmund operator is a bounded integral operator on $L^2(\mathbb{R})$, with a kernel satisfying the estimates

$$|K(x,y)| \leq C_0 \left|x-y\right|^{-1},$$
$$|\partial_x K + \partial_y K| \leq C_1 \left|x-y\right|^{-N-1},$$

For such an operator one can prove that

$$|(A_j)_{n,k}| + |(B_j)_{n,k}| + |(C_j)_{n,k}| \leq C_N 2^{-j}(|k-n|+1)^{-N-1},$$

if the wavelet has N number of vanishing moments.

Example 11.3. To the right in Figure 11.1 we have marked all matrix elements of the A_j, B_j, and C_j blocks exceeding a small threshold. As can be seen we get good band-diagonal approximations of these matrices. The matrix is the non-standard representation of the logarithmic potential in the Haar wavelet basis. □

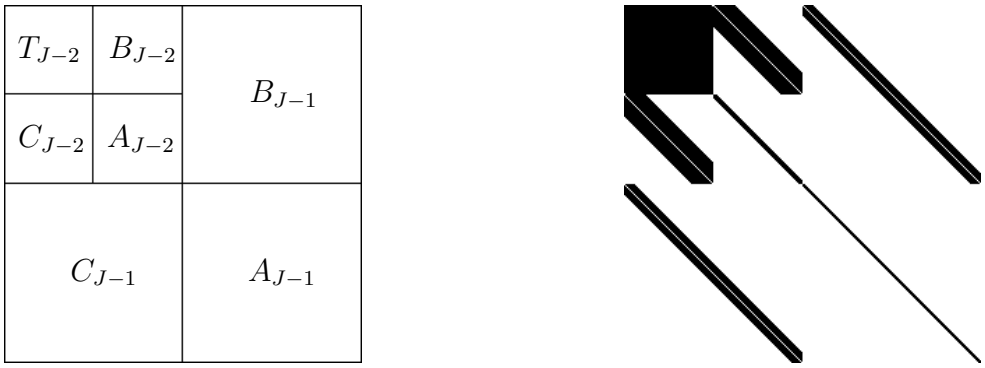

Figure 11.1: The non-standard form of the logarithmic potential.

11.6 Multilevel Iterative Methods

We will now propose a method for the iterative solution of a system of equations based on the non-standard form. The method is very similar to a multigrid V-cycle. It is here assumed that the reader is familiar with basic iterative methods such as Jacobi and Gauss-Seidel. Further, a knowledge of the multigrid method is needed in order to understand the connection with our wavelet based method. We refer the reader to the references in the Notes section at the end of the chapter.

To start with, let us make a comparison between the condition numbers of the T_j and A_j matrices. In Table 11.1 we have listed the condition numbers for the discretization of the logarithmic potential in the Haar Basis. Here N is the number of basis functions in V_j. Wee see that the condition number of T_j grows linearly in the number of unknowns N. On the other hand, the condition number of A_j stays bounded for all j and is about 2. This suggest that we iterate with the matrices A_j, that is, on the W_j-spaces.

j	N	$\kappa(T_j)$	$\kappa(A_j)$
5	32	57	1.94
6	64	115	1.97
7	128	230	1.98
8	256	460	1.99

Table 11.1: Condition numbers for the logarithmic potential.

To begin with, we rewrite the block matrix form (11.4) of $T_J u_J = f_J$ as

$$\begin{aligned} A_{J-1} w_{J-1} &= d_{J-1} - B_{J-1} u_{J-1}, \\ T_{J-1} u_{J-1} &= f_{J-1} - C_{J-1} w_{J-1}. \end{aligned}$$

Inspired by the multigrid method, we begin by solving the first equation for w_{J-1} approximately using a simple smoother. This should work well since A_{J-1} is well conditioned. Next, we update the right-hand side of the second equation and solve it for u_{J-1}. Now, since T_{J-1} is still ill conditioned we solve this equation recursively by splitting T_{J-1} one step further. When we have reached a sufficiently coarse-scale operator T_L, we solve the equation for u_L exactly though. Finally, we update the right-hand side of the first equation and repeat the above steps a number of times.

Based on this we propose the following recursive algorithm, see Figure 11.2. The number of loops K is a small number, typically less than 5. The

function $u_j = \text{Solve}(u_j^{(0)}, f_j)$

if $j = L$
  Solve $T_j u_j = f_j$ using Gaussian elimination
else
  Project $u_j^{(0)}$ onto V_{j-1} and W_{j-1} to get $u_{j-1}^{(0)}$ and $w_{j-1}^{(0)}$
  Project f_j onto V_{j-1} and W_{j-1} to get f_{j-1} and d_{j-1}
  for $k = 1, \ldots, K$
    $u_{j-1}^{(k)} = \text{Solve}(u_{j-1}^{(k-1)}, f_{j-1} - C_{j-1} w_{j-1}^{(k-1)})$
    $w_{j-1}^{(k)} = \text{Iter}(w_{j-1}^{(k-1)}, d_{j-1} - B_{j-1} u_{j-1}^{(k)})$
  end
  $u_j = u_{j-1}^{(K)} + w_{j-1}^{(K)}$
end

Figure 11.2: The wavelet-multigrid method.

function $\text{Iter}(w_j^{(0)}, d_j)$ solves $A_j w_j = d_j$ approximately using a simple iterative method with initial vector $w_j^{(0)}$.

11.7 Notes

The non-standard form of an operator was invented by Beylkin, Coifman, and Rokhlin in [4], *Fast Wavelet Transforms and Numerical Algorithms I.* It can be seen as a generalisation of the Fast Multipole Method (FMM) for computing potential interactions by Greengard and Rokhlin, see *A Fast Algorithm for Particle Simulations* [15]. Methods for solving equations in the non-standard form has further been developed primarily by Beylkin, see for example *Wavelets and Fast Numerical Algorithms* [3]. For an introduction to multigrid and iterative methods we refer the reader to the book *A Multigrid Tutorial* [5] by Briggs.

Chapter 12

Functional Analysis

Functional analysis in this chapter will mean the study of global differentiability of functions expressed in terms of their derivatives being, say, square integrable (or belonging to certain Banach spaces). The corresponding wavelet descriptions will be made in terms of orthogonal MRA-wavelets (Chapter 4).

Wavelet representations are also well suited to describe local differentiability properties of functions. However, for this matter, we will only give references in Notes at the end of the chapter.

12.1 Differentiability and Wavelet Representation

Differentiability properties of a function may be expressed in terms of the behaviour of its Fourier transform for large values of the frequency variable. Analogously, these properties can be described in terms of the behaviour of the wavelet coefficients on small scales.

In order to avoid introducing additional mathematical techniques (Paley-Littlewood decompositions of L^p), we will mainly consider L^2, *i.e.*, $p = 2$. The norm in L^p is defined by

$$\|f\|_p := \left(\int_{-\infty}^{\infty} |f(t)|^p \, dt \right)^{1/p}$$

Moreover, we consider only orthogonal wavelets associated with an MRA, which vanish outside a bounded interval.

