Part II

Wavelets and applications to image processing

Chapter 5

Wavelets

In Part I on Fourier analysis our focus was to approximate periodic functions or vectors in terms of trigonometric functions. We saw that the Discrete Fourier transform could be used to obtain the representation of a vector in terms of such functions, and the computations could be done efficiently with the FFT algorithm. This was useful for analyzing, filtering, and compression of sound and other discrete data. However, Fourier series and the DFT also have some serious limitations:

1. First of all, the functions used in the approximation are periodic with short periods. In contrast, for most functions encountered in applications the frequency content changes with time. Although Fourier analysis tools also exist for analyzing non-periodic functions, these tools mostly have a theoretical significance and are rarely used in practice, because of lack of efficient implementations.

2. Secondly, all components of the Fourier basis vectors are nonzero — in fact they all have absolute value 1 at all points. This means that, in order to compute a value using the representation in the Fourier basis, we must for each instance in time sum over all N vectors in the basis. This is time-consuming when N is large.

In this chapter we are going to introduce the basic properties of an alternative to Fourier analysis, namely wavelets. Similar to Fourier analysis, wavelets are also based on the idea of transforming a function to a different basis. But in contrast to Fourier analysis, where the basis is fixed, wavelets provide a general framework with many different types of bases. In this chapter, we introduce the framework via the simplest wavelets. We then discuss some general wavelet concepts before we consider a second example.

(a) (b)

Figure 5.1: A view of Earth from space (a) and a zoomed in view (b).

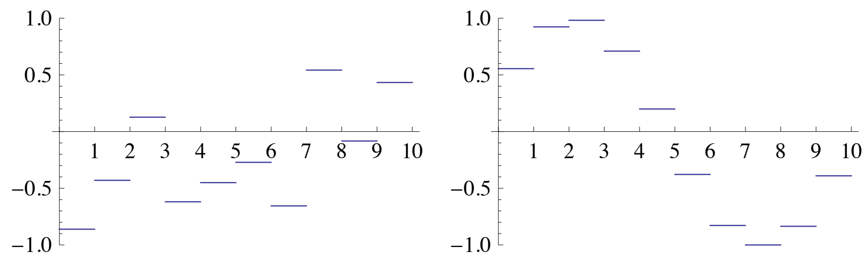
5.1 Why wavelets?

Figure 5.1 shows two views of the Earth. The one on the left is the startup image in Google Earth, a program for viewing satellite images, maps and other geographic information. The right image is a zoomed-in view of a small part of the Earth. There is clearly an amazing amount of information available behind a program like Google Earth, with images detailed enough to differentiate between buildings and even trees or cars all over the Earth. So when the Earth is spinning in the opening screen, all the Earth's buildings appear to be spinning with it! If this was the case, the Earth would not be spinning on the screen. There would just be so much information to process that a laptop would not be able to display a rotating Earth.

There is a simple reason that the globe can be shown spinning in spite of the huge amounts of information that need to be handled. We are going to see later that a digital image is just a rectangular array of numbers that represent the colour at a dense set of points. As an example, the images in Figure 5.1 are both made up of a grid with 1576 points in the horizontal direction and 1076 points in the vertical direction, for a total of 1 695 776 points. The colour at a point is represented by three eight-bit integers, which means that the image file contains a total of 5 087 328 bytes. So regardless of how close to the surface of the Earth our viewpoint is, the resulting image always contains the same number of points. This means that when we are far away from the Earth we can use a very coarse model of the geographic information that is being displayed, but as we zoom in, we need to display more details and therefore need a more accurate model.

Observation 5.1. When discrete information is displayed in an image, there is no need to use a mathematical model that contains more detail than what is visible in the image.

A consequence of Observation 5.1 is that for applications like Google Earth we should use a mathematical model that makes it easy to switch between



(a) (b)

Figure 5.2: Two examples of piecewise constant functions.

different levels of detail, or different resolutions. Such models are called *multiresolution models*, and wavelets are prominent examples of this kind of models.

5.2 Wavelets constructed from piecewise constant functions

There are many different kinds of wavelets that all share certain standard properties. In this section we will introduce the simplest wavelets and through this also the general framework for constructing wavelets. The construction goes in two steps: First we introduce the resolution spaces, and then the detail spaces and wavelets.

5.2.1 Resolution spaces

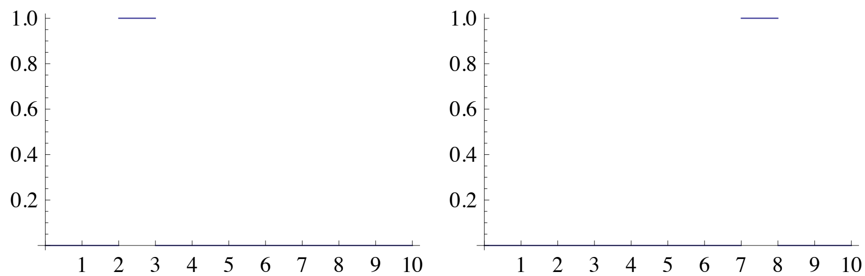
The starting point is the space of piecewise constant functions on an interval $[0, N)$.

Definition 5.2 (The resolution space V_0)**.** Let N be a natural number. The resolution space V_0 is defined as the space of functions defined on the interval $[0, N)$ that are constant on each subinterval $[n, n+1)$ for $n = 0, \ldots, N-1$.

Two examples of functions in V_0 for $N = 10$ are shown in Figure 5.2. It is easy to check that V_0 is a linear space, and for computations it is useful to know the dimension of the space and have a basis.

Lemma 5.3. Define the function $\phi(t)$ by

$$\phi(t) = \begin{cases} 1, & \text{if } 0 \leq t < 1; \\ 0, & \text{otherwise}; \end{cases} \tag{5.1}$$



(a) (b)

Figure 5.3: The functions ϕ_2 (a) and ϕ_7 (b) in V_0.

and set $\phi_n(t) = \phi(t-n)$ for any integer i. The space V_0 has dimension N, and the N functions $\{\phi_n\}_{n=0}^{N-1}$ form an orthonormal basis for V_0 with respect to the standard inner product

$$\langle f, g\rangle = \int_0^N f(t)g(t)\,dt. \tag{5.2}$$

In particular, any $f \in V_0$ can be represented as

$$f(t) = \sum_{n=0}^{N-1} c_n \phi_n(t) \tag{5.3}$$

for suitable coefficients $(c_i)_{i=0}^{N-1}$. The function ϕ_n is referred to as the *characteristic* function of the interval $[n, n+1)$

Two examples of the basis functions defined in Lemma 5.5 are shown in Figure 5.3.

Proof. Two functions ϕ_{n_1} and ϕ_{n_2} with $n_1 \neq n_2$ clearly satisfy $\int \phi_{n_1}(t)\phi_{n_2}(t)dt = 0$ since $\phi_{n_1}(t)\phi_{n_2}(t) = 0$ for all values of x. It is also easy to check that $\|\phi_n\| = 1$ for all n. Finally, any function in V_0 can be written as a linear combination the functions ϕ_0, ϕ_1, ..., ϕ_{N-1}, so the conclusion of the lemma follows. □

In our discussion of Fourier analysis, the starting point was the function $\sin 2\pi t$ that has frequency 1. We can think of the space V_0 as being analogous to this function: The function $\sum_{n=0}^{N-1}(-1)^n\phi_n(t)$ is (part of the) square wave that we discussed in Chapter 1, and which also oscillates regularly like the sine function, see Figure 5.4 (a). The difference is that we have more flexibility since we have a whole space at our disposal instead of just one function — Figure 5.4 (b) shows another function in V_0.

In Fourier analysis we obtained a linear space of possible approximations by including sines of frequency 1, 2, 3, ..., up to some maximum. We use a similar

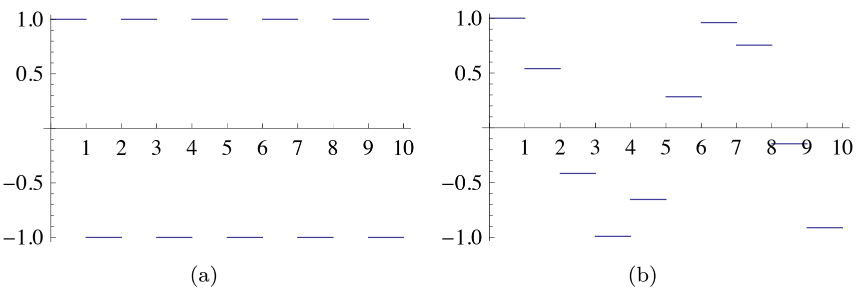


Figure 5.4: The square wave in V_0 (a) and an approximation to $\cos t$ from V_0.

approach for constructing wavelets, but we double the frequency each time and label the spaces as V_0, V_1, V_2, ...

Definition 5.4 (Refined resolution spaces)**.** The space V_m for the interval $[0, N)$ is the space of piecewise linear functions defined on $[0, N)$ that are constant on each subinterval $[n/2^m, (n+1)/2^m)$ for $n = 0, 1, \ldots, 2^m N - 1$.

Some examples of functions in the spaces V_1, V_2 and V_3 for the interval $[0, 10]$ are shown in Figure 5.5. As m increases, we can represent smaller details. In particular, the function in (d) is a piecewise constant function that oscillates like $\sin 2\pi 2^2 t$ on the interval $[0, 10]$.

It is easy to find a basis for V_m, we just use the characteristic functions of each subinterval.

Lemma 5.5. Let $[0, N)$ be a given interval with N some positive integer, and let V_m denote the resolution space of piecewise constant functions for some integer $m \geq 0$. Then the dimension of V_m is $2^m N$. Define the functions

$$\phi_{m,n}(t) = 2^{m/2}\phi(2^m t - n), \quad \text{for } n = 0, 1, \ldots, 2^m N - 1, \tag{5.4}$$

where ϕ is the characteristic function of the interval $[0, 1]$. The functions $\{\phi_{m,n}\}_{n=0}^{2^m N-1}$ form an orthonormal basis for V_m, and any function $f \in V_m$ can be represented as

$$f(t) = \sum_{n=0}^{2^m N-1} c_n \phi_{m,n}(t)$$

for suitable coefficients $(c_n)_{n=0}^{2^m N-1}$.

Proof. The functions given in (5.25) are exactly the characteristic functions of the subintervals $[n/2^m, (n+1)2^m)$ which we referred to in Definition 5.4, so the proof is very similar to the proof of Lemma 5.5. The one mysterious thing may

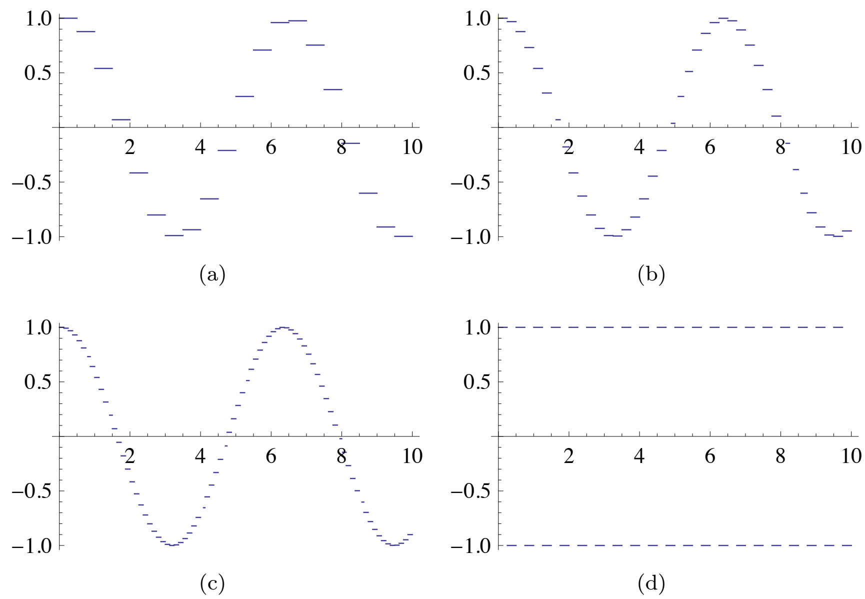


Figure 5.5: Piecewise constant approximations to $\cos t$ on the interval $[0, 10]$ in the spaces V_1 (a), V_2 (b), and V_3 (c). The plot in (d) shows the square wave in V_2.

be the normalisation factor $2^{-m/2}$. This comes from the fact that

$$\int_0^N \phi(2^m t - n)^2\,dt = \int_{n/2^m}^{(n+1)/2^m} \phi(2^m t - n)^2\,dt = 2^{-m}\int_0^1 \phi(u)^2\,du = 2^{-m}.$$

The normalisation therefore ensures that $\|\phi_{m,n}\| = 1$. □

In the theory of wavelets, the function ϕ is also called a *scaling function*. The origin behind this name is that the scaled (and translated) functions $\phi_{m,n}$ of ϕ are used as basis functions for the refined resolution spaces. Later on we will see that other scaling functions ϕ can be chosen, where the scaled versions $\phi_{m,n}$ will be used to define similar resolution spaces, with slightly different properties.

5.2.2 Function approximation property

Each time m is increased by 1, the dimension of V_m doubles, and the subinterval on which the functions in V_m are constant are halved in size. It therefore seems reasonable that, for most functions, we can find good approximations in V_m provided m is big enough.

Theorem 5.6. Let f be a given function that is continuous on the interval $[0, N]$. Given $\epsilon > 0$, there exists an integer $m \geq 0$ and a function $g \in V_m$ such that

$$\big|f(t) - g(t)\big| \leq \epsilon$$

for all t in $[0, N]$.

Proof. Since f is (uniformly) continuous on $[0, N]$, we can find an integer m so that $\big|f(t_1) - f(t_2)\big| \leq \epsilon$ for any two numbers t_1 and t_2 in $[0, N]$ with $|t_1 - t_2| \leq 2^{-m}$. Define the approximation g by

$$g(t) = \sum_{n=0}^{2^m N-1} f\big(t_{m,n+1/2}\big)\phi_{m,n}(t),$$

where $t_{m,n+1/2}$ is the midpoint of the subinterval $\big[n2^{-m}, (n+1)2^{-m}\big)$,

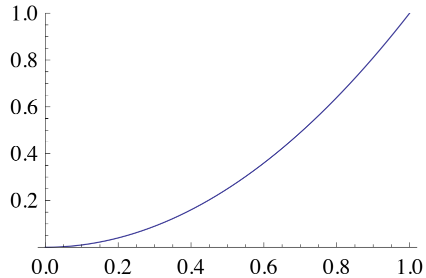
$$t_{m,n+1/2} = (n + 1/2)2^{-m}.$$

For t in this subinterval we then obviously have $|f(t) - g(t)| \leq \epsilon$, and since these intervals cover $[0, N]$, the conclusion holds for all $t \in [0, N]$. □

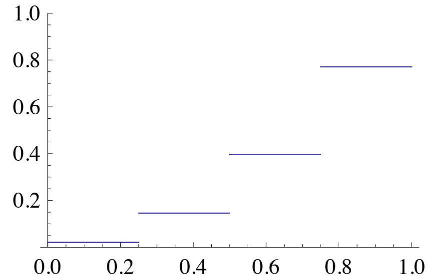
Theorem 5.6 does not tell us how to find the approximation g although the proof makes use of an approximation that interpolates f at the midpoint of each subinterval. Note that if we measure the error in the L^2-norm, we have

$$\|f - g\|^2 = \int_0^N \big|f(t) - g(t)\big|^2\,dt \leq N\epsilon^2,$$

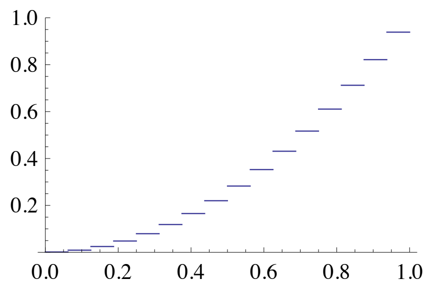
so $\|f - g\| \leq \epsilon\sqrt{N}$. We therefore have the following corollary.



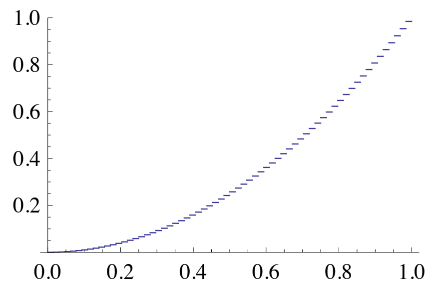
(a) The function to be approximated.



(b) The projection onto V_2.



(c) The projection onto V_4.



(d) The projection onto V_6.

Figure 5.6: Comparison of the function defined by $f(t) = t^2$ on $[0, 1]$ with the projection onto different spaces V_m.

Corollary 5.7. Let f be a given continuous function on the interval $[0, N]$ and let $\text{proj}_{V_m}(f)$ denote the best approximation to f from V_m. Then

$$\lim_{m\to\infty} \|f - \text{proj}_{V_m}(f)\| = 0.$$

Figure 5.6 illustrates how some of the approximations of the function $f(x) = x^2$ from the resolution spaces for the interval $[0, 1]$ improve with increasing m.

5.2.3 Detail spaces and wavelets

So far we have described a family of function spaces that allow us to determine arbitrarily good approximations to a continuous function. The next step is to introduce the so-called detail spaces and the wavelet functions. For this we focus on the two spaces V_0 and V_1.

We start by observing that since

$$[n, n+1) = [2n/2, (2n+1)/2) \cup [(2n+1)/2, 2n/2)$$

we have

$$\phi_{0,n} = \frac{1}{\sqrt{2}}\phi_{1,2n} + \frac{1}{\sqrt{2}}\phi_{1,2n+1}. \tag{5.5}$$

This provides a formal proof of the intuitive observation that $V_0 \subset V_1$. For if $g \in V_0$, then we can write

$$g(t) = \sum_{n=0}^{N-1} c_n \phi_{0,n}(t) = \sum_{n=0}^{N-1} c_n \big(\phi_{1,2n} + \phi_{1,2n+1}\big)/\sqrt{2}.$$

The right-hand side clearly lies in V_1. A similar argument shows that $V_k \subset V_{k+1}$ for any integer $k \geq 0$.

Lemma 5.8. The spaces V_0, V_1, ..., V_m, ... are nested,

$$V_0 \subset V_1 \subset V_2 \subset \cdots \subset V_m \cdots .$$

The next step is to investigate what happens if we start with a function g_1 in V_1 and project this to an approximation g_0 in V_0.

Lemma 5.9. Let proj_{V_0} denote the orthogonal projection onto the subspace V_0. Then the projection of a basis function $\phi_{1,n}$ is given by

$$\mathrm{proj}_{V_0}(\phi_{1,n}) = \begin{cases} \phi_{0,n/2}/\sqrt{2}, & \text{if } n \text{ is even;} \\ \phi_{0,(n-1)/2}/\sqrt{2}, & \text{if } n \text{ is odd.} \end{cases} \tag{5.6}$$

If $g_1 \in V_1$ is given by

$$g_1 = \sum_{n=0}^{2N-1} c_{1,n}\phi_{1,n}, \tag{5.7}$$

then

$$\mathrm{proj}_{V_0}(g_1) = g_0 = \sum_{n=0}^{N-1} c_{0,n}\phi_{0,n}$$

where $c_{0,n}$ is given by

$$c_{0,n} = \frac{c_{1,2n} + c_{1,2n+1}}{\sqrt{2}}. \tag{5.8}$$

Proof. We first observe that $\phi_{1,n}(t) \neq 0$ if and only if $n/2 \leq t < (n+1)/2$. Suppose that n is even. Then the intersection

$$\left[\frac{n}{2}, \frac{n+1}{2}\right) \cap [n_1, n_1 + 1) \tag{5.9}$$

is nonempty only if $n_1 = \frac{n}{2}$. Using the orthogonal decomposition formula we get

$$\begin{aligned}
\text{proj}_{V_0}(\phi_{1,n}) &= \sum_{k=0}^{N-1} \langle \phi_{1,n}, \phi_{0,k} \rangle \phi_{0,k} = \langle \phi_{1,n}, \phi_{0,n_1} \rangle \phi_{0,n_1} \\
&= \int_{n/2}^{(n+1)/2} \sqrt{2}\, dt\, \phi_{0,n/2} = \frac{1}{\sqrt{2}} \phi_{0,n/2}.
\end{aligned}$$

When n is odd, the intersection (5.9) is nonempty only if $n_1 = (n-1)/2$, which gives the second formula in (5.6) in the same way.

We project the function g_1 in V_1 using the formulas in (5.6). We split the sum in (5.7) into even and odd values of n,

$$g_1 = \sum_{n=0}^{2N-1} c_{1,n}\phi_{1,n} = \sum_{n=0}^{N-1} c_{1,2n}\phi_{1,2n} + \sum_{n=0}^{N-1} c_{1,2n+1}\phi_{1,2n+1}. \tag{5.10}$$

We can now apply the two formulas in (5.6),

$$\begin{aligned}
\text{proj}_{V_0}(g_1) &= \text{proj}_{V_0}\left(\sum_{n=0}^{N-1} c_{1,2n}\phi_{1,2n} + \sum_{n=0}^{N-1} c_{1,2n+1}\phi_{1,2n+1} \right) \\
&= \sum_{n=0}^{N-1} c_{1,2n}\, \text{proj}_{V_0}(\phi_{1,2n}) + \sum_{n=0}^{N-1} c_{1,2n+1}\, \text{proj}_{V_0}(\phi_{1,2n+1}) \\
&= \sum_{n=0}^{N-1} c_{1,2n}\phi_{0,n}/\sqrt{2} + \sum_{n=0}^{N-1} c_{1,2n+1}\phi_{0,n}/\sqrt{2} \\
&= \sum_{n=0}^{N-1} \frac{c_{1,2n} + c_{1,2n+1}}{\sqrt{2}} \phi_{0,n}
\end{aligned}$$

which proves (5.8) □

When $g_1 \in V_1$ is projected onto V_0, the result $g_0 = \text{proj}_{V_0} g_1$ is in general different from g_0. We can write $g_1 = g_0 + e_0$, where $e_0 = g_1 - g_0$ represents the error we have commited in making this projection. e_0 lies in the ortogonal complement of V_0 in V_1 (in particular, $e_0 \in V_1$).

Definition 5.10. We will denote by W_0 the orthogonal complement of V_0 in V_1. We also call W_0 a *detail space*

The name detail space is used since $e_0 \in W_0$ can be considered as the detail which is left out when considering g_0 instead of g_1 (due to the expression $g_1 = g_0 + e_0$). We will write $V_1 = V_0 \oplus W_0$ to say that any element in V_1 can be written uniquely as a sum of an element in V_0, and an element in the orthogonal complement W_0. $\oplus$ here denotes what is called a direct sum, which can be more generally defined as follows for any vector spaces which are linearly independent:

Definition 5.11 (Direct sum of vector spaces)**.** Assume that $U, V \subset W$ are vector spaces, and that U and V are mutually linearly independent. By $U \oplus V$ we mean the vector space consisting of all vectors of the form $\boldsymbol{u} + \boldsymbol{v}$, where $\boldsymbol{u} \in U$, $\boldsymbol{v} \in V$. We will also call $U \oplus V$ the *direct sum* of U and V.

This definition also makes sense if we have several vector spaces, since the direct sum clearly obeys the associate law $U \oplus (V \oplus W) = (U \oplus V) \oplus W$, i.e. we can define $U \oplus V \oplus W = U \oplus (V \oplus W)$. We will have use for this use of direct sum of several vector space in the next section.

In other words, the resolution space V_1 is the direct sum of the lower order resolution space V_0, and the detail space W_0. The expression $g_1 = g_0 + e_0$ is thus a decomposition into a low-resolution approximation, and the details which are left out in this approximation. In the context of our Google Earth example, in Figure 5.1 you should interpret g_0 as the image in (a), g_1 as the image in (b), and e_0 as the additional details which are needed to reproduce (b) from (a). While Lemma 5.12 explained how we can compute the low level approximation g_0 from g_1, the next result states how we can compute the detail/error e_0 from g_1.

Lemma 5.12. With W_0 the orthogonal complement of V_0 in V_1, set

$$\hat{\psi}_{0,n} = \frac{\phi_{1,2n} - \phi_{1,2n+1}}{2}$$

for $n = 0, 1, \ldots, N-1$. Then $\hat{\psi}_{0,n} \in W_0$ and

$$\operatorname{proj}_{W_0}(\phi_{1,n}) = \begin{cases} \hat{\psi}_{0,n/2}, & \text{if } n \text{ is even;} \\ -\hat{\psi}_{0,(n-1)/2}, & \text{if } n \text{ is odd.} \end{cases} \tag{5.11}$$

If $g_1 \in V_1$ is given by $g_1 = \sum_{n=0}^{2N-1} c_{1,n}\phi_{1,n}$, then

$$\operatorname{proj}_{W_0}(g_1) = e_0 = \sum_{n=0}^{N-1} \hat{w}_{0,n}\hat{\psi}_{0,n}$$

where $\hat{w}_{0,n}$ is given by

$$\hat{w}_{0,n} = c_{1,2n} - c_{1,2n+1}. \tag{5.12}$$

Proof. We start by determining the error when $\phi_{1,n}$, for n even, is projected onto V_0. The error is then

$$\begin{aligned}
\text{proj}_{W_0}(\phi_{1,n}) &= \phi_{1,n} - \frac{\phi_{0,n/2}}{\sqrt{2}} \\
&= \phi_{1,n} - \frac{1}{\sqrt{2}}\left(\frac{1}{\sqrt{2}}\phi_{1,n} + \frac{1}{\sqrt{2}}\phi_{1,n+1}\right) \\
&= \frac{1}{2}\phi_{1,n} - \frac{1}{2}\phi_{1,n+1} \\
&= \hat{\psi}_{0,n/2}.
\end{aligned}$$

Here we used the relation (5.6) in the second equation. When n is odd we have

$$\begin{aligned}
\text{proj}_{W_0}(\phi_{1,n}) &= \phi_{1,n} - \frac{\phi_{0,(n-1)/2}}{\sqrt{2}} \\
&= \phi_{1,n} - \frac{1}{\sqrt{2}}\left(\frac{1}{\sqrt{2}}\phi_{1,n-1} + \frac{1}{\sqrt{2}}\phi_{1,n}\right) \\
&= \frac{1}{2}\phi_{1,n} - \frac{1}{2}\phi_{1,n-1} \\
&= -\hat{\psi}_{0,(n-1)/2}.
\end{aligned}$$

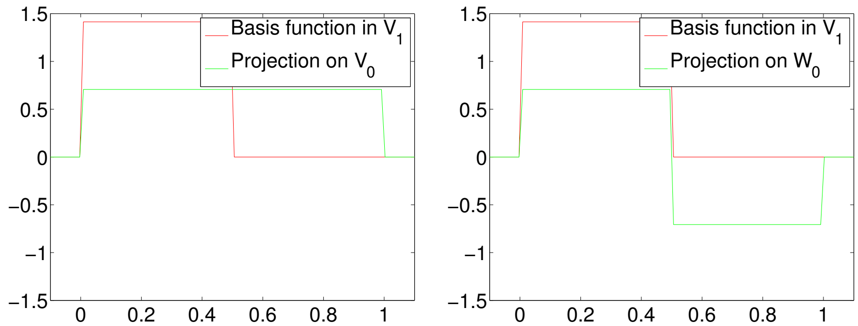
For a general function g_1 we first split the sum into even and odd terms as in (5.10) and then project each part onto W_0,

$$\begin{aligned}
\text{proj}_{W_0}(g_1) &= \text{proj}_{W_0}\left(\sum_{n=0}^{N-1} c_{1,2n}\phi_{1,2n} + \sum_{n=0}^{N-1} c_{1,2n+1}\phi_{1,2n+1}\right) \\
&= \sum_{n=0}^{N-1} c_{1,2n}\,\text{proj}_{W_0}(\phi_{1,2n}) + \sum_{n=0}^{N-1} c_{1,2n+1}\,\text{proj}_{W_0}(\phi_{1,2n+1}) \\
&= \sum_{n=0}^{N-1} c_{1,2n}\hat{\psi}_{0,n} - \sum_{n=0}^{N-1} c_{1,2n+1}\hat{\psi}_{0,n} \\
&= \sum_{n=0}^{N-1} (c_{1,2n} - c_{1,2n+1})\hat{\psi}_{0,n}
\end{aligned}$$

which is (5.12) □

In Figure 5.7 we have useed lemmas 5.9 and 5.12 to plot the projections of $\phi_{1,0} \in V_1$ onto V_0 and W_0. It is an interesting exercise to see from the plots why exactly these functions should be least-squares approximations of $\phi_{1,n}$. It is also an interesting exercise to prove the following from lemmas 5.9 and 5.12:

Proposition 5.13. Let $f(t) \in V_1$, and let $f_{n,1}$ be the value f attains on $[n, n+1/2)$, and $f_{n,2}$ the value f attains on $[n+1/2, n+1)$. Then $\text{proj}_{V_0}(f)$ is the function in V_0 which equals $(f_{n,1} + f_{n,2})/2$ on the interval $[n, n+1)$.