Recall that, for a 2π-periodic function $f \in L^2(0, 2\pi)$ with $f^{(\alpha)} \in L^2(0, 2\pi)$,

it is possible to show that for the Fourier coefficients c_n (Parseval)

$$\frac{1}{2\pi}\|f^{(\alpha)}\|_2^2 = \sum_{n\neq 0} |n^\alpha c_n|^2. \tag{12.1}$$

We now come to the main result of this chapter. The following theorem has an extension to $L^p, 1 < p < \infty$ cited below. However, the crucial lemma is essentially the same for $p = 2$.

Theorem 12.1. Under the same assumptions on the function f now defined on $\mathbb{R}$, *i.e.*, $f \in L^2(\mathbb{R})$ with $f^{(\alpha)} \in L^2(\mathbb{R})$, we will show that for the wavelet coefficients $w_{j,k}$ ($D^N\psi \in L^2$)

$$\|f^{(\alpha)}\|_2^2 \sim \sum_{j,k} |2^{\alpha j} w_{j,k}|^2 \quad (0 \le \alpha \le N)$$

holds when $\int x^\alpha \psi(x)dx = 0$ for $0 \le \alpha \le N$. Here $\sim$ denotes that the quotient of the two expressions is bounded from below and from above by positive constants depending only on the wavelet ψ.

□

The proof will be based on a lemma related to the so-called Bernstein and Jackson theorems. The Bernstein theorem expresses the behaviour of the derivatives of a function in terms of the function and its spectral content. The Jackson theorem describes the spectral approximability of a function in terms of its differentiability.

Lemma: For $f(x) = \sum_k w_{0,k}\psi(x-k)$, ($D^\alpha f \in L^2$) holds ($0 \le \alpha \le N$)

$$\|D^\alpha f\|_2^2 \sim \|f\|_2^2$$

Proof of the theorem: Apply the lemma with $g(x) = f(2^{-j}x)$ to obtain $\|D^\alpha f\|_2^2 \sim 2^{2j\alpha}\sum_k |w_{0,k}|^2$ when $f(x) = \sum_k w_{0,k} 2^{j/2}\psi(2^j x - k)$. This proves the theorem. (In general, only a summation over the dilation index, j, remains and the corresponding subspaces are orthogonal.)

□

Proof of the lemma: Starting from $D^\alpha f(x) = \sum_k w_{0,k} D^\alpha\psi(x-k)$, note that $|D^\alpha f(x)|^2 \le \sum_k |w_{0,k}|^2 |D^\alpha\psi(x-k)| \sum_k |D^\alpha\psi(x-k)|$ by the Cauchy-Schwarz inequality. Integrating this yields

$$\|D^\alpha f\|_2^2 \le \sup_x \sum_k |D^\alpha\psi(x-k)| \sum_k |w_{0,k}|^2 \int |D^\alpha\psi(x-k)|dx \le C\sum_k |w_{0,k}|^2.$$

The proof of the remaining inequality is based on the existence of a function $\Psi(x)$ vanishing outside a bounded interval such that $D^\alpha \Psi = \psi$, $\alpha \leq N$ fixed. Assuming this, we have (partial integration)

$$\begin{aligned} f(x) &= \sum_k w_{0,k}\psi(x-k) = \sum_k \int f(y)\psi(y-k)\,dy\,\psi(x-k) \\ &= (-1)^\alpha \sum_k \int D^\alpha f(y)\Psi(y-k)\,dy\,\psi(x-k) \end{aligned}$$

and

$$\begin{aligned} \|f\|_2^2 &= \int\int |\sum_k D^\alpha f(y)\Psi(y-k)\psi(x-k)|^2 dxdy \leq \\ &\leq C\|D^\alpha f\|_2^2 \end{aligned}$$

There remains to prove the existence of Ψ. We need only prove this for $N = 1$ and then use induction. Put $\Psi(x) = \int_{-\infty}^x \psi(x)dx = -\int_x^\infty \psi(x)dx$. Then clearly Ψ vanishes outside a bounded interval, $\Psi' = \psi$. Finally, $\int \Psi(x)dx = -\int x\Psi'(x)dx = -\int x\psi(x)dx = 0$ by a partial integration, which now makes it possible use an induction argument.

□

The general result with the same wavelet ψ is ($1 < p < \infty, D^\alpha f \in L^p, 0 \leq \alpha \leq N$)

$$\|D_f^\alpha\|_p \sim \|(\sum_k |2^{j\alpha} w_{j,k}\, 2^{j/2}\chi_{[0,1]}(2^j \cdot -k)|^2)^{1/2}\|_p$$

where $w_{j,k} = \langle f, \psi_{j,k}\rangle$.

This is connected with the classical Paley-Littlewood decomposition ($f \in L^p$, $1 < p < \infty$)

$$\|f\|_p \sim \|(\sum_j |\mu_j * f|^2)^{1/2}\|_p$$

where, *e.g.*, $\hat{\mu}_j(\omega) = \hat{\mu}(2^{-j}\omega) \geq 0, \hat{\mu}$ is infinitely differentiable, and vanishes outside the set $\{\omega; \pi/2 \leq |\omega| \leq 2\pi\}$ and $\sum_j \hat{\mu}(2^{-j}\omega) = 1$ for $\omega \neq 0$.

There are also results showing that certain wavelet bases are *unconditional* in L^p, $1 < p < \infty$, in the Hardy space H^1 - replacing L^1 - and in the space BMO (functions of bounded mean oscillation) - replacing L^∞.

The case L^2 is treated in this chapter: the Parseval relation shows that the wavelet bases are then unconditional, since the wavelet coefficients enter

with their absolute values only. This means that the (L^2-)convergence of the wavelet expansion is unconditional in the sense that it will not depend on, for example, the order of summation or the signs (phases) of the coefficients. This is in contrast to, for example, the basis ${e^{in\omega t}}_n$ for $L^p(0, 2\pi)$, $p \neq 2$, which is not unconditional.

Exercises 12.1

12.1. Formulate and prove the analogue in $\mathbb{R}$ of the identity 12.1.

12.2. Show that for $f(x) = \sum_k w_{0,k}\psi(x-k)$ with ψ as in Theorem 12.1 holds

$$\|D^\alpha f\|_p \sim \|f\|_p \quad (0 \leq \alpha \leq N)$$

using the Hölder inequality

$$\left| \int_{-\infty}^{\infty} f(t)\overline{g(t)}\, dt \right| \leq \|f\|_p \, \|g\|_q \quad (1/p + 1/q = 1,\ 1 \leq p \leq \infty)$$

where the Cauchy-Schwarz inequality was applied for $p = 2$, and writing out $w_{0,k}$.

12.3. Relate the Bernstein-Jackson lemma to the following approximation statement in Section 4.7 of Chapter 4.

$$\|f - f_j\|_2 \leq C\, 2^{-j\alpha} \, \|D^\alpha f\|_2$$

where f_j is the projection on V_j, the closed linear span of the functions $\{\varphi_{j,k}\}_k$.

12.2 Notes

For further material, we refer to the books by Meyer [23], Kahane & Lemarié [21], Hernández & Weiss [16]. Wavelets and local regularity are treated in a recent dissertation by Andersson [2], which also contains many references.

Chapter 13

An Analysis Tool

We will give two examples of how the continuous wavelet transform in Chapter 7 may be used in signal processing, and also indicate an algorithmic shortcut if the wavelet is associated with a multiresolution analysis; see Chapter 4.

13.1 Two Examples

Recall the definition of the continuous wavelet transform

$$W_\psi f(a,b) = \int_{-\infty}^{\infty} f(t)\psi\left(\frac{t-b}{a}\right)|a|^{-1/2}\,dt$$

where $f \in L^1,\ \psi \in L^1 \cap L^2,\ a \neq 0$; ψ is real-valued for simplicity, and fulfills

$$\int_0^\infty |\hat{\psi}(\xi)|^2\,d\xi/\xi \ < \ \infty$$

Example 13.1. In this example, we show the continuous wavelet transform of a chirp. The *complex* Morlet wavelet can be viewed in Figure 13.1:

$$\begin{aligned} \psi(t) &= e^{i\omega_0 t}e^{-t^2/2} \\ \widehat{\psi}(\omega) &= (2\pi)^{1/2}e^{-(\omega-\omega_0)^2/2} \end{aligned}$$

where now $\widehat{\psi}(0) \approx 0$ only, and $\omega_0 = 5.336$. This choice of ω_0 makes the real part of ψ, Reψ, have its first maximum value outside the origin as half the the modulus value there. In order to make $\widehat{\psi}(0) = 0$, a small correction may be added, *e.g.*, $-(2\pi)^{1/2}\exp(-\omega^2/2 - \omega_0^2/2)$.)

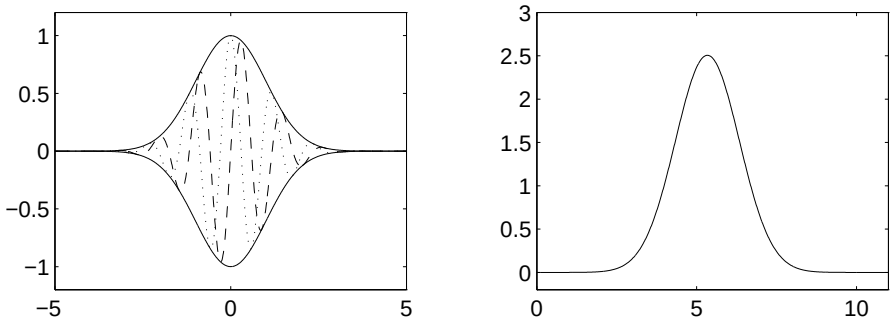

Figure 13.1: Left: Re$\psi(t)$ (dotted), Im$\psi(t)$ (dashed). Right: $\widehat{\psi}(\omega)$.

A chirp is chosen as the function $\cos(300t^3)\exp(-4t^2)$. It is shown together with its transform[1]

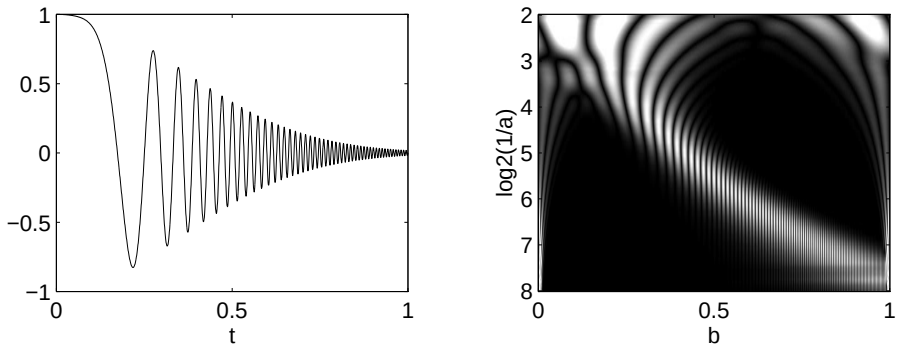

Figure 13.2: Left: $\cos(300t^3)\exp(-4t^2)$. Right: The transform.

□

Example 13.2. In Figures 13.3, 13.4, 13.5,[2] the continuous wavelet transform of three different signals are shown, which represent measurements in the velocity field of a fluid. The wavelet is a *complex* Morlet wavelet shown in Figure 13.1. The figures are to be viewed rotated a quarter turn clockwise relative to their captions. The signals are shown at the top (*sic!*) for easy reference. On the horizontal axis is the parameter b and on the vertical a, which may be viewed as time and frequency respectively (but are certainly not that exactly).

[1] The transform was calculated by MATLAB, using the Morlet wavelet (Figure 13.2) in WAVELAB.

[2] These three figures have been produced by C-F Stein, using a program written and put to his disposal by M. Holschneider.

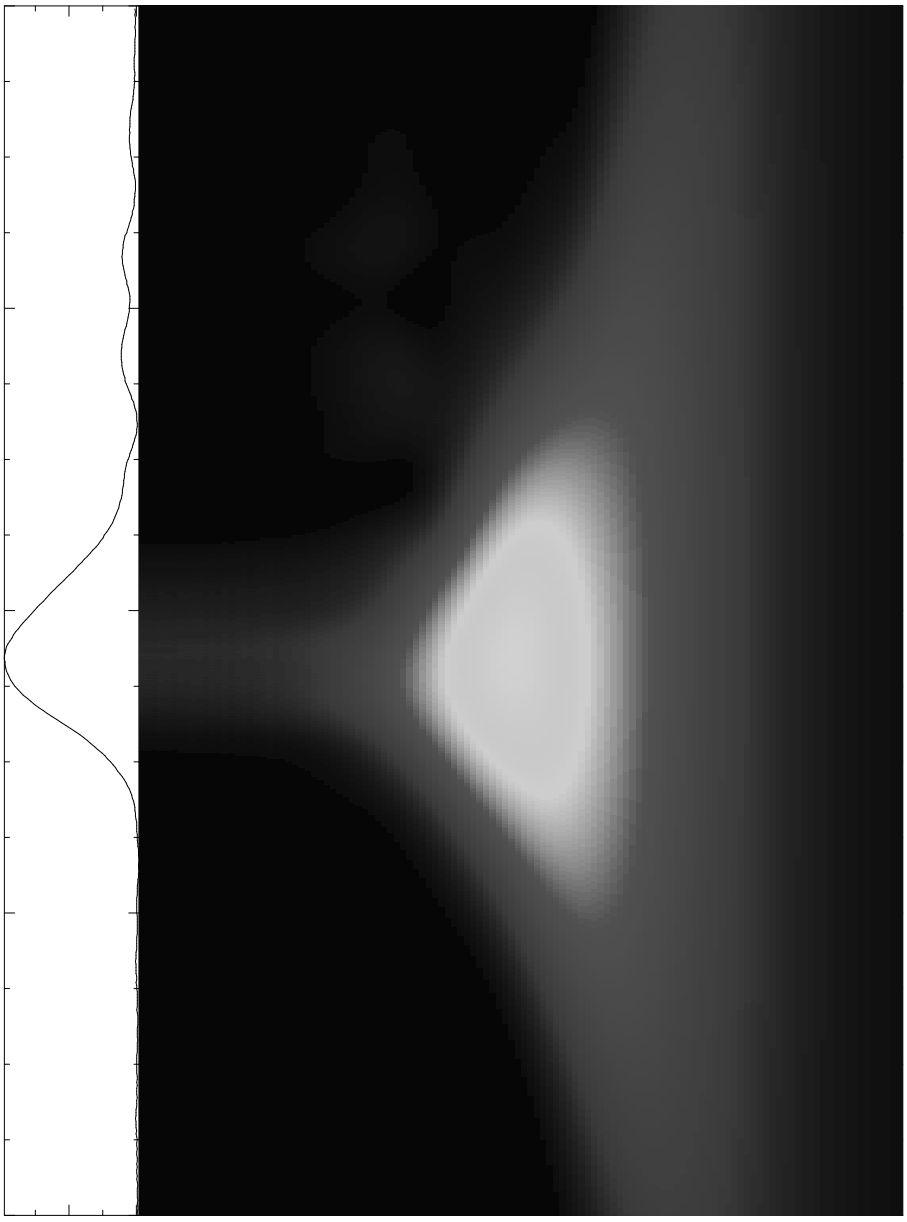

Figure 13.3: A streaky structure.