(a) $\phi_{1,0} \in V_1$ together with its projection onto V_0

(b) $\phi_{1,0} \in V_1$ together with its projection onto W_0

Figure 5.7: The projection of a basis function in V_1 onto V_0 and W_0.

Moreover, $\text{proj}_{W_0}(f)$ is the function in V_0 which equals $(f_{n,1} - f_{n,2})/2$ on the interval $[n, n+1)$.

In other words, the projection on V_0 is constructed by averaging on two subintervals, while the projection on W_0 is constructed by taking the difference from the mean. This sounds like a reasonable candidate for the least-squares approximations. In the exercise we generalize these observations.

Consider the functions $\hat{\psi}_{0,n} = (\phi_{1,2n} - \phi_{1,2n+1})/2$ from Lemma 5.12. They are clearly orthogonal since their nonzero parts do not overlap. We also note that $\|\hat{\psi}_{0,n}\| = \sqrt{2}/2$, since it has absolute value $\sqrt{2}/2$ on two intervals of length $1/2$. The functions defined by $\psi_{0,n}(t) = \sqrt{2}\,\hat{\psi}_{0,n}(t)$ will therefore form an orthonormal set.

Lemma 5.14. Define the function ψ by

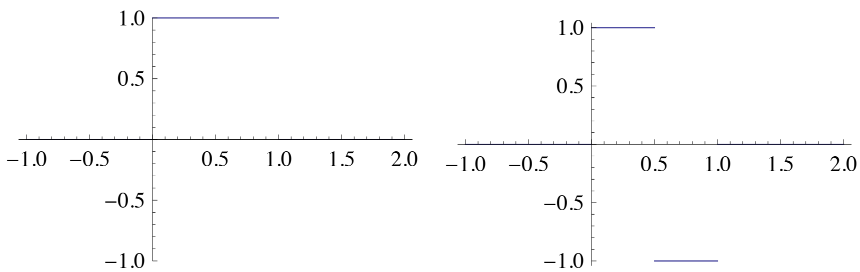
$$\psi(t) = \big(\phi_{1,0}(t) - \phi_{1,1}(t)\big)/\sqrt{2} = \phi(2t) - \phi(2t-1) \tag{5.13}$$

and set

$$\psi_{0,n}(t) = \psi(t-n) = \big(\phi_{1,2n}(t) - \phi_{1,2n+1}(t)\big)/\sqrt{2} \quad \text{for } n = 0,\, 1,\, \ldots,\, N-1. \tag{5.14}$$

Then the set $\{\psi_{0,n}\}_{n=0}^{N-1}$ is an orthonormal basis for W_0, the orthogonal complement of V_0 in V_1.

Later we will encounter other functions, which also will be denoted by ψ, and have similar properties as stated in Lemma 5.14. In the theory of wavelets, such ψ are called *mother wavelets*. In Figure 5.8 we have plotted the functions ϕ and ψ. There is one important property of ψ, which we will return to:



(a) The function ϕ (a)
(b) The function ψ

Figure 5.8: The functions we used to analyse the space of piecewise constant functions

Observation 5.15. We have that $\int_0^N \psi(t)dt = 0$.

This can be seen directly from the plot in Figure 5.8, since the parts of the graph above and below the x-axis cancel.

We now have all the tools needed to define the Discrete Wavelet Transform.

Theorem 5.16 (Discrete Wavelet Transform)**.** The space V_1 can be decomposed as the orthogonal sum $V_1 = V_0 \oplus W_0$ where W_0 is the orthogonal complement of V_0 in V_1, and V_1 therefore has the two bases

$$\boldsymbol{\phi}_1 = (\phi_{1,n})_{n=0}^{2N-1} \quad \text{and} \quad (\boldsymbol{\phi}_0, \boldsymbol{\psi}_0) = \big((\phi_{0,n})_{n=0}^{N-1}, (\psi_{0,n})_{n=0}^{N-1}\big).$$

The Discrete Wavelet Transform (DWT) is the change of coordinates from the basis $\boldsymbol{\phi}_1$ to the basis $(\boldsymbol{\phi}_0, \boldsymbol{\psi}_0)$. If

$$g_1 = \sum_{n=0}^{2N-1} c_{1,n}\phi_{1,n} \in V_1$$
$$g_0 = \sum_{n=0}^{N-1} c_{0,n}\phi_{0,n} \in V_0$$
$$e_0 = \sum_{n=0}^{N-1} w_{0,n}\psi_{0,n} \in W_0$$

and $g_1 = g_0 + e_0$, then the DWT is given by

$$c_{0,n} = (c_{1,2n} + c_{1,2n+1})/\sqrt{2} \tag{5.15}$$
$$w_{0,n} = (c_{1,2n} - c_{1,2n+1})/\sqrt{2}. \tag{5.16}$$

Conversely, the Inverse Discrete Wavelet Transform (IDWT) is the change of coordinates from the basis $(\boldsymbol{\phi}_0, \boldsymbol{\psi}_0)$ to the basis $\boldsymbol{\phi}_1$, and is given by

$$c_{1,2n} = (c_{0,n} + w_{0,n})/\sqrt{2} \tag{5.17}$$
$$c_{1,2n+1} = (c_{0,n} - w_{0,n})/\sqrt{2}. \tag{5.18}$$

Proof. Most of this theorem has already been established. In particular, the formulas (5.15)–(5.16) are just (5.8) and (5.12). What remains is to prove the formulas (5.17)–(5.18). For this we note from (5.5) and (5.14) that

$$g_0 + e_0 = \sum_{n=0}^{N-1} c_{0,n}\phi_{0,n} + \sum_{n=0}^{N-1} w_{0,n}\psi_{0,n} \tag{5.19}$$
$$= \sum_{n=0}^{N-1} c_{0,n}(\phi_{1,2n} + \phi_{1,2n+1})/\sqrt{2} + \sum_{n=0}^{N-1} w_{0,n}(\phi_{1,2n} - \phi_{1,2n+1})/\sqrt{2} \tag{5.20}$$
$$= \sum_{n=0}^{N-1} (c_{0,n} + w_{0,n})\phi_{1,2n}/\sqrt{2} + (c_{0,n} - w_{0,n})\phi_{1,2n+1}/\sqrt{2}. \tag{5.21}$$

□

It is common to reorder the basis vectors in $(\boldsymbol{\phi}_0, \boldsymbol{\psi}_0)$ to

$$\mathcal{C}_1 = \{\phi_{0,0}, \psi_{0,0}, \phi_{0,1}, \psi_{0,1}, \cdots, \phi_{0,N-1}, \psi_{0,N-1}\}. \tag{5.22}$$

The subscript 1 is used since $\mathcal{C}_1$ is a basis for V_1. This reordering of the basis functions is useful since it makes it easier to write down the change of coordinates matrices. To be more precise, from formulas (5.17)–(5.18) it is apparent that $P_{\boldsymbol{\phi}_1 \leftarrow \mathcal{C}_1}$ is the matrix where

$$\begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix}$$

is repeated along the main diagonal N times. Also, from formulas (5.15)–(5.16) it is apparent that $P_{\mathcal{C}_1 \leftarrow \boldsymbol{\phi}_1}$ is the same matrix. Such matrices are called *block diagonal matrices*. This particular block diagonal matrix is clearly orthogonal, since it transforms one orthonormal base to another.

Exercises for Section 5.2

Ex. 1 — Show that the coordinate vector for $f \in V_0$ in the basis $\{\phi_{0,0}, \phi_{0,1}, \ldots, \phi_{0,N-1}\}$ is $(f(0), f(1), \ldots .f(N-1))$.

Ex. 2 — Show that

$$\mathrm{proj}_{V_0}(f) = \sum_{n=0}^{N-1} \left(\int_n^{n+1} f(t)dt \right) \phi_{0,n}(t) \tag{5.23}$$

for any f. Show also that the first part of Proposition 5.13 follows from this.

Ex. 3 — Show that

$$\begin{aligned} &\| \sum_n \left(\int_n^{n+1} f(t)dt \right) \phi_{0,n}(t) - f \|^2 \\ &= \langle f, f \rangle - \sum_n \left(\int_n^{n+1} f(t)dt \right)^2 . \end{aligned}$$

This, together with the previous exercise, gives us an expression for the least-squares error for f from V_0 (at least after taking square roots).

Ex. 4 — Consider the projection T of V_1 onto V_0.

a. Show that $T(\phi) = \phi$ and $T(\psi) = 0$.
b. Show that the matrix of T relative to $\mathcal{C}_1$ is given by the diagonal matrix where 1 and 0 are repeated alternatingly on the diagonal, N times (i.e. 1 at the even indices, 0 at the odd indices).
c. Show in a similar way that the projection of V_1 onto W_0 has a matrix relative to $\mathcal{C}_1$ given by the diagonal matrix where 1 and 0 also are repeated alternatingly on the diagonal, but with the opposite order.

Ex. 5 — Use lemma 5.9 to write down the matrix for the linear transformation $\mathrm{proj}_{V_0} : V_1 \to V_0$ relative to the basis $\boldsymbol{\phi}_1$ and $\boldsymbol{\phi}_0$. Also, use lemma 5.12 to write down the matrix for the linear transformation $\mathrm{proj}_{W_0} : V_1 \to W_0$ relative to the basis $\boldsymbol{\phi}_1$ and $\boldsymbol{\psi}_0$.

Ex. 6 — Show that

$$\mathrm{proj}_{W_0}(f) = \sum_{n=0}^{N-1} \left(\int_n^{n+1/2} f(t)dt - \int_{n+1/2}^{n+1} f(t)dt \right) \psi_{0,n}(t) \tag{5.24}$$

for any f. Show also that the second part of Proposition 5.13 follows from this.

5.3 Multiresolution analysis for piecewise constant functions

In the Section 5.2 we introduced the important decomposition $V_1 = V_0 \oplus W_0$ which lets us rewrite a function in V_1 as an approximation in V_0 and the corresponding error in W_0 which is orthogonal to the approximation. The resolution spaces V_m were in fact defined for all integers $m \geq 0$. It turns out that all these resolution spaces can be decomposed in the same way as V_1.

Definition 5.17. The orthogonal complement of V_{m-1} in V_m is denoted W_{m-1}. All the spaces $\{W_k\}_k$ are also called detail spaces.

The first question we will try to answer is how we can, for $f \in V_m$, extract the corresponding detail in W_{m-1}.

5.3.1 Extraction of details at higher resolutions

We first need to define $\psi_{m,n}$ in terms of ψ, similarly to how we defined $\phi_{m,n}$ in terms of ϕ,

$$\psi_{m,n}(t) = 2^{m/2}\psi(2^m t - n), \quad \text{for } n = 0, 1, \ldots, 2^m N - 1. \tag{5.25}$$

As in Lemma 5.14, it is straightforward to prove that $\boldsymbol{\psi}_m = \{\psi_{m,n}\}_{n=0}^{2^m N-1}$ is an orthonormal basis for W_m. Moreover, we have the following result, which is completey analogous to Theorem 5.16.

Theorem 5.18. The space V_m can be decomposed as the orthogonal sum $V_m = V_{m-1} \oplus W_{m-1}$ where W_{m-1} is the orthogonal complement of V_{m-1} in V_m, and V_m has the two bases

$$\boldsymbol{\phi}_m = (\phi_{m,n})_{n=0}^{2^m N-1}$$

and

$$(\boldsymbol{\phi}_{m-1}, \boldsymbol{\psi}_{m-1}) = \big((\phi_{m-1,n})_{n=0}^{2^{m-1}N-1}, (\psi_{m-1,n})_{n=0}^{2^{m-1}N-1}\big).$$

If

$$g_m = \sum_{n=0}^{2^m N-1} c_{m,n}\phi_{m,n} \in V_m,$$

$$g_{m-1} = \sum_{n=0}^{2^{m-1}N-1} c_{m-1,n}\phi_{m-1,n} \in V_{m-1},$$

$$e_{m-1} = \sum_{n=0}^{2^{m-1}N-1} w_{m-1,n}\psi_{m-1,n} \in W_{m-1},$$

and $g_m = g_{m-1} + e_{m-1}$, then the change of coordinates from the basis $\boldsymbol{\phi}_m$ to the basis $(\boldsymbol{\phi}_{m-1}, \boldsymbol{\psi}_{m-1})$ is given by

$$c_{m-1,n} = (c_{m,2n} + c_{m,2n+1})/\sqrt{2}, \tag{5.26}$$

$$w_{m-1,n} = (c_{m,2n} - c_{m,2n+1})/\sqrt{2}. \tag{5.27}$$

Conversely, the change of coordinates from the basis $(\boldsymbol{\phi}_{m-1}, \boldsymbol{\psi}_{m-1})$ to the basis $\boldsymbol{\phi}_m$ is given by

$$c_{m,2n} = (c_{m-1,n} + w_{m-1,n})/\sqrt{2}, \tag{5.28}$$

$$c_{m,2n+1} = (c_{m-1,n} - w_{m-1,n})/\sqrt{2}. \tag{5.29}$$

We will omit the proof of Theorem 5.18, and only remark that it can be proved by making the substitution $t \to 2^m u$ in Lemma 5.9 and Lemma 5.12, and then following the proof of Theorem 5.16. Clearly, we can now find the change of coordinate matrices as before, and as before this is most easily expressed if we reorder the basis vectors for $(\boldsymbol{\phi}_m, \boldsymbol{\psi}_m)$ again as in Equation (5.22), i.e. we define

$$\mathcal{C}_m = \{\phi_{m,0}, \psi_{m,0}, \phi_{m,1}, \psi_{m,1}, \cdots, \phi_{m,2^{m-1}N-1}, \psi_{m,2^{m-1}N-1}\}. \tag{5.30}$$

The bases $\boldsymbol{\phi}_m$ and $\mathcal{C}_m$ are both referred to as *wavelet bases*. It is now apparent that both change of coordinates matrices $P_{\boldsymbol{\phi}_m \leftarrow \mathcal{C}_m}$, $P_{\mathcal{C}_m \leftarrow \boldsymbol{\phi}_m}$ can be obtained by repeating the matrix

$$\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

along the diagonal, but this time it is repeated $2^{m-1}N$ times. In mathematical statements in the following, we will always express a change of coordinates in terms of the wavelet bases $\boldsymbol{\phi}_m$ and $\mathcal{C}_m$, due to the nice expression this matrix then has. In implementations, however, we also need to reorder $\mathcal{C}_m$ to $(\boldsymbol{\phi}_{m-1}, \boldsymbol{\psi}_{m-1})$, in order to prepare for successive changes of coordinates, as we will now describe.