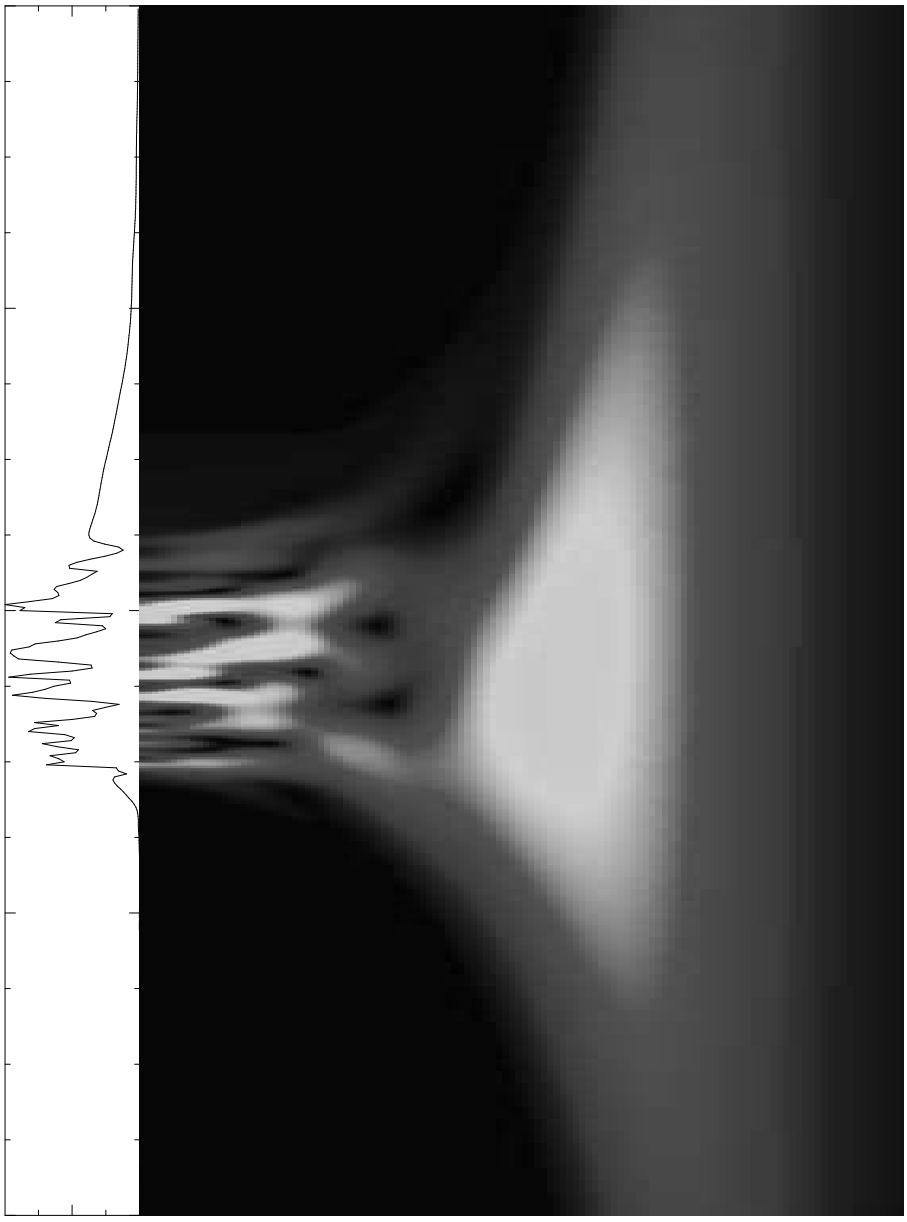

Figure 13.4: A turbulent spot.

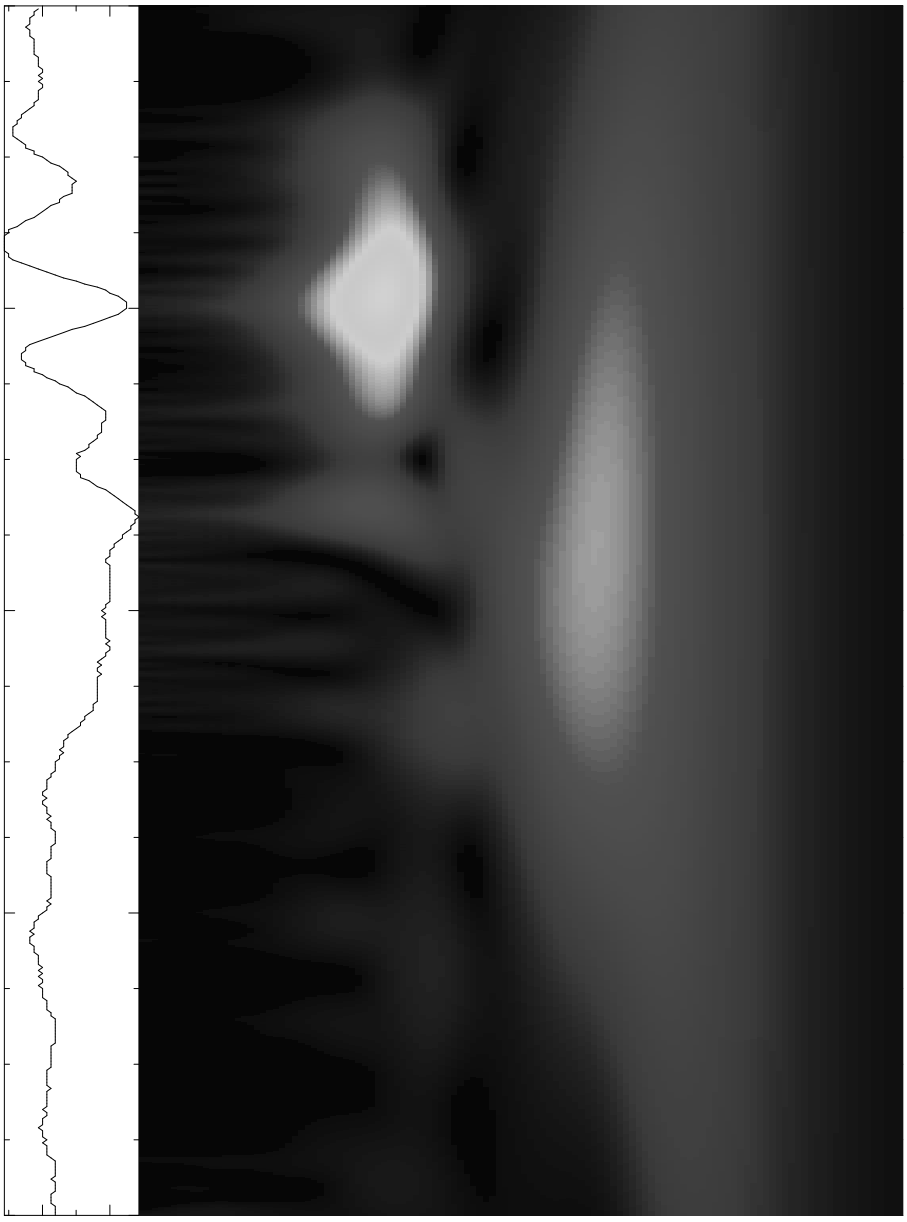

Figure 13.5: A wave packet.

□

13.2 A Numerical Sometime Shortcut

When the continuous wavelet transform

$$W_\psi f(a,b) = \int_{-\infty}^{\infty} f(t)\psi\left(\frac{t-b}{a}\right)|a|^{-1/2}\,dt$$

is implemented numerically, this involves usually the Fast Fourier Transform algorithm. However, if the wavelet ψ is associated with an MRA (Chapter 4), there is a possibility to use this structure as follows.

Note that, for $a = 2^{-j},\ b = 2^{-j}k$

$$w_{j,k} := W_\psi f(2^{-j}, 2^{-j}k) = \int_{-\infty}^{\infty} f(t)2^{j/2}\psi(2^j t - k)\,dt$$

i.e., these are the wavelet coefficients defined in Chapter 4.

Thus, starting from a calculation of the projection on the finest scales of interest, the space V_0, say, of the MRA, we may proceed as with the discrete wavelet algorithm in Chapter 4. This projection is then

$$s_{0,k} := \int_{-\infty}^{\infty} f(t)\phi(t-k)\,dt$$

13.3 Notes

A recent application of the continuous wavelet transform to the analysis of data concerning the polar motion, the Chandler wobble, can be found in Gibert *et al.* [14].

For further reading, we suggest the books by Holschneider [18], and Kahane & Lemarié [21].

Chapter 14

Feature Extraction

For most signal and image classification problems the dimension of the input signal is very large. For example, a typical segment of an audio signal has a few thousand samples, and a common image size is 512×512 pixels. This makes it practically impossible to directly employ a traditional classifier, such as linear discriminant analysis (LDA), on the input signal. Therefore one usually divides a classifier into two parts. First, one maps the input signal into some lower dimensional space containing the most relevant features of the different classes of possible input signals. Then, these features are fed into a traditional classifier such as an LDA or an artificial neural net (ANN). Most literature on classification focuses on the properties of the second step. We will present a rather recent wavelet based technique for the first step, that is, the feature extractor. This technique expands the input signal into a large time-frequency dictionary consisting of, for instance, wavelet packets and local trigonometric bases. It then finds, using the best-basis algorithm, the one basis that best discriminates the different classes of input signals. We refer to these bases as local discriminant bases.

14.1 The Classifier

To begin with we set up some notation. Let $\mathcal{X}$ be the set of all input signals, and let $\mathcal{Y}$ be the corresponding set of class labels. To simplify matters, we further assume that we only have two classes, $\mathcal{Y} = \{1, 2\}$. The input signal space $\mathcal{X} \subset \mathbb{R}^n$, and as we already mentioned n is typically of the order of 10^3-10^6. A function $d : \mathcal{X} \to \mathcal{Y}$, which assigns a class label to each input signal $x \in \mathcal{X}$, is called a classifier. We also assume we have access to a training set consisting of N pairs of input signals and corresponding class labels, $\{(x_1, y_1), \ldots, (x_N, y_N)\}$. Let $N = N_1 + N_2$, where N_1 and N_2 are the

number of class 1 and 2 signals, respectively. We use the notation $x_i^{(y)}$ to denote that a signal in the training set belongs to class y. Let $P(A, y)$ be a probability distribution on $\mathcal{X} \times \mathcal{Y}$, where $A \subset \mathcal{X}$ and $y \in \mathcal{Y}$. This means that,

$$P(A, y) = P(X \in A, Y = y) = \pi_y P(X \in A | Y = y),$$

where $X \in \mathcal{X}$ and $Y \in \mathcal{Y}$ are random variables. Here π_y is the prior probability of class y, which is usually estimated as $\pi_y = N_y/N$, $y = 1, 2$. The optimal classifier for this set up is the Bayes classifier. To obtain it we need an estimate of $P(A, y)$. The number of training samples is small compared to the size of the input signals though, that is, $N \ll n$. This makes it impossible to reliably estimate $P(A, y)$ in practice.

Feature extraction is the resolution to the problem with the high dimensionality of the input signal space. It is essential to extract relevant features of the signal for classification purposes. In practice, it is known that multivariate data in $\mathbb{R}^n$ are almost never n-dimensional. Rather, the data exhibit an intrinsic lower dimensional structure. Hence, the following approach of splitting the classifier into two functions is often taken:

$$d = g \circ f.$$

Here, $f : \mathcal{X} \to \mathcal{F}$ is a feature extractor mapping the input signal space into a lower dimensional feature space $\mathcal{F} \subset \mathbb{R}^m$. The dimension m of the feature space is typically at least ten times smaller than the dimension n of the input signal space. The feature extractor is followed by a traditional classifier $g : \mathcal{F} \to \mathcal{Y}$, which should work well if the different signal classes are well separated in the feature space. We will describe an automatic procedure for constructing the feature extractor given the training data.

14.2 Local Discriminant Bases

Let $\mathcal{D}$ be a time-frequency dictionary consisting of a collection of basis vectors $\{\psi_j\}_{j=1}^M$. If, for instance, this dictionary consists of all the basis vectors in a given wavelet packet basis then $M = n(1 + \log_2 n)$. The dictionary with its basis vectors contains a huge number ($> 2^n$) of orthonormal bases of $\mathbb{R}^n$, see Chapter 9. A wavelet packet dictionary should be well suited for signal classification purposes since discriminant features are probably separated in the time-frequency domain, but not necessarily in the time (or space) domain. Given a signal and a cost function, we can expand the signal into an optimal basis using the best-basis algorithm. For signal and image compression applications the cost function could be the entropy of the expansion

coefficients for a given basis. Our objective is to separate different classes of input signals, and we will use other cost functions to discriminate between the characteristic features of the different classes. For classification purposes, we prefer to use the term discriminate function rather than cost function, and the dictionary of bases is called local discriminant bases (LDB).