Let us return to our interpretation of the Discrete Wavelet Transform as writing a function $g_1 \in V_1$ as a sum of a function $g_0 \in V_0$ at low resolution, and a detail function $e_0 \in W_0$. Theorem 5.18 states similarly how we can write $g_m \in V_m$ as a sum of a function $g_{m-1} \in V_{m-1}$ at lower resolution, and a detail function $e_{m-1} \in W_{m-1}$. The same decomposition can of course be applied to g_{m-1} in V_{m-1}, then to the resulting approximation g_{m-2} in V_{m-2}, and so on,

$$\begin{aligned} V_m &= V_{m-1} \oplus W_{m-1} \\ &= V_{m-2} \oplus W_{m-2} \oplus W_{m-1} \\ &\vdots \\ &= V_0 \oplus W_0 \oplus W_1 \oplus \cdots \oplus W_{m-2} \oplus W_{m-1}. \end{aligned} \tag{5.31}$$

This change of coordinates corresponds to replacing as many ϕ-functions as we can with ψ-functions, i.e. replacing the original function with a sum of as much detail at different resolutions as possible. Let us give a name to the bases we will use for these direct sums.

Definition 5.19 (Canonical basis for direct sum)**.** Let $C_1, C_2 \ldots, C_n$ be independent vector spaces, and let $\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_n$ be corresponding bases. The basis $\{\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_n\}$, i.e., the basis where the basis vectors from $\mathcal{B}_i$ are included before $\mathcal{B}_j$ when $i < j$, is referred to as the canonical basis for $C_1 \oplus C_2 \oplus \cdots \oplus C_n$ and is dentoed $\mathcal{B}_1 \oplus \mathcal{B}_2 \oplus \ldots \oplus \mathcal{B}_n$.

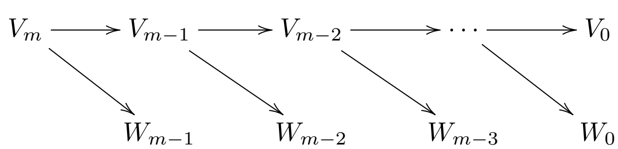
When we above say "basis for $V_0 \oplus W_0 \oplus W_1 \oplus \cdots \oplus W_{m-2} \oplus W_{m-1}$", we really mean the canonical basis for this space. In general, the Discrete Wavelet Transform is used to denote a change of coordinates from ϕ_m to the canonical basis, for any m.

Definition 5.20 (m-level Discrete Wavelet Transform)**.** Let F_m denote the change of coordinates matrix from $\boldsymbol{\phi}_m$ to the canonical basis

$$\boldsymbol{\phi}_0 \oplus \boldsymbol{\psi}_0 \oplus \boldsymbol{\psi}_1 \oplus \cdots \oplus \oplus \boldsymbol{\psi}_{m-2} \oplus \boldsymbol{\psi}_{m-1}$$

for $V_0 \oplus W_0 \oplus W_1 \oplus \cdots \oplus W_{m-2} \oplus W_{m-1}$. The matrix F_m is called a (m-level) Discrete Wavelet Transform, or a DWT. After this change of coordinates, the resulting coordinates are called wavelet coefficients. The change of coordinates the opposite way is called an (m-level) Inverse Discrete Wavelet Transform, or IDWT.

Clearly, this generalizes the Discrete Wavelet Transform defined in Section 5.2. At each level in a DWT, V_k is split into one part from V_{k-1}, and one part from W_{k-1}. We can visualize this with the following figure, where the arrows represent changes of coordinates:



The part from W_{k-1} is not subject to further transformation. This is seen in the figure since W_{m-1} is a leaf node, i.e. there are no arrows going out from W_{m-1}. In a similar illustration for the IDWT, the arrows would go the opposite way. The Discrete Wavelet Transform is the analogue in a wavelet setting to the Discrete Fourier transform. When applying the DFT to a vector of length N, one starts by viewing this vector as coordinates relative to the standard basis. When applying the DWT to a vector of length N, one instead views the vector as coordinates relative to the basis $\boldsymbol{\phi}_m$. This makes sense in light of Exercise 5.2.1.

The DWT is what is used in practice when transforming a signal using wavelets, and it is straightforward to implement: One simply needs to iterate(5.26)-(5.27) for $m, m-1, \ldots, 1$, also at each step, the coordinates in $\boldsymbol{\phi}_{m-1}$ should be placed before the ones in $\boldsymbol{\psi}_{m-1}$, due to the order of the basis vectors in the canonical basis of the direct sum. At each step, only the first coordinates are further transformed. The following function, called `DWTHaarImpl`, follows this procedure. It takes as input the number of levels m, as well as the input vector $\boldsymbol{x}$, runs the m-level DWT on $\boldsymbol{x}$, and returns the result:

```
function xnew=DWTHaarImpl(x,m)
  xnew=x;
  for mres=m:(-1):1
    len=length(xnew)/2^(m-mres);
    c=(xnew(1:2:(len-1))+xnew(2:2:len))/sqrt(2);
    w=(xnew(1:2:(len-1))-xnew(2:2:len))/sqrt(2);
    xnew(1:len)=[c w];
  end
```

Note that this implementation is not recursive, contrary to the FFT. The for-loop here runs through the different resolutions. Inside the loop we perform the change of coordinates from $\boldsymbol{\phi}_k$ to $(\boldsymbol{\phi}_{k-1}, \boldsymbol{\psi}_{k-1})$ by applying equations (5.26)-(5.27). This works on the first coordinates, since the coordinates from $\boldsymbol{\phi}_k$ are stored first in

$$V_k \oplus W_k \oplus W_{k+1} \oplus \cdots \oplus W_{m-2} \oplus W_{m-1}.$$

Finally, the $\boldsymbol{c}$-coordinates are stored before the $\boldsymbol{w}$-coordinates, again as required by the order in the canonical basis. In this implementation, note that the first levels require the most multiplications, since the latter levels leave an increasing part of the coordinates unchanged. Note also that the change of coordinates matrix is a very sparse matrix: At each level a coordinate can be computed from only two of the other coordinates, so that this matrix has only two nonzero elements in each row/column. The algorithm clearly shows that there is no need to perform a full matrix multiplication to perform the change of coordinates.

The corresponding function for the IDWT, called `IDWTHaarImpl`, goes as follows:

```
function x=IDWTHaarImpl(xnew,m)
  x=xnew;
  for mres=1:m
    len=length(x)/2^(m-mres);
    ev=(x(1:(len/2))+x((len/2+1):len))/sqrt(2);
    od=(x(1:(len/2))-x((len/2+1):len))/sqrt(2);
    x(1:2:(len-1))=ev;
    x(2:2:len)=od;
  end
```

Here the steps are simply performed in the reverse order, and by iterating equations (5.28)-(5.29).

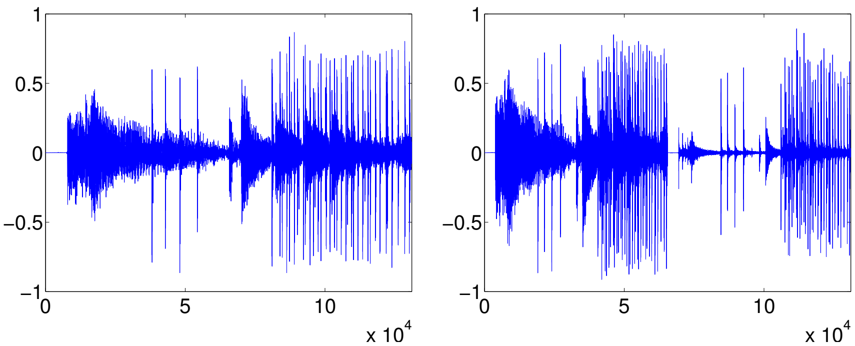
You may be puzzled by the names `DWTHaarImpl` and `IDWTHaarImpl`. In the next sections we will consider other cases, where the underlying function ϕ may be a different function, not necessarily piecewise constant. It will turn out that much of the analysis we have done makes sense for other functions ϕ as well, giving rise to other structures which we also will refer to as wavelets. The wavelet resulting from piecewise constant functions is thus simply one example out of many, and it is commonly referred to as the *Haar wavelet*.

Example 5.21. When you run a DWT you may be led to believe that coefficients from the lower order resolution spaces may correspond to lower frequencies. This sounds reasonable, since the functions $\phi(2^m t - n) \in V_m$ change more quickly than $\phi(t-n) \in V_0$. However, the functions $\phi_{m,n}$ do not correspond to pure tones in the setting of wavelets. But we can still listen to sound from the different resolution spaces. In Exercise 9 you will be asked to implement a function which runs an m-level DWT on the first samples of the sound file `castanets.wav`, extracts the coefficients from the lower order resolution spaces, transforms the values back to sound samples with the IDWT, and plays the result. When you listen to the result the sound is clearly recognizable for lower values of m, but is degraded for higher values of m. The explanation is that too much of the detail is omitted when you use a higher m. To be more precise, when listening to the sound by throwing away wvereything from the detail spaces $W_0, W_1, \ldots, W_{m-1}$, we are left with a 2^{-m} share of the data. Note that this procedure is mathematically not the same as setting some DFT coefficients to zero, since the DWT does not operate on pure tones.

It is of interest to plot the samples of our test audio file `castanets.wav`, and compare it with the first order DWT coefficients of the same samples. This is shown in Figure 5.9. The first part half of the plot represents the low-resolution approximation of the sound, the second part represents the detail/error. We see that the detail is quite significant in this case. This means that the first order wavelet approximation does not give a very good approximation to the sound. In the exercises we will experiment more on this.

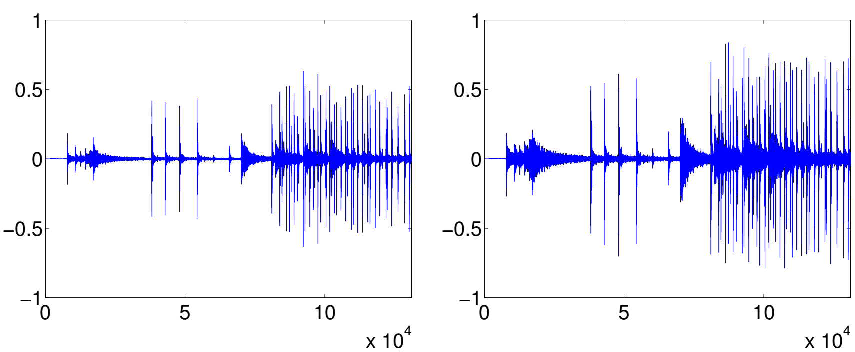
It is also interesting to plot only the detail/error in the sound, for different resolutions. For this, we must perform a DWT so that we get a representation in the basis $V_0 \oplus W_0 \oplus W_1 \oplus \cdots \oplus W_{m-1}$, set the coefficicents from V_0 to sero, and transform back with the IDWT. In figure 5.10 the error is shown for the test audio file `castanets.wav` for $m = 1$, $m = 2$. This clearly shows that the error is larger when two levels of the DWT are performed, as one would suspect. It is also seen that the error is larger in the part of the file where there are bigger variations. This also sounds reasonable.

The previous example illustrates that wavelets as well may be used to perform operations on sound. As we will see later, however, our main application for wavelets will be images, where they have found a more important role than for sound. Images typically display variations which are less abrupt than the ones found in sound. Just as the functions above had smaller errors in the corresponding resolution spaces than the sound had, images are thus more suited for for use with wavelets. The main idea behind why wavelets are so useful comes



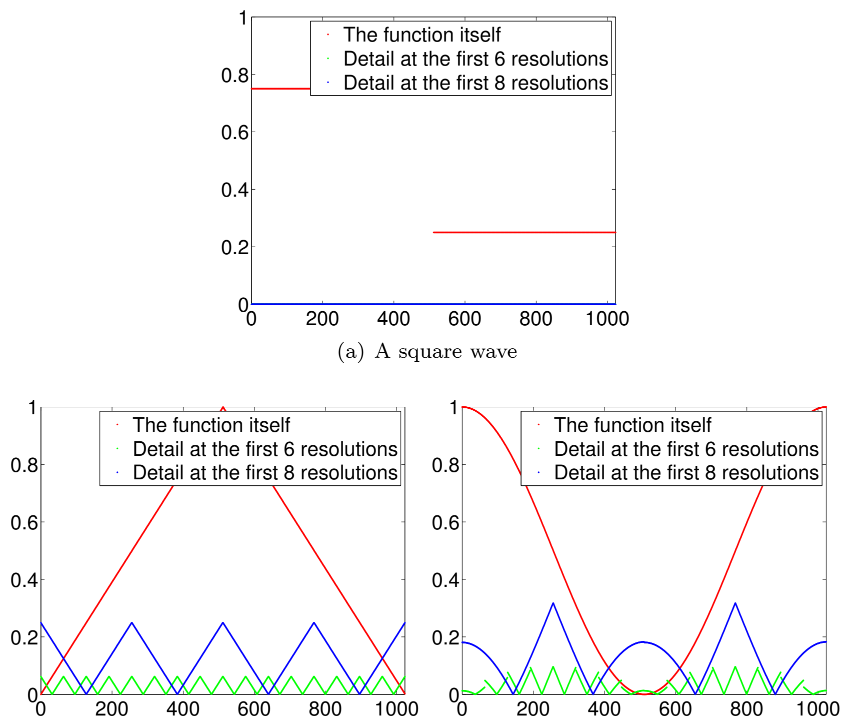
(a) The 2^{17} first sound samples (b) The first order DWT coefficients

Figure 5.9: The sound samples and the DWT coefficients of the sound `castanets.wav`.