Before we discuss how to choose the cost function to discriminate between different bases,let us suppose that we have already computed the best basis. We then form an orthonormal $n \times n$ matrix

$$\mathcal{W} = \begin{pmatrix} \psi_1 & \dots & \psi_n \end{pmatrix},$$

where $\psi_1, \dots, \psi_n$ are the basis vectors of the best basis. Applying the transpose of this matrix to an input signal x, gives us the coordinates of the signal in the best basis (we regard x and the ψ_i:s as column vectors in $\mathbb{R}^n$)

$$\mathcal{W}^{\mathrm{T}} x = \begin{pmatrix} {\psi_1}^{\mathrm{T}} x \\ \vdots \\ {\psi_n}^{\mathrm{T}} x \end{pmatrix} = \begin{pmatrix} \langle x, \psi_1 \rangle \\ \vdots \\ \langle x, \psi_n \rangle \end{pmatrix}.$$

The LDB classifier is defined as

$$d = g \circ (P_m \mathcal{W}^{\mathrm{T}}),$$

where the $m \times n$ matrix P_m is a feature selector, which selects the most important m ($< n$) features (coordinates) in the best basis.

14.3 Discriminant Measures

The dictionary $\mathcal{D}$ of LDBs can also be expressed as the set of all possible orthonormal bases that can be constructed from the basis functions $\{\psi_j\}_{j=1}^{M}$. Let B be such a basis and let $\Delta(B)$ be a discriminant function measuring the ability of the basis B to separate (discriminate) the different classes of input signals. Then, the optimal basis W is chosen as the one that maximizes the discriminant function

$$W = \operatorname*{argmax}_{B \in \mathcal{D}} \Delta(B).$$

To define Δ we start by defining a discriminant measure for each unit basis vector $\psi_i \in \mathcal{D}$. Let $X \in \mathcal{X}$ be a random sample from the input signal space, and let

$$Z_i = \langle X, \psi_i \rangle.$$

Then Z_i is also a random variable, and we sometimes write $Z_i^{(y)}$ to emphasize that the random variable X is of class y. Now, we are interested in the probability density function (pdf) of Z_i for each class y, which we denote as $q_i^{(y)}(z)$. We can estimate these pdfs by expanding the available signals in the training set into the basis functions of the dictionary. An estimate $\widehat{q}_i^{(y)}$, for $q_i^{(y)}$, can then be computed by a pdf estimation technique called averaged shifted histograms (ASH), for example.

Once we have estimated the pdfs $\widehat{q}_i^{(1)}$ and $\widehat{q}_i^{(2)}$, for class 1 and 2, we need a distance function $\delta(\widehat{q}_i^{(1)}, \widehat{q}_i^{(2)})$, that measures the ability of the direction ψ_i to separate the two classes. If the two pdfs are similar δ should be close to zero. The best direction is the one for which the two pdfs look most different from one another. For this direction δ should attain a maximum positive value. There are several ways to measure the discrepancy between two pdfs of which we mention

$$\delta(p,q) = \int (\sqrt{p(z)} - \sqrt{q(z)})^2 \, dz \qquad \text{(Hellinger distance)}$$

$$\delta(p,q) = \left(\int (p(z) - q(z))^2 \, dz \right)^{1/2} \qquad (\ell^2\text{-distance})$$

The choice of the distance function is, of course, problem dependent.

When we now know how to construct a discriminant function for a single basis vector, we must construct one for a complete basis $B \in \mathcal{D}$. Suppose $B = \begin{pmatrix} \psi_1 & \dots & \psi_n \end{pmatrix}$, and define the discriminant powers δ_i for direction ψ_i as

$$\delta_i = \delta(\widehat{q}_i^{(1)}, \widehat{q}_i^{(2)}).$$

Further let $\{\delta_{(i)}\}$ be the decreasing rearrangement of $\{\delta_i\}$, that is, the discriminant powers sorted in decreasing order. The discriminant function for the basis B is then finally defined as the sum of the k ($< n$) largest discriminant powers

$$\Delta(B) = \sum_{i=1}^{k} \delta_{(i)}.$$

One possibility could be to let $k = m$, the dimension of the feature space. This is not necessary though, and the choice of k needs further research.

14.4 The LDB Algorithm

We conclude the discussion above by formulating the following algorithm.

Algorithm. (**The LDB Algorithm**)

1. Expand all the signals in the training set into the time-frequency dictionary $\mathcal{D}$.
2. Estimate the projected pdfs $\widehat{q}_i^{(y)}$ for each basis vector ψ_i and class y.
3. Compute the discriminant power δ_i of each basis vector ψ_i in the dictionary.
4. Choose the best basis as $W = \underset{B \in \mathcal{D}}{\operatorname{argmax}}\, \Delta(B)$, where $\Delta(B) = \sum_{i=1}^{k} \delta_{(i)}$ is the sum of the k largest discriminant powers of the basis B.
5. Define the matrix P_m so that it selects the m largest discriminant powers.
6. Design the classifier g on the m features from the previous step.

All of these steps take no more than $\mathcal{O}(n \log n)$ operations (see Chapter 9), making this algorithm computationally efficient.

14.5 Notes

Linear discriminant bases for classification purposes was introduced by Saito and Coifman in the paper *Local discriminant bases* [25]. Improvements to the LDB method were later described by the same authors in [26]. They have performed successful experiments on geophysical acoustic waveform classification, radar signal classification, and classification of neuron firing patterns of monkeys.

Chapter 15

Implementation Issues

This chapter is concerned with two issues related to the actual implementation of the discrete wavelet transform. The first concerns finite length signals and how to extend, or otherwise treat, them. The second, how to process the sample values of a continuous-time function, before the discrete wavelet transform is being implemented in a filter bank. (The discrete wavelet transform was defined in Chapter 4.)

15.1 Finite Length Signals

In applications we always have signals of finite length. So far, we have assumed that the signals extended indefinitely. Let us now start with a signal of finite length L, with sample values $x_0, x_1, \ldots, x_{L-1}$. Proper handling of finite length signals is especially important in image processing applications, where poor treatment of the boundaries clearly becomes visible in, for instance, a compressed image. We will describe how to handle finite length in one dimension, and for images we simply do the same for one column or row at a time.

The problem lies in the computation of convolutions. The sum $\sum_k h_k x_{n-k}$ normally involves, for instance, x_{-1}, which is not defined. We can deal with this problem in two ways. First, we can extend, in some way, the signal beyond the boundaries. The other possibility is to change to special boundary-corrected filters near the end-points of the signal. In both cases there exists several alternatives, of which we will present the most important. We start with different extension techniques, and then we go on with boundary-corrected filters.

Extension of Signals

We will describe three extension methods: extension by zeros (zero-padding), extension by periodicity (wraparound), and extension by reflection (symmetric extension). See Figure 15.1.

Zero-padding simply sets the rest of the values to zero, giving the infinite sequence

$$\ldots, 0, 0, x_0, x_1, \ldots, x_{L-2}, x_{L-1}, 0, 0, \ldots.$$

Unless $x_0 = x_{L-1} = 0$, zero-padding introduces a jump in the function. Zero-padding of a continuous function defined on $(0, L)$ would in general produce a discontinuous function. Therefore, zero-padding is, in most cases, not used.

Wraparound creates a periodic signal with period L, by defining $x_{-1} = x_{L-1}$, $x_L = x_0$, etc. We get

$$\ldots, x_{L-2}, x_{L-1}, x_0, x_1, \ldots, x_{L-2}, x_{L-1}, x_0, x_1, \ldots.$$

For signals that are naturally periodic, wraparound is better than zero-padding. The discrete Fourier transform (DFT) of a vector in $\mathbb{R}^L$ is the convolution, of the periodic extension, of the vector with the filter $1, W^1, \ldots, W^{L-1}$, where $W = e^{-i2\pi/L}$.

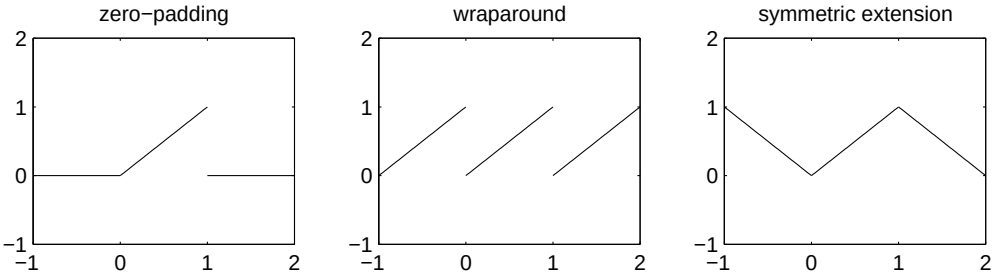

Figure 15.1: Extension methods

There are two ways to extend a discrete-time signal symmetrically – either we repeat the first and last sample, or we do not.

If we do not repeat, it corresponds to the extension $f(-t) = f(t)$ in continuous time. In discrete time we take $x_{-1} = x_1$, $x_L = x_{L-2}$, and so on, to get

$$\ldots, x_2, x_1, x_0, x_1, \ldots, x_{L-2}, x_{L-1}, x_{L-2}, x_{L-3}, \ldots.$$

We call this whole-point symmetry since the point of symmetry is around $t = 0$. The extended signal has period $2L - 1$.

Half-point symmetry is the other possible symmetric extension method. It does repeat the first and last sample value, and its point of symmetry is around $t = 1/2$. We let $x_{-1} = x_0$, and $x_L = x_{L-1}$, etc. We then get a signal with period $2L$,

$$\ldots, x_1, x_0, x_0, x_1, \ldots, x_{L-2}, x_{L-1}, x_{L-1}, x_{L-2}, \ldots.$$

The two symmetric extension techniques give rise to two versions of the discrete cosine transform (DCT). The DCT is used in the JPEG image compression standard, for example.

In continuous time, symmetric extension gives a continuous function. This is the advantage. It does introduce a jump in the first derivative, though.

Symmetric Extension and Filters

From now on, let us suppose that we have extended the signal using whole- or half-point symmetry. Then, we are faced with another problem. The outputs of the analysis part of the filter bank have been downsampled, after the filtering. We want these output signals to be symmetric as well, since the filter bank is repeated on the lowpass output in the discrete wavelet transform. How do we achieve this?

For a biorthogonal filter bank, with symmetric (or antisymmetric) filters, we use whole-point symmetric extension for odd-length filters, and half-point symmetric extension for even-length filters.

We refer to the books and articles in the Notes section for further details about extension and its practical implementation.

Boundary-Corrected Filters

Boundary-corrected filters are quite complicated to construct. In continuous time this corresponds to defining a wavelet basis for a finite interval $[0, L]$. Such construction are still at the research stage. Let us therefore describe one example of boundary-corrected wavelets.

The example we will consider is the piecewise linear hat function as the scaling function. It is supported on two intervals and the corresponding wavelet on three intervals, see Figure 15.2. These are the synthesis functions. The filters are 0.5, 1, 0.5, and 0.1, -0.6, 1, -0.6, 0.1, respectively. This is an example of a semi-orthogonal wavelet basis. The approximation space V_0 is orthogonal to the detail space W_0, but the basis functions within those spaces are not orthogonal; the hat function is not orthogonal to its translates. This basis is useful when discretizing certain differential equations, using the

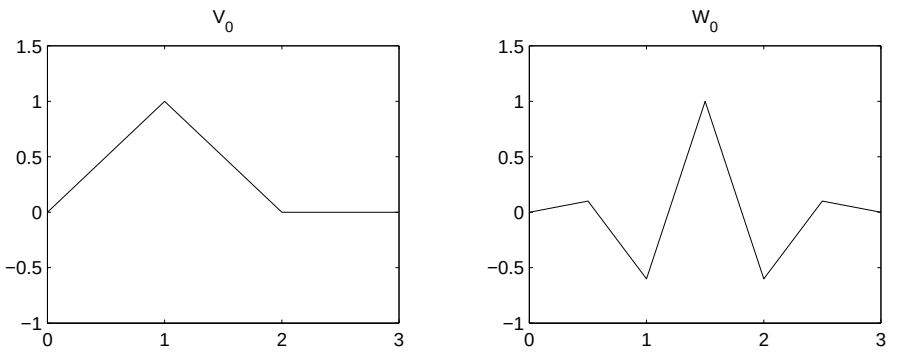

Figure 15.2: Piecewise linear scaling function and wavelet.

Galerkin method. Then, one does not need to know the dual scaling function and wavelet. See Chapter 11 for details on how to solve differential equations using wavelets.

Finally, in Figure 15.3, we see two boundary-corrected wavelets. Depending on whether we have Dirichlet or Neumann conditions at the boundary of the differential equation, we may want to force the boundary condition to 0.

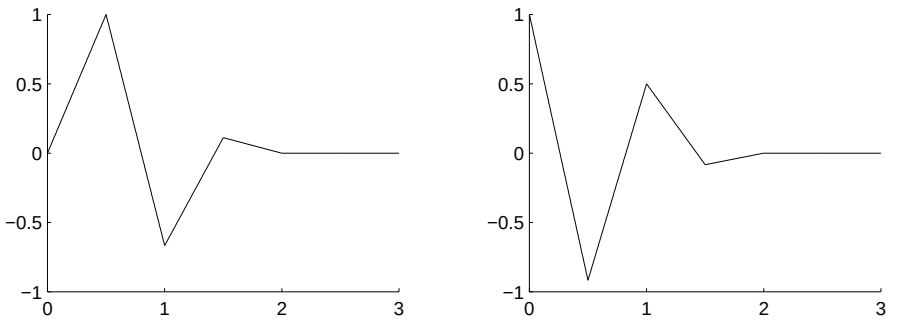

Figure 15.3: Boundary corrected wavelets.

15.2 Pre- and Post-Filtering

When implementing the discrete wavelet transform we need the scaling coefficients $s_{J,k}$, at some fine scale J. The problem is this: we are normally given the sample values $f(2^{-J}k)$ of a continuous-time function $f(t)$, and these are not equal to $s_{J,k}$. It is a common mistake to believe so! We will now give some suggestions on how to handle this.