(a) $m = 1$ (b) $m = 2$

Figure 5.10: The error (i.e. the contribution from $W_0 \oplus W_1 \oplus \cdots \oplus W_{m-1}$) in the sound file `castanets.wav`, for different values of m.



(a) A square wave

(b) $f(t) = 1 - 2|1/2 - t/N|$

(c) $f(t) = 1/2 + \cos(2\pi t/N)/2$

Figure 5.11: The error (i.e. the contribution from $W_0 \oplus W_1 \oplus \cdots \oplus W_{m-1}$) for $N = 1024$ for different functions $f(t)$, for different values of m.

from the fact that the detail, i.e., wavelet coefficients corresponding to the spaces W_k, are often very small. After a DWT one is therefore often left with a couple of significant coefficients, while most of the coefficients are small. The approximation from V_0 can be viewed as a good approximation, even though it contains much less information. This gives another reason why wavelets are popular for images: Detailed images can be very large, but when they are downloaded to a web browser, the browser can very early show a low-resolution of the image, while waiting for the rest of the details in the image to be downloaded. When we later look at how wavelets are applied to images, we will need to handle one final hurdle, namely that images are two-dimensional.

Example 5.22. Above we plotted the DWT coefficients of a sound, as well as the detail/error. We can also experiment with samples generated from a mathematical function. Figure 5.11 plots the error for different functions, with $N = 1024$. In these cases, we see that we require large m before the detail/error becomes significant. We see also that there is no error for the square wave. The reason is that the square wave is a piecewise constant function, so that it can be represented exactly by the ϕ-functions. For the other functions, however, this is not the case, so we here het an error.

Above we used the functions DWTHaarImpl, IDWTHaarImpl to plot the error. For the functions we plotted in the previous example it is also possible to compute the wavelet coefficients, which we previously have denoted by $w_{m,n}$, exactly. You will be asked to do this in exercises 12 and 13. The following example shows the general procedure which can be used for this:

Example 5.23. Let us compute the wavelet coefficients $w_{m,n}$ for the function $f(t) = 1-t/N$. This function decreases linearly from 1 to 0 on $[0, N]$. Since the $w_{m,n}$ are coefficients in the basis $\{\psi_{m,n}\}$, it follows by the orthogonal decomposition formula that $w_{m,n} = \langle f, \psi_{m,n}\rangle = \int_0^N f(t)\psi_{m,n}(t)dt$. Using the definition of $\psi_{m,n}$ we get that

$$w_{m,n} = \int_0^N (1-t/N)\psi_{m,n}(t)dt = 2^{m/2}\int_0^N (1-t/N)\psi(2^m t - n)dt.$$

Moreover $\psi_{m,n}$ is nonzero only on $[2^{-m}n, 2^{-m}(n+1))$, and is 1 on $[2^{-m}n, 2^{-m}(n+1/2))$, and -1 on $[2^{-m}(n+1/2), 2^{-m}(n+1))$. We can therefore write

$$\begin{aligned}
w_{m,n} &= 2^{m/2}\int_{2^{-m}n}^{2^{-m}(n+1/2)}(1-t/N)dt - 2^{m/2}\int_{2^{-m}(n+1/2)}^{2^{-m}(n+1)}(1-t/N)dt \\
&= 2^{m/2}\left[t - \frac{t^2}{2N}\right]_{2^{-m}n}^{2^{-m}(n+1/2)} - 2^{m/2}\left[t - \frac{t^2}{2N}\right]_{2^{-m}(n+1/2)}^{2^{-m}(n+1)} \\
&= 2^{m/2}\left(2^{-m}(n+1/2) - \frac{2^{-2m}(n+1/2)^2}{2N} - 2^{-m}n + +\frac{2^{-2m}n^2}{2N}\right) \\
&- 2^{m/2}\left(2^{-m}(n+1) - \frac{2^{-2m}(n+1)^2}{2N} - 2^{-m}(n+1/2) + \frac{2^{-2m}(n+1/2)^2}{2N}\right) \\
&= 2^{m/2}\left(\frac{2^{-2m}n^2}{2N} - \frac{2^{-2m}(n+1/2)^2}{N} + \frac{2^{-2m}(n+1)^2}{2N}\right) \\
&= \frac{2^{-3m/2}}{2N}\left(n^2 - 2(n+1/2)^2 + (n+1)^2\right) \\
&= \frac{1}{N2^{2+3m/2}}.
\end{aligned}$$

We see in particular that $w_{m,n} \to 0$ when $m \to \infty$. We see also that there were a lot of computations even in this very simple example. For most functions we therefore usually do not compute $w_{m,n}$ exactly. Instead we use implementations like DWTHaarImpl, IDWTHaarImpl, and run them on a computer.

5.3.2 Matrix factorization of the DWT

In this section we will write down a matrix factorization of the DWT. This factorization is not used much in mathematical statements, since one typically hides this in implementations of the DWT. This is very similar to the case for the

FFT, where the matrix factorizations grow increasingly complex when $N = 2^n$ is large, but where the algorithms are still very compact. We need the concept of a direct sum of matrices before we can write down the DWT factorization:

Definition 5.24 (Direct sum of matrices)**.** Let $T_1, T_2, \ldots, T_n$ be square matrices. By the direct sum of $T_1, \ldots, T_n$, denoted $T_1 \oplus T_2 \oplus \cdots \oplus T_n$, we mean the block-diagonal matrix where the matrices T_1, T_2,...,T_n are placed along the diagonal, with zeros everywhere else.

We can now establish the matrix factorization of the DWT and IDWT in terms of the direct sum of matrices:

Theorem 5.25 (Matrix of the m-level DWT)**.** Define the $(2^m N) \times (2^m N)$ change of coordinate matrices

$$\begin{aligned} G_m &= (P_{\boldsymbol{\phi}_m \leftarrow \mathcal{C}_m})^T \\ H_m &= P_{\mathcal{C}_m \leftarrow \boldsymbol{\phi}_m} \end{aligned}$$

The m-level DWT and IDWT can be expressed as

$$\begin{aligned} F_m &= (P_{2^1 N} H_1 \oplus I_{2^m N - 2^1 N})(P_{2^2 N} H_2 \oplus I_{2^m N - 2^2 N}) \\ &\qquad \cdots (P_{2^{m-1} N} H_{m-1} \oplus I_{2^m N - 2^{m-1} N}) P_{2^m N} H_m, \\ (F_m)^{-1} &= (P_{2^m N} G_m)^T ((P_{2^{m-1} N} G_{m-1})^T \oplus I_{2^m N - 2^{m-1} N}) \\ &\qquad \cdots ((P_{2^2 N} G_2)^T \oplus I_{2^m N - 2^2 N})((P_{2^1 N} G_1)^T \oplus I_{2^m N - 2^1 N}). \end{aligned}$$

where P_N is the matrix we used to group a vector into its even- and odd indexed samples in Section 4.1 (i.e. $P_N \boldsymbol{x} = (\boldsymbol{x}^{(e)}, \boldsymbol{x}^{(o)})$).

Proof. The m level DWT performs m changes of coordinates in order. For $k = 0, 1, \ldots, m-1$, these steps are (in this order), the change of coordinates from the canonical basis of $V_{m-k} \oplus_{r=m-k}^{m-1} W_r$ to the canonical basis of $V_{m-k-1} \oplus_{r=m-k-1}^{m-1} W_r$. This change of coordinates only transforms the coordinates from V_{m-k}, and there are $2^{m-k} N$ such coordinates. The remaining $2^m N - 2^{m-k} N$ coordinates are left unchanged, which corresponds to

$$0, 2^m N - 2^{m-1} N, \ldots, 2^m N - 2^{m-(m-2)} N, ..., 2^m N - 2^{m-(m-1)} N$$

coordinates for $k = 0, 1, \ldots, m-1$, which explain the I_0, $I_{2^m N - 2^{m-1} N}$,..., $I_{2^m N - 2^2 N}$, $I_{2^m N - 2^1 N}$ matrices above from right to left. The change of coordinates from V_{m-k} to $V_{m-k-1} \oplus W_{m-k-1}$ is implemented with the change of coordinates matrix H_{m-k}, followed by a reoredering of the coordinates so that the even-indexed ones come first. It is clear that this can be implemented as $P_{2^{m-k} N} H_{m-k}$, where $P_{2^{m-k} N}$ is defined as in Section 4.1. This explains the matrices $P_{2^m N} H_m$, $P_{2^{m-1} N} H_{m-1}$,..., $P_{2^2 N} H_2$, $P_{2^1 N} H_1$ above, from right to left.

The m-level IDWT is the product of the inverse matrices in the opposite order. We have that

$$\begin{aligned}(P_{2^{m-k}N}H_{m-k} \oplus I_{2^mN-2^{m-k}N})^{-1} &= (P_{2^{m-k}N}H_{m-k})^{-1} \oplus I_{2^mN-2^{m-k}N}\\ &= (G_{m-k})^T(P_{2^{m-k}N})^T \oplus I_{2^mN-2^{m-k}N}\\ &= (P_{2^{m-k}N}G_{m-k})^T \oplus I_{2^mN-2^{m-k}N}\end{aligned}$$

where we used Exercise 5. The result now follows. □

A good question is why we use the transpose of G_m in its definition. We will discuss this later.

5.3.3 Summary

Let us finally summarize the properties of the spaces V_m. We showed that they were nested, i.e.

$$V_0 \subset V_1 \subset V_2 \subset \cdots \subset V_m \subset \cdots.$$

We also showed that continuous functions could be approximated arbitrarily well from V_m, as long as m was chosen large enough. Moreover it is clear that the space V_0 is closed under all translates, at least if we view the functions in V_0 as periodic with period N, defined as previously on the period $[0, N)$ (translating with N then means that we get the same function back). In the following we will always identify a function with this periodic extension, just as we did in Fourier analysis. When performing this identification, it is also clear that $f(t) \in V_m$ if and only if $g(t) = f(2t) \in V_{m+1}$. We have therefore shown that the scaling funtion ϕ fits in with the following general framework.

Definition 5.26. A Multiresolution analysis, or MRA, is a nested sequence of function spaces

$$V_0 \subset V_1 \subset V_2 \subset \cdots \subset V_m \subset \cdots \tag{5.32}$$

so that

1. Any continuous function can be approximated arbitrarily well from V_m, as long as m is large enough,
2. $f(t) \in V_0$ if and only if $f(2^m t) \in V_m$,
3. $f(t) \in V_0$ if and only if $f(t-n) \in V_0$ for all n.
4. There is a function ϕ, called a scaling function, so that $\{\phi(t-n)\}_{0\leq n<N}$ is a basis for V_0. .

Note that, while the basis function we have seen up to now have been orthogonal, we state here that we allow them to be simply a basis as well. The reason is that it will turn out that the assumption of orthogonality may be too strict, in that it makes it difficult to construct interesting wavelets. We will return to this. The concept of Multiresolution Analysis is much used, and one can find a wide variety of functions ϕ (not only piecewise constant functions), which gives rise to a Multiresolution Analysis. With a multiresolution analysis there is another important thing we also need: We need to be able to efficiently compute the decomposition of $g_m \in V_m$ into the low resolution approximation $g_{m-1} \in V_{m-1}$ and the detail $e_{m-1} \in W_{m-1}$. This requires that we have a simple expression for the corresponding projections. In particular, we need to find a basis for W_m, which hopefully also is orthonormal. Once we have this, the orthogonal decomposition formula can be used to compute the projections, i.e. we can compute the detail and low resolution approximations. Let us summarize this in the following recipe for constructing wavelets:

Idea 5.27 (Recipe for constructing wavelets)**.** In order to construct wavelets which are useful for practical purposes, we need to do the following:

1. Find a function ϕ which gives rise to a multiresolution analysis, and so that we easily can compute the projection from V_1 onto V_0.
2. Find a function ψ so that $\{\psi(t-n)\}_{0\leq n<N}$ is an orthonormal basis for W_0, and so that we easily can compute the projection from V_1 onto W_0.

If we can achieve this, the m-level Discrete Wavelet Transform can be defined and computed similarly as in the case when ϕ is a piecewise constant function, with the obvious replacements.

In the next sections we will follow this recipe in order to contruct other wavelets. Along the way we will run into other questions which are interesting. One of them is, given the resolution spaces, is there a unique choice of ϕ, ψ? If not, are any choices of ϕ, ψ better than others? How can we quantify how good such a choice is?

Exercises for Section 5.3

Ex. 1 — Generalize exercise 5.2.4 to the projections from V_{m+1} onto V_m amd W_m.

Ex. 2 — Show that $f(t) \in V_m$ if and only if $g(t) = f(2t) \in V_{m+1}$.

Ex. 3 — Let $C_1, C_2 \ldots, C_n$ be independent vector spaces, and let $T_i : C_i \to C_i$ be linear transformations. The direct sum of T_1, T_2,…,T_n which is written $T_1 \oplus T_2 \oplus \ldots \oplus T_n$ denotes the linear transformation from $C_1 \oplus C_2 \oplus \cdots \oplus C_n$ to itself defined by

$$T_1 \oplus T_2 \oplus \ldots \oplus T_n(\boldsymbol{c}_1 + \boldsymbol{c}_2 + \cdots + \boldsymbol{c}_n) = T_1(\boldsymbol{c}_1) \oplus T_2(\boldsymbol{c}_2) \oplus \cdots \oplus T_n(\boldsymbol{c}_n)$$

when $\boldsymbol{c}_1 \in C_1$, $\boldsymbol{c}_2 \in C_2$, …, $\boldsymbol{c}_n \in C_n$. Show that, if $\mathcal{B}_i$ is a basis for C_i then

$$[T_1 \oplus T_2 \oplus \ldots \oplus T_n]_{\mathcal{B}_1 \oplus \mathcal{B}_2 \oplus \ldots \oplus \mathcal{B}_n} = [T_1]_{\mathcal{B}_1} \oplus [T_2]_{\mathcal{B}_2} \oplus \cdots \oplus [T_n]_{\mathcal{B}_n},$$

Here three new concepts are used: a direct sum of matrices, a direct sum of bases, and a direct sum of linear transformations.

Ex. 4 — Assume that T_1 and T_2 are matrices, and that the eigenvalues of T_1 are equal to those of T_2. What are the eigenvalues of $T_1 \oplus T_2$? Can you express the eigenvectors of $T_1 \oplus T_2$ in terms of those of T_1 and T_2?

Ex. 5 — Assume that A and B are square matrices which are invertible. Show that $A \oplus B$ is invertible, and that $(A \oplus B)^{-1} = A^{-1} \oplus B^{-1}$.

Ex. 6 — Let A, B, C, D be square matrices of the same dimensions. Show that $(A \oplus B)(C \oplus D) = (AC) \oplus (BD)$.

Ex. 7 — Assume that you run an m-level DWT on a vector of length r. What value of N does this correspond to? Note that an m-level DWT performs a change of coordinates from V_m to $V_0 \oplus W_0 \oplus W_1 \oplus \cdots \oplus W_{m-2} \oplus W_{m-1}$.

Ex. 8 — Run a 2-level DWT on the first 2^{17} sound samples of the audio file `castanets.wav`, and plot the values of the resulting DWT-coefficients. Compare the values of the coefficients from V_0 with those from W_0 and W_1.

Ex. 9 — In this exercise we will experiment with applying an m-level DWT to a sound file.

a. Write a function

```
function playDWTlower(m)
```

which

1. reads the audio file `castanets.wav`,
2. performs an m-level DWT to the first 2^{17} sound samples of $\boldsymbol{x}$ using the function `DWTHaarImpl`,
3. sets all wavelet coefficients representing detail to zero (i.e. keep only wavelet coefficients from V_0 in the decomposition $V_0 \oplus W_0 \oplus W_1 \oplus \cdots \oplus W_{m-2} \oplus W_{m-1}$),
4. performs an IDWT on the resulting coefficients using the function `IDWTHaarImpl`,
5. plays the resulting sound.

b. Run the function `playDWTlower` for different values of m. For which m can you hear that the sound gets degraded? How does it get degraded? Compare with what you heard through the function `playDFTlower` in Example 3.15, where you performed a DFT on the sound sample instead, and set some of the DFT coefficients to zero.

c. Do the sound samples returned by `playDWTlower` lie in $[-1, 1]$?

Ex. 10 — Attempt to construct a (nonzero) sound where the function `playDWTlower` form the previous exercise does not change the sound for $m = 1, 2$.

Ex. 11 — Repeat Exercise 9, but this time instead keep only wavelet coefficients from the detail spaces $W_0, W_1, \ldots$. Call the new function `playDWTlowerdifference`. What kind of sound do you hear? Can you recognize the original sound in what you hear?

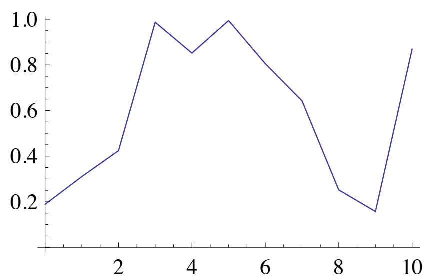
Ex. 12 — Compute the wavelet detail coefficients analytically for the functions in Example 5.22, i.e. compute the quantities $w_{m,n} = \int_0^N f(t)\psi_{m,n}(t)dt$ similarly to how this was done in Example 5.23.

Ex. 13 — Compute the wavelet detail coefficients analytically for the functions $f(t) = \left(\frac{t}{N}\right)^k$, i.e. compute the quantities $w_{m,n} = \int_0^N \left(\frac{t}{N}\right)^k \psi_{m,n}(t)dt$ similarly to how this was done in Example 5.23. How do these compare with the coefficients from the Exercise 12?

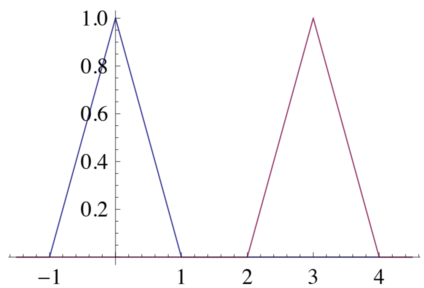
5.4 Wavelets constructed from piecewise linear functions

In Section 5.3 we started with the simple space of functions that are constant on each interval between two integers, which has a very simple orthonormal basis given by translates of the characteristic function of the interval $[0, 1)$. From this we constructed a so-called multiresolution analysis of successively refined spaces of piecewise constant functions that may be used to approximate any continuous function arbitrarily well. We then saw how a given function in a fine space could be projected orthogonally into the preceding coarser space. The computations were all taken care of with the Discrete Wavelet Transform.

In many situations, piecewise constant functions are too simple, and in this section we are going to extend the construction of wavelets to piecewise linear functions. The advantage is that piecewise linear functions are better for approximating smooth functions and data than piecewise constants, which should translate into smaller components (errors) in the detail spaces in many practical situations. As an example, this would be useful if we are interested in



(a) A piecewise linear function.



(b) The two functions $\phi(t)$ and $\phi(t-3)$.

Figure 5.12: Some piecewise linear functions.

compression.

5.4.1 Multiresolution analysis

Our experience from deriving Haar wavelets will guide us in the construction of piecewise linear wavelets. The first task is to define the underlying function spaces.

Definition 5.28 (Resolution spaces of piecewise linear functions)**.** The space V_m is the subspace of continuous functions on $\mathbb{R}$ which are periodic with period N, and linear on each subinterval of the form $[n2^{-m}, (n+1)2^{-m})$.

Any $f \in V_m$ is uniquely determined by its values on $[0, N)$. Figure 5.12 (a) shows an example of a piecewise linear function in V_0 on the interval $[0, 10]$. We note that a piecewise linear function in V_0 is completely determined by its value at the integers, so the functions that are 1 at one integer and 0 at all others are particularly simple and therefore interesting, see Figure 5.12 (b). These simple functions are all translates of each other and can therefore be built from one scaling function, as is required for a multiresolution analysis.

Recall that the *support* of a function f defined on a subset I of $\mathbb{R}$ is given by the closure of the set of points where the function is nonzero,

$$\operatorname{supp}(f) = \overline{\{t \in I \mid f(t) \neq 0\}}.$$

Lemma 5.29. Let the function ϕ be defined by

$$\phi(t) = \begin{cases} 1+t, & \text{if } -1 \leq t < 0; \\ 1-t, & \text{if } 0 \leq t < 1; \\ 0, & \text{otherwise}; \end{cases} \tag{5.33}$$

and for any $m \geq 0$ set

$$\phi_{m,n}(t) = \phi(2^m t - n) \quad \text{for } n = 0,\, 1,\, \ldots,\, 2^m N - 1,$$

or in vector notation

$$\boldsymbol{\phi}_m = (\phi_{m,0}, \phi_{m,1}, \ldots, \phi_{m,2^m N-1}).$$

The functions $\{\phi_{m,n}\}_{n=0}^{2^m N-1}$, restricted to the interval $[0,N]$, form a basis for the space V_m for this interval. In other words, the function ϕ is a scaling function for the spaces V_0, V_1, ... Moreover, the function $\phi_{0,n}(t)$ is the function in V_0 with smallest support that is nonzero at $t = n$.

Proof. The proof is similar for all the resolution spaces, so it is sufficient to consider the proof in the case of V_0. The function ϕ is clearly linear between each pair of neighbouring integers, and it is also easy to check that it is continuous. Its restriction to $[0,N]$ therefore lies in V_0. And as we noted above $\phi_{0,n}(t)$ is 0 at all the integers except at $t = n$ where its value is 1.

A general function f in V_0 is completely determined by its values at the integers in the interval $[0,N]$ since all straight line segments between neighbouring integers are then fixed. Note that we can also write f as

$$f(t) = \sum_{n=0}^{N-1} f(n)\phi_{0,n}(t) \tag{5.34}$$

since this function agrees with f at the integers in the interval $[0,N]$ and is linear on each subinterval between two neighbouring integers. This means that V_0 is spanned by the functions $\{\phi_{0,n}\}_{n=0}^{N-1}$. On the other hand, if f is identically 0, all the coefficients in (5.34) are also 0, so $\{\phi_{0,n}\}_{n=0}^{N-1}$ are linearly independent and therefore a basis for V_0.

Suppose that the function $g \in V_0$ has smaller support than $\phi_{0,n}$, but is nonzero at $t = n$. Then g must be identically zero either on $[n-1,n)$ or on $[n,n+1]$, since a straight line segment cannot be zero on just a part of an interval between integers. But then g cannot be continuous, which contradicts the fact the it lies in V_0. □

The function ϕ and its translates and dilates are often referred to as hat functions for obvious reasons.

A formula like (5.34) is also valid for functions in V_m.

Lemma 5.30. A function $f \in V_m$ may be written as

$$f(t) = \sum_{n=0}^{2^m N-1} f(n/2^m)\phi_{m,n}(t). \tag{5.35}$$

An essential property of a multiresolution analysis is that the spaces should be nested.

Lemma 5.31. The piecewise linear resolution spaces are nested,

$$V_0 \subset V_1 \subset \cdots \subset V_m \subset \cdots .$$

Proof. We only need to prove that $V_0 \subset V_1$ since the other inclusions are similar. But this is immediate since any function in V_0 is continuous, and linear on any subinterval in the form $[n/2, (n+1)/2)$. □

In the piecewise constant case, we saw in Lemma 5.5 that the scaling functions were automatically orthogonal since their supports did not overlap. This is not the case in the linear case, but we could orthogonalise the basis $\boldsymbol{\phi}_m$ with the Gram-Schmidt process from linear algebra. The disadvantage is that we lose the nice local behaviour of the scaling functions and end up with basis functions that are nonzero over all of $[0, N]$. And for most applications, orthogonality is not essential; we just need a basis.

Let us sum up our findings so far.

Observation 5.32. The spaces V_0, V_1, …, V_m, … form a multiresolution analysis generated by the scaling function ϕ.

The next step in the derivation of wavelets is to find formulas that let us express a function given in the basis $\boldsymbol{\phi}_0$ for V_0 in terms of the basis $\boldsymbol{\phi}_1$ for V_1.

Lemma 5.33. The function $\phi_{0,n}$ satisfies the relation

$$\phi_{0,n} = \frac{1}{2}\phi_{1,2n-1} + \phi_{1,2n} + \frac{1}{2}\phi_{1,2n+1}. \tag{5.36}$$

A general function g_0 in V_0 is also in V_1, and if

$$g_0 = \sum_{n=0}^{N-1} c_{0,n}\phi_{0,n} = \sum_{n=0}^{2N-1} c_{1,n}\phi_{1,n}$$

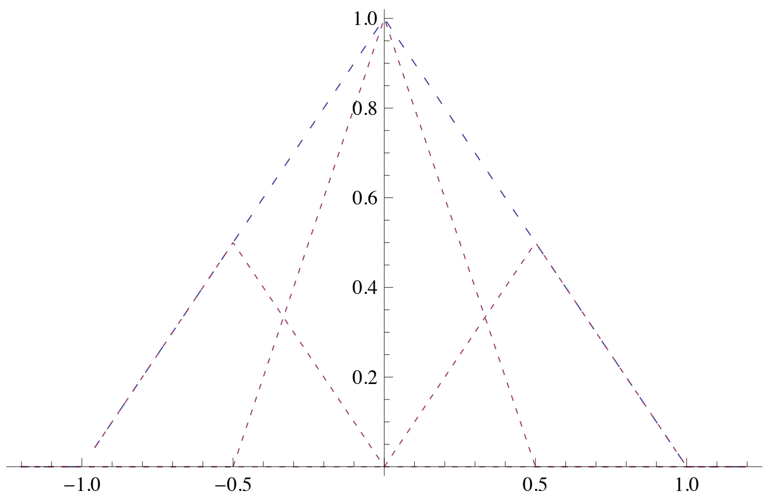
then

$$c_{1,2n} = c_{0,n}, \qquad \text{for } n = 0, 1, \ldots, N-1; \tag{5.37}$$

$$c_{1,2n+1} = (c_{0,n} + c_{0,(n+1) \bmod N})/2, \qquad \text{for } n = 0, 1, \ldots, N-1. \tag{5.38}$$

Proof. Since $\phi_{0,n}$ is in V_0 it may be expressed in the basis $\boldsymbol{\phi}_1$ with formula (5.35),

$$\phi_{0,n}(t) = \sum_{k=0}^{2N-1} \phi_{0,n}(k/2)\phi_{1,k}(t).$$



The relation (5.36) now follows since

$$\phi_{0,n}\big((2n-1)/2\big) = \phi_{0,n}\big((2n+1)/2\big) = 1/2, \quad \phi_{0,n}(2n/2) = 1,$$

and $\phi_{0,n}(k/2) = 0$ for all other values of k.

To prove (5.37) and (5.38), we use (5.36),

$$\begin{aligned} g_0 &= \sum_{n=0}^{N-1} c_{0,n}\phi_{0,n} \\ &= \sum_{n=0}^{N-1} c_{0,n}(\phi_{1,2n-1}/2 + \phi_{1,2n} + \phi_{1,2n+1}/2) \\ &= \sum_{n=0}^{N-1} c_{0,n}\phi_{1,2n} + \sum_{n=0}^{N-1} c_{0,(n+1) \bmod N}\phi_{1,2n+1}/2 + \sum_{n=0}^{N-1} c_{0,n}\phi_{1,2n+1}/2 \\ &= \sum_{n=0}^{N-1} c_{0,n}\phi_{1,2n} + \sum_{n=0}^{N-1} (c_{0,n} + c_{0,(n+1) \bmod N})\phi_{1,2n+1}/2, \end{aligned}$$

where we have performed a substitution of the form $n \to n+1$. The result now follows by comparing with $\sum_{n=0}^{2N-1} c_{1,n}\phi_{1,n}$. □

The relations in Lemma 5.33 can also be expressed in matrix form. If we set

$$\boldsymbol{c}_1 = (c_{1,n})_{n=0}^{2N-1}, \quad \boldsymbol{c}_1^e = (c_{1,2n})_{n=0}^{N-1}, \quad \boldsymbol{c}_1^o = (c_{1,2n+1})_{n=0}^{N-1},$$

we may write the equations (5.37) and (5.38) as

$$\begin{pmatrix} \boldsymbol{c}_1^e \\ \boldsymbol{c}_1^o \end{pmatrix} = \begin{pmatrix} I \\ A_0 \end{pmatrix} \boldsymbol{c}_0, \tag{5.39}$$

where I is the $N \times N$ identity matrix and A_0 is the $N \times N$ circulant Toeplitz matrix given by

$$A_0 = \frac{1}{2} \begin{pmatrix} 1 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 1 \\ 1 & 0 & 0 & \cdots & 0 & 1 \end{pmatrix}. \tag{5.40}$$

The formulas (5.37)–(5.38), or alternatively (5.39), show how a function in V_0 and can be represented in V_1. Analogous formulas let us rewrite a function in V_k in terms of the basis for V_{k+1}.