Pre-filtering

Suppose we are given the sample values $f(2^{-J}k)$ of a lowpass filtered signal f, with $\hat{f}(\omega) = 0$ for $|\omega| \geq 2^J\pi$. These sample values will be related to the approximation of f at the scale 2^{-J}, through the projection

$$f_J(t) := P_J f = \sum_k \langle f, \varphi_{J,k}\rangle \varphi_{J,k}(t).$$

A relation appears if we compute the scaling coefficients $s_{J,k} = \langle f, \varphi_{J,k}\rangle$ approximately with some numerical integration method, *e.g.*, using a rectangle approximation

$$\begin{aligned} s_{J,k} &= \int_{-\infty}^{\infty} f(t)\varphi_{J,k}(t)\,dt \\ &\approx 2^{-J}\sum_l f(2^{-J}l)\varphi_{J,k}(2^{-J}l) \\ &= 2^{-J/2}\sum_l f(2^{-J}l)\varphi(l-k). \end{aligned}$$

Note that the last expression is a filtering of the samples of f, where the filter coefficients are $2^{-J/2}\varphi(-l)$. This is called *pre-filtering*. There exists other pre-filtering methods. If, for instance, f is band-limited, this can be taken into account, to compute the scaling coefficients $s_{J,k}$ more accurately. It is common practice to use the sample values directly as the scaling coefficients, which then introduces an error. This error will have its main influence at the smallest scales, that is, for $j < J$ close to J (see Exercise 15.1).

Post-filtering

The sample values can be reconstructed approximately, through a filtering of the scaling coefficients $s_{J,k}$ with the filter coefficients $2^{J/2}\varphi(k)$

$$\begin{aligned} f(2^{-J}k) &\approx f_J(2^{-J}k) \\ &= \sum_l s_{J,l}\varphi_{J,l}(2^{-J}k) \\ &= 2^{J/2}\sum_l s_{J,l}\varphi(k-l). \end{aligned}$$

This is a convolution and it is called the *post-filtering* step.

Exercises 15.2

15.1. Suppose that we are given the sample values $f(2^{-J}k)$ of a lowpass filtered signal f, with $\widehat{f}(\omega) = 0$ for $|\omega| \geq 2^J\pi$. For ω in the pass-band, verify that

$$\widehat{f}(\omega) = 2^{-J} \sum_k f(2^{-J}k) e^{-i2^{-J}k\omega},$$

and that the Fourier transform of the function

$$f_J(t) = \sum_k f(2^{-J}k)\varphi_{J,k}(t)$$

equals

$$\widehat{f}_J(\omega) = 2^{-J/2}\widehat{\varphi}(2^{-J}\omega) \sum_k f(2^{-J}k) e^{-i2^{-J}k\omega}.$$

This indicates how a filter might be constructed to compensate for the influence of the scaling function.

15.3 Notes

The overview article of Jawerth and Sweldens [20] describes how to define orthogonal wavelets on an interval. The book [27] by Nguyen and Strang discusses finite length signals and also contains useful references for further reading.

Bibliography

[1] P. Andersson, *Characterization of pointwise hölder regularity*, Appl. Comput. Harmon. Anal. **4** (1997), 429–443.

[2] _____, *Wavelets and local regularity*, Ph.D. thesis, Chalmers University of Technology and Göteborg University, 1997.

[3] G. Beylkin, *Wavelets and fast numerical algorithms*, Lecture Notes for short course, AMS-93, Proceedings of Symposia in Applied Mathematics, vol. 47, 1993, pp. 89–117.

[4] G. Beylkin, R. Coifman, and V. Rokhlin, *Fast wavelet transforms and numerical algorithms i*, Comm. Pure and Appl. Math. **44** (1991), 141–183.

[5] W. Briggs, *A multigrid tutorial*, SIAM, 1987.

[6] A.R. Calderbank, I. Daubechies, W. Sweldens, and B-L Yeo, *Wavelet transforms that map integers to integers*, Appl. Comput. Harmon. Anal. **5** (1998), 312–369.

[7] C.K. Chui, *Introduction to wavelets*, New York: Academic Press, 1992.

[8] _____, *Wavelets: a mathematical tool for signal analysis*, Philadelphia: SIAM, 1997.

[9] A. Cohen and I. Daubechies, *Non-separable bidimensional wavelets*, Revista Matemática Iberoamericana **9** (1993), no. 1, 51–137.

[10] A. Cohen and J-M Schlenker, *Compactly supported bidimensional wavelet bases with hexagonalsymmetry*, Constr. Approx. **9** (1993), 209–236.

[11] I. Daubechies, *Ten lectures on wavelets*, SIAM, 1992.

[12] I. Daubechies and W. Sweldens, *Factoring wavelet transforms into lifting steps*, Tech. report, Bell Laboratories, Lucent Technologies, 1996.

[13] D.L. Donoho, *Nonlinear wavelet methods for recovery of signals, densities and spectra from indirect and noisy data*, Proc. Symposia in Applied Mathematics (I. Daubechies, ed.), American Mathematical Society, 1993.

[14] D. Gibert, M. Holschneider, and J.L. LeMouel, *Wavelet analysis of the chandler wobble*, J Geophys Research - Solid Earth **103** (1998), no. B11, 27069–27089.

[15] L. Greengard and V. Rokhlin, *A fast algorithm for particle simulations*, Journal of Computational Physics **73(1)** (1987), 325–348.

[16] E. Hernández and G. Weiss, *A first course on wavelets*, CRC Press, 1996.

[17] M. Hilton, B. Jawerth, and A. Sengupta, *Compressing still and moving images with wavelets*, To appear in Multimedia Systems, Vol. 2, No. 1, 1994.

[18] M. Holschneider, *Wavelets: An analysis tool*, Oxford: Clarendon Press, 1995.

[19] B.B. Hubbard, *World according to wavelets: The story of a mathematical technique in the making*, Wellesley, Mass : A K Peters, 1998.

[20] B. Jawerth and W. Sweldens, *An overview of wavelet based multiresolution analyses*, SIAM Rev. **36** (1994), no. 3, 377–412.

[21] J-P Kahane and P.G. Lemarié-Rieusset, *Fourier series and wavelets*, Gordon & Breach, 1995.

[22] S. Mallat, *A wavelet tour of signal processing*, Academic Press, 1998.

[23] Y. Meyer, *Ondelettes et opérateurs: I*, Hermann, 1990.

[24] ———, *Wavelets: Algorithms & applications*, SIAM, 1993.

[25] N. Saito and R. Coifman, *Local discriminant bases*, Mathematical Imaging: Wavelet Applications in Signal and Image Processing (A.F. Laine and M.A. Unser, eds.), vol. 2303, SPIE, 1994.

[26] ———, *Improved local discriminant bases using emperical probability estimation*, Statistical Computing, Amer. Stat. Assoc., 1996.

[27] G. Strang and T. Nguyen, *Wavelets and filter banks*, Wellesley-Cambridge Press, 1996.

[28] R. Strichartz, *Wavelets and self-affine tilings*, Constructive Approximation (1993), no. 9, 327–346.

[29] W. Sweldens and P. Schröder, *Building your own wavelets at home*, Tech. report, University of South Carolina, Katholieke Universiteit Leuven, 1995.

[30] M. Vetterli and J. Kovacevic, *Wavelets and subband coding*, Prentice Hall, 1995.

[31] M.V. Wickerhauser, *Lectures on wavelet packets algorithms*, Tech. report, Department of Mathematics, Washington University, St Louis, 1991.

Index