5.4.2 Detail spaces and wavelets

The next step in our derivation of wavelets for piecewise linear functions is the definition of the detail spaces. In the case of V_0 and V_1, we need to determine a space W_0 so that V_1 is the direct sum of V_0 and W_0. In the case of piecewise constants we started with a function g_1 in V_1, computed the least squares approximation g_0 in V_0, and then defined the space W_0 as the space of all possible error functions. This is less appealing in the linear case since we do not have an orthogonal basis for V_0.

As in the case of piecewise constants we start with a function g_1 in V_1, but we use an extremely simple approximation method, we simply drop every other coefficient.

Definition 5.34. Let g_1 be a function in V_1 given by

$$g_1 = \sum_{n=0}^{2N-1} c_{1,n}\ \phi_{1,n}. \tag{5.41}$$

The approximation $g_0 = S(g_1)$ in V_0 interpolates g_1 at the integers,

$$g_0(n) = g_1(n), \quad n = 0, 1, \ldots, N-1. \tag{5.42}$$

It is very easy to see that the coefficients of g_0 actually can be obtained by dropping every other coefficient:

Lemma 5.35. Let g_1 be given by (5.41) and suppose that $g_0 = \sum_{n=0}^{N-1} c_{0,n}\phi_{0,n}$ in V_0 interpolates g_1 at the integers in $0, \ldots, N-1$. Then the coefficients are given by

$$c_{0,n} = c_{1,2n}, \quad \text{for } n = 0, 1, \ldots, N-1, \tag{5.43}$$

and

$$S(\phi_{1,n}) = \begin{cases} \phi_{0,n/2}, & \text{if } n \text{ is an even integer;} \\ 0, & \text{otherwise.} \end{cases}$$

Once the method of approximation is determined, it is straightforward to determine the detail space as the space of error functions. With the notation from Definition 5.34, the error is given by $e_0 = g_1 - g_0$. Since g_0 interpolates g_1 at the integers, the error is 0 there,

$$e_0(n) = 0, \quad \text{for } n = 0, 1, \ldots, N-1.$$

Conversely, any function in V_1 which is 0 at the integers may be viewed as an error function in the above sense. This provides the basis for a precise description of the error functions.

Lemma 5.36. Suppose the function g_0 in V_0 interpolates a function g_1 in V_1 at the integers. Then the error $e_0 = g_1 - g_0$ lies in the space W_0 defined by

$$W_0 = \{f \in V_1 \mid f(n) = 0, \quad \text{for } n = 0, 1, \ldots, N-1.$$

A basis for W_0 is given by the wavelets $\{\psi_{0,n}\}_{n=0}^{N-1}$ defined by

$$\psi_{0,n} = \phi_{1,2n+1}, \quad \text{for } n = 0, 1, \ldots, N-1.$$

If $g_1 = \sum_{n=0}^{2N-1} c_{1,n}\ \phi_{1,n}$ is approximated by g_0 in V_0 as in Lemma 5.35, the error is given by $e_0 = \sum_{n=0}^{N-1} w_{0,n}\psi_{0,n}$ where

$$w_{0,n} = c_{1,2n+1} - \frac{1}{2}(c_{1,2n} + c_{1,(2n+2) \bmod 2N}). \tag{5.44}$$

Proof. We must show that any error function can be written in terms of the wavelets. First of all we note that the wavelets are linearly independent since their supports do not intersect. Let $g_1 \in V_1$ be as in (5.41) and let the $g_0 \in V_0$

be the approximation described by Lemma 5.35. Then the error is given by

$$
\begin{aligned}
e_0 &= g_1 - g_0 \\
&= \sum_{n=0}^{2N-1} c_{1,n}\phi_{1,n} - \sum_{n=0}^{N-1} c_{0,n}\phi_{0,n} \\
&= \sum_{n=0}^{N-1} c_{1,2n}\phi_{1,2n} + \sum_{n=0}^{N-1} c_{1,2n+1}\phi_{1,2n+1} - \sum_{n=0}^{N-1} c_{1,2n}\phi_{0,n} \\
&= \sum_{n=0}^{N-1} c_{1,2n}\phi_{1,2n} + \sum_{n=0}^{N-1} c_{1,2n+1}\phi_{1,2n+1} \\
&\qquad - \frac{1}{2}\sum_{n=0}^{N-1} c_{1,2n}\phi_{1,2n-1} - \sum_{n=0}^{N-1} c_{1,2n}\phi_{1,2n} - \frac{1}{2}\sum_{n=0}^{N-1} c_{1,2n}\phi_{1,2n+1} \\
&= \sum_{n=0}^{N-1} c_{1,2n}\phi_{1,2n} + \sum_{n=0}^{N-1} c_{1,2n+1}\phi_{1,2n+1} \\
&\qquad - \frac{1}{2}\sum_{n=0}^{N-1} c_{1,(2n+2) \bmod 2N}\phi_{1,2n+1} - \sum_{n=0}^{N-1} c_{1,2n}\phi_{1,2n} - \frac{1}{2}\sum_{n=0}^{N-1} c_{1,2n}\phi_{1,2n+1} \\
&= \sum_{n=0}^{N-1}\Big(c_{1,2n+1} - \frac{1}{2}(c_{1,2n} + c_{1,(2n+2) \bmod 2N})\Big)\phi_{1,2n+1}.
\end{aligned}
$$

In the third equation we split the sum in g_1 into even and odd terms and used the definition of g_0 in (5.43). In the next step we then rewrote g_0 in V_1 using formula (5.37), and finally we rewrote $\phi_{0,n}$ using formula (5.36). □

We now have all the ingredients to formulate an analog of Theorem 5.18 that describes how V_m can be expressed as a direct sum of V_{m-1} and W_{m-1}. The formulas for $m = 1$ generalise without change, except that the upper bound on the summation indices must be adjusted.

Theorem 5.37. The space V_m can be decomposed as the direct sum $V_m = V_{m-1} \oplus W_{m-1}$ where W_{m-1} is the space of all functions in V_m that are zero at the points $\{n/2^{m-1}\}_{n=0}^{N2^{m-1}-1}$. The space V_m has the two bases

$$\boldsymbol{\phi}_m = (\phi_{m,n})_{n=0}^{2^m N-1}$$

and

$$(\boldsymbol{\phi}_{m-1}, \boldsymbol{\psi}_{m-1}) = \big((\phi_{m-1,n})_{n=0}^{2^{m-1}N-1}, (\psi_{m-1,n})_{n=0}^{2^{m-1}N-1}\big).$$

If $g_m \in V_m$, $g_{m-1} \in V_{m-1}$, and $e_{m-1} \in W_{m-1}$ are given by

$$g_m = \sum_{n=0}^{2^m N-1} c_{m,n}\phi_{m,n},$$

$$g_{m-1} = \sum_{n=0}^{2^{m-1}N-1} c_{m-1,n}\phi_{m-1,n},$$

$$e_{m-1} = \sum_{n=0}^{2^{m-1}N-1} w_{m-1,n}\psi_{m-1,n},$$

and $g_m = g_{m-1} + e_{m-1}$, then the change of coordinates from the basis $\boldsymbol{\phi}_m$ to the basis $(\boldsymbol{\phi}_{m-1}, \boldsymbol{\psi}_{m-1})$ is given by

$$c_{m-1,n} = c_{m,2n},$$
$$w_{m-1,n} = c_{m,2n+1} - \big(c_{m,2n} + c_{m,(2n+2) \bmod 2N}\big)/2.$$

Conversely, the change of coordinates from the basis $(\boldsymbol{\phi}_{m-1}, \boldsymbol{\psi}_{m-1})$ to the basis $\boldsymbol{\phi}_m$ is given by

$$c_{m,2n} = c_{m-1,n}, \tag{5.45}$$
$$c_{m,2n+1} = w_{m-1,n} + (c_{m-1,n} + c_{m-1,n+1})/2. \tag{5.46}$$

The matrix notation in (5.39) may be generalised to cover Theorem 5.37. With the natural extension of the notation in (5.39) we see that IDWT given by (5.45) and (5.46) can be expressed as

$$\begin{pmatrix} \boldsymbol{c}_m^e \\ \boldsymbol{c}_m^o \end{pmatrix} = \begin{pmatrix} I & 0 \\ A_{m-1} & I \end{pmatrix} \begin{pmatrix} \boldsymbol{c}_{m-1} \\ \boldsymbol{w}_{m-1} \end{pmatrix} \tag{5.47}$$

where both identity matrices have dimension $N2^{m-1}$. The matrix A_{m-1} is a $\big(N2^{m-1}\big) \times \big(N2^{m-1}\big)$ matrix which is the natural generalisation of the matrix A_0 defined in (5.40). The DWT is simply the inverse of (5.39) and is given by

$$\begin{pmatrix} \boldsymbol{c}_{m-1} \\ \boldsymbol{w}_{m-1} \end{pmatrix} = \begin{pmatrix} I & 0 \\ -A_{m-1} & I \end{pmatrix} \begin{pmatrix} \boldsymbol{c}_m^e \\ \boldsymbol{c}_m^o \end{pmatrix}. \tag{5.48}$$

There is another simple expression for the DWT we will have use for. From Equation (5.36) and from the definition of ψ we have

$$\phi_{0,n} = \frac{1}{2}\phi_{1,2n-1} + \phi_{1,2n} + \frac{1}{2}\phi_{1,2n+1}$$
$$\psi_{0,n} = \phi_{1,2n+1}.$$

Again, it is custom to use the normalized functions $\phi_{m,n}(t) = 2^{1/2}\phi(2^m t - n)$.

Using these instead, the two equations above take the form

$$\begin{aligned}
\phi_{0,n} &= \frac{1}{2\sqrt{2}}\phi_{1,2n-1} + \frac{1}{\sqrt{2}}\phi_{1,2n} + \frac{1}{2\sqrt{2}}\phi_{1,2n+1} \\
\psi_{0,n} &= \frac{1}{\sqrt{2}}\phi_{1,2n+1}.
\end{aligned} \tag{5.49}$$

These two relations together give all columns in the change of coordinate matrix $P_{\boldsymbol{\phi}_1 \leftarrow \mathcal{C}_1}$, when the spaces $\boldsymbol{\phi}_m$, $\mathcal{C}_m$ instead are defined in terms of the function ψ, and the normalized ϕ. In particular, the first two columns in this matrix are

$$\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 \\ 1/2 & 1 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ 1/2 & 0 \end{pmatrix}. \tag{5.50}$$

The remaining columns are obtained by shifting this, as in a circulant Toeplitz matrix. Similarly we can compute the change of coordinate matrix the opposite way, $P_{\mathcal{C}_1 \leftarrow \boldsymbol{\phi}_1}$: Equations (5.49) can be written

$$\begin{aligned}
\frac{1}{\sqrt{2}}\phi_{1,2n} &= \phi_{0,n} - \frac{1}{2\sqrt{2}}\phi_{1,2n-1} - \frac{1}{2\sqrt{2}}\phi_{1,2n+1} \\
\frac{1}{\sqrt{2}}\phi_{1,2n+1} &= \psi_{0,n},
\end{aligned}$$

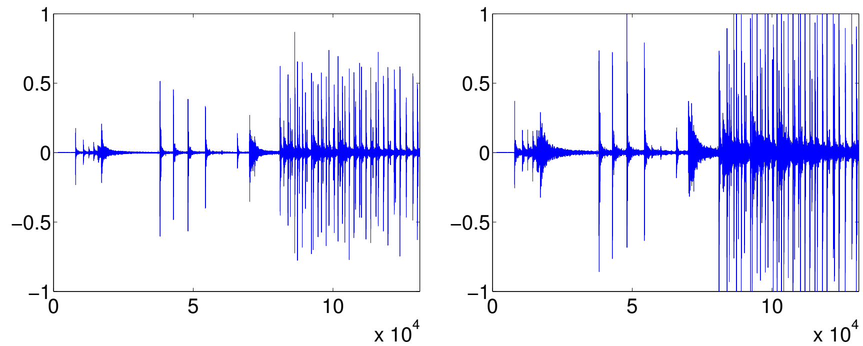
from which it follows that

$$\begin{aligned}
\phi_{1,2n} &= \sqrt{2}\phi_{0,n} - \frac{1}{2}\phi_{1,2n-1} - \frac{1}{2}\phi_{1,2n+1} \\
&= -\frac{\sqrt{2}}{2}\psi_{0,n-1} + \sqrt{2}\phi_{0,n} - \frac{\sqrt{2}}{2}\psi_{0,n} \\
\phi_{1,2n+1} &= \sqrt{2}\psi_{0,n},
\end{aligned}$$

which in the same way as above give the following two first columns in the change of coordinate matrix $P_{\mathcal{C}_1 \leftarrow \boldsymbol{\phi}_1}$:

$$\sqrt{2} \begin{pmatrix} 1 & 0 \\ -1/2 & 1 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ -1/2 & 0 \end{pmatrix}. \tag{5.51}$$

Also here, the remaining columns are obtained by shifting this, as in a circulant Toeplitz matrix.



(a) $m = 1$ (b) $m = 2$

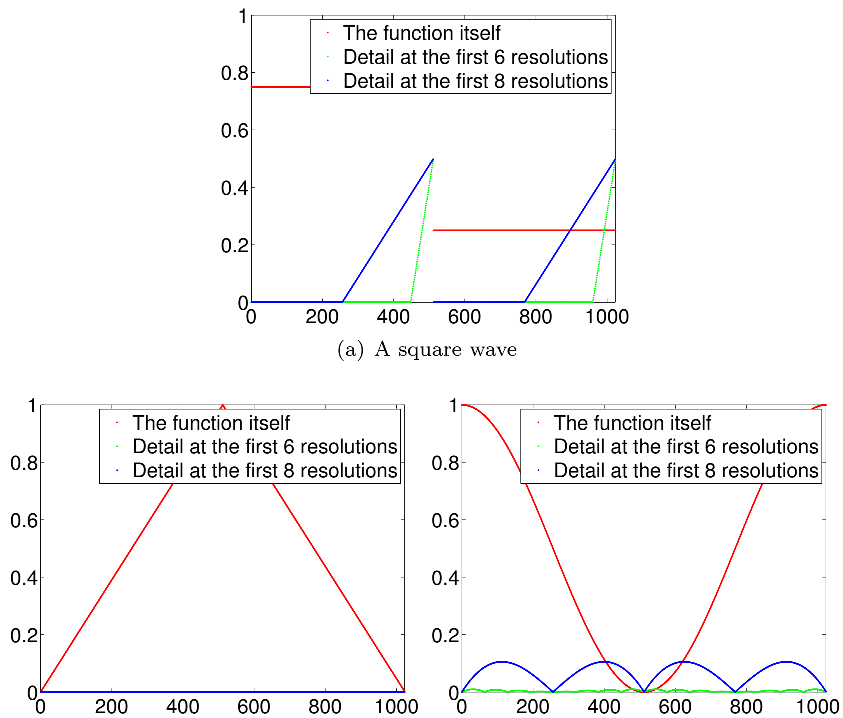
Figure 5.13: The error (i.e. the contribution from $W_0 \oplus W_1 \oplus \cdots \oplus W_{m-1}$) in the sound file `castanets.wav`, for different values of m.

Example 5.38. In Section 5.6 we will construct an algorithm which performs DWT/IDWT, for a general wavelet. In particular, this algorithm can be used for the wavelet we constructed in this section. Let us also for this wavelet plot the detail/error in the test audio file `castanets.wav` for different resolutions, as we did in Example 5.21. The result is shown in Figure 5.13. When comparing with Figure 5.10 we see much of the same, but it seems here that the error is bigger than before. In the next section we will try to explain why this is the case, and construct another wavelet based on piecewise linear functions which remedies this.

Example 5.39. Let us also repeat Exercise 5.22, where we plotted the detail/error at different resolutions, for the samples of a mathematical function. Figure 5.14 shows the new plot. With the square wave we see now that there is an error. The reason is that a piecewise constant function can not be represented exactly by piecewise linear functions, due to discontinuity. For the second function we see that there is no error. The reason is that this function is piecewise constant, so there is no error when we represent the function from the space V_0. With the third function, hoewever, we see an error.

Exercises for Section 5.4

Ex. 1 — Show that, for $f \in V_0$ we have that $[f]_{\phi_0} = (f(0), f(1), \ldots, f(N-1))$. This generalizes the result for piecewise constant functions.



(a) A square wave

(b) $f(t) = 1 - 2|1/2 - t/N|$

(c) $f(t) = 1/2 + \cos(2\pi t/N)/2$

Figure 5.14: The error (i.e. the contribution from $W_0 \oplus W_1 \oplus \cdots \oplus W_{m-1}$) for $N = 1024$ for different functions $f(t)$, for different values of m.

Ex. 2 — Show that

$$\langle \phi_{0,n}, \phi_{0,n} \rangle = \frac{2}{3}$$
$$\langle \phi_{0,n}, \phi_{0,n\pm 1} \rangle = \frac{1}{6}$$
$$\langle \phi_{0,n}, \phi_{0,n\pm k} \rangle = 0 \text{ for } k > 1.$$

As a consequence, the $\{\phi_{0,n}\}_n$ are neither orthogonal, nor have norm 1.

Ex. 3 — The convolution of two functions defined on $(-\infty, \infty)$ is defined by

$$(f * g)(x) = \int_{-\infty}^{\infty} f(t)g(x-t)dt.$$

Show that we can obtain the piecewise linear ϕ we have defined as $\phi = \chi_{[-1/2,1/2)} * \chi_{[-1/2,1/2)}$ (recall that $\chi_{[-1/2,1/2)}$ is the function which is 1 on $[-1/2, 1/2)$ and 0 elsewhere). This gives us a nice connection between the piecewise constant scaling function (which is similar to $\chi_{[-1/2,1/2)}$) and the piecewise linear scaling function in terms of convolution.

5.5 Alternative wavelets for piecewise linear functions

The direct sum decomposition that we derived in Section 5.4 was very simple, but also has its shortcomings. To see this, set $N = 1$ and consider the space V_{10}, which has dimension 2^{10}, which in most cases will mean that the function g_{10} will be a very good representation of the underlying data. However, when we compute g_{m-1} we just pick every other coefficient from g_m. By the time we get to g_0 we are just left with the first and last coefficient from g_{10}. In some situations this may be adequate, but usually not.

To address this shortcoming, let us return to the piecewise constant wavelet, and assume that $f \in V_m$. By the orthogonal decomposition theorem we have

$$f = \sum_{n=0}^{N-1} \langle f, \phi_{0,n} \rangle \phi_{0,n} + \sum_{r=0}^{m-1} \sum_{n=0}^{2^r N-1} \langle f, \psi_{r,n} \rangle \psi_{r,n}. \tag{5.52}$$

If f is s times differentiable, it can be represented as $f = P_s(x) + Q_s(x)$, where P_s is a polynomial of degree s, and Q_s is a function which is very small (P_s could for instance be a Taylor series expansion of f). If in addition $\langle t^k, \psi \rangle = 0$, for $k = 1, \ldots, s$, we have also that $\langle t^k, \psi_{r,t} \rangle = 0$ for $r \leq s$, so that $\langle P_s, \psi_{r,t} \rangle = 0$

also. This means that (5.52) can be written

$$
\begin{aligned}
f &= \sum_{n=0}^{N-1}\langle P_s+Q_s,\phi_{0,n}\rangle\phi_{0,n} + \sum_{r=0}^{m-1}\sum_{n=0}^{2^rN-1}\langle P_s+Q_s,\psi_{r,n}\rangle\psi_{r,n} \\
&= \sum_{n=0}^{N-1}\langle P_s+Q_s,\phi_{0,n}\rangle\phi_{0,n} + \sum_{r=0}^{m-1}\sum_{n=0}^{2^rN-1}\langle P_s,\psi_{r,n}\rangle\psi_{r,n} + \sum_{r=0}^{m-1}\sum_{n=0}^{2^rN-1}\langle Q_s,\psi_{r,n}\rangle\psi_{r,n} \\
&= \sum_{n=0}^{N-1}\langle f,\phi_{0,n}\rangle\phi_{0,n} + \sum_{r=0}^{m-1}\sum_{n=0}^{2^rN-1}\langle Q_s,\psi_{r,n}\rangle\psi_{r,n}.
\end{aligned}
$$

Here the first sum lies in V_0. We see that the wavelet coefficients from W_r are $\langle Q_s,\psi_{r,n}\rangle$, which are very small since Q_s is small. This means that the detail in the different spaces W_r is very small, which is exactly what we aimed for. Let us summarize this as follows:

Theorem 5.40 (Vanishing moments)**.** We say that ψ has k vanishing moments if the integrals $\int_{-\infty}^{\infty} t^l\psi(t)dt = 0$ for all $0 \le l \le k-1$. If a function $f \in V_m$ is r times differentiable, and ψ has r vanishing moments, then f can be approximated well from V_0. Moreover, the quality of this approximation improves when r increases.

It is also clear from the argument that if f is a polynomial of degree less than or equal to $k-1$ and ψ has k vanishing moments, then the wavelet detail coefficients are exactly 0. This theorem at least says what we have to aim for when the wavelet basis is orthonormal. The concept of vanishing moments also makes sense when the wavelet is not orthonormal, however. One can show that it is desirable to have many vanishing moments for other wavelets also. We we will not go into this (the wavelets used in practice turn out to be "almost" orthonormal, although they are not actually orthonormal. In such cases the computations above serve as good approximations, so that it is desirable to have many vanishing moments also here).

The Haar wavelet has one vanishing moment, since $\int_0^N \psi(t)dt = 0$ as we noted in Observation 5.15. It is an exercise to see that the Haar wavelet has only one vanishing moment, i.e. $\int_0^N t\psi(t)dt \neq 0$.

Now consider the wavelet we have used up to now for piecewise linear functions, i.e. $\psi(t) = \phi_{1,1}(t)$. Clearly this has no vanishing moments, since $\psi(t) \ge 0$ for all t. This is thus not a very good choice of wavelet. Let us see if we can construct an alternative function $\hat{\psi}$, which has two vanishing moments, i.e. one more than the Haar wavelet.

Idea 5.41. Adjust the wavelet construction in Theorem 5.37 so that the new wavelets $\{\hat{\psi}_{m-1,n}\}_{n=0}^{N2^m-1}$ in W_{m-1} satisfy

$$\int_0^N \hat{\psi}_{m-1,n}(t)\,dt = \int_0^N t\hat{\psi}_{m-1,n}(t)\,dt = 0, \tag{5.53}$$

for $n = 0, 1, \ldots, N2^m - 1$.

As usual, it is sufficient to consider what happens when V_1 is written as a direct sum of V_0 and W_0. From Idea 5.41 we see that we need to enforce two conditions for each wavelet function. If we adjust the wavelets in Theorem 5.37 by adding multiples of the two neighbouring hat functions, we have two free parameters,

$$\hat{\psi}_{0,n} = \psi_{0,n} - \alpha\phi_{0,n} - \beta\phi_{0,n+1} \tag{5.54}$$

that we may determine so that the two conditions in (5.53) are enforced. If we do this, we get the following result:

Lemma 5.42. The function

$$\hat{\psi}_{0,n}(t) = \psi_{0,n}(t) - \frac{1}{4}\big(\phi_{0,n}(t) + \phi_{0,n+1}(t)\big) \tag{5.55}$$

satisfies the conditions

$$\int_0^N \hat{\psi}_{0,n}(t)\,dt = \int_0^N t\hat{\psi}_{0,n}(t)\,dt = 0.$$

Using Equation (5.36), which stated that

$$\phi_{0,n}(t) = \frac{1}{2}\phi_{1,2n-1} + \phi_{1,2n} + \frac{1}{2}\phi_{1,2n+1} \tag{5.56}$$

we get

$$\begin{aligned}
\hat{\psi}_{0,n}(t) &= \psi_{0,n}(t) - \frac{1}{4}\big(\phi_{0,n}(t) + \phi_{0,n+1}(t)\big) \\
&= \phi_{0,2n+1}(t) - \frac{1}{4}\big(\frac{1}{2}\phi_{1,2n-1} + \phi_{1,2n} + \frac{1}{2}\phi_{1,2n+1} + \frac{1}{2}\phi_{1,2n+1} + \phi_{1,2n+2} + \frac{1}{2}\phi_{1,2n+3}\big) \\
&= -\frac{1}{8}\phi_{1,2n-1} - \frac{1}{4}\phi_{1,2n} + \frac{3}{4}\phi_{1,2n+1} - \frac{1}{4}\phi_{1,2n+2} - \frac{1}{8}\phi_{1,2n+3}. \qquad (5.57)
\end{aligned}$$

Note that what we did here is equivalent to finding the coordinates of $\hat{\psi}$ in the basis $\boldsymbol{\phi}_1$: Equation (5.55) says that

$$[\hat{\psi}]_{\boldsymbol{\phi}_0 \oplus \boldsymbol{\psi}_0} = (-1/4, -1/4, 0, \ldots, 0) \oplus (1, 0, \ldots, 0). \tag{5.58}$$

Since the IDWT is the change of coordinates from $\boldsymbol{\phi}_0 \oplus \boldsymbol{\psi}_0$ to $\boldsymbol{\phi}_1$, we could also have computed $[\hat{\psi}]_{\boldsymbol{\phi}_1}$ by taking the IDWT of $(-1/4, -1/4, 0, \ldots, 0) \oplus (1, 0, \ldots, 0)$. In the next section we will consider more general implementations of the DWT and the IDWT, which we thus can use instead of performing the computation above.

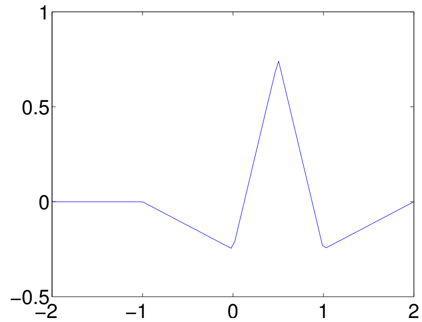


Figure 5.15: The function ψ we constructed as an alternative wavelet for piecewise linear functions.

Again, it is custom to use the normalized functions $\phi_{m,n}(t) = 2^{1/2}\phi(2^m t - n)$. Using these instead, the two equations above take the form

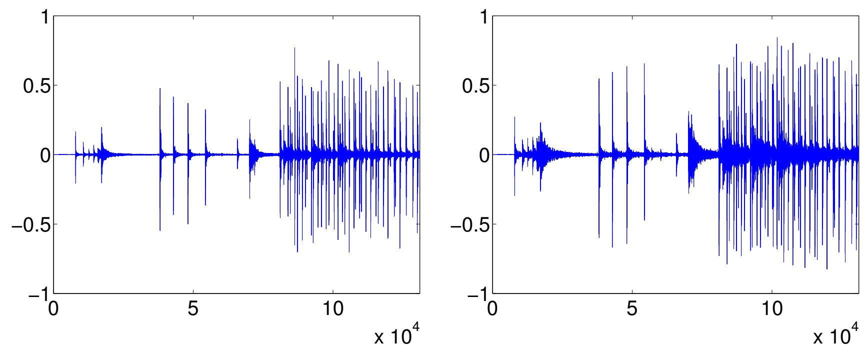
$$\phi_{0,n}(t) = \frac{1}{2\sqrt{2}}\phi_{1,2n-1} + \frac{1}{\sqrt{2}}\phi_{1,2n} + \frac{1}{2\sqrt{2}}\phi_{1,2n+1}$$
$$\hat{\psi}_{0,n}(t) = -\frac{1}{8\sqrt{2}}\phi_{1,2n-1} - \frac{1}{4\sqrt{2}}\phi_{1,2n} + \frac{3}{4\sqrt{2}}\phi_{1,2n+1} - \frac{1}{4\sqrt{2}}\phi_{1,2n+2} - \frac{1}{8\sqrt{2}}\phi_{1,2n+3}.$$

These two relations together give all columns in the change of coordinate matrix $P_{\boldsymbol{\phi}_1 \leftarrow \mathcal{C}_1}$, when the spaces $\boldsymbol{\phi}_m$, $\mathcal{C}_m$ instead are defined in terms of the function $\hat{\psi}$, and the normalized ϕ. In particular, the first two columns in this matrix are

$$\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & -1/4 \\ 1/2 & 3/4 \\ 0 & -1/4 \\ 0 & -1/8 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ 1/2 & -1/8 \end{pmatrix}. \tag{5.59}$$

The first column is the same as before, since there was no change in the definition of ϕ. The remaining columns are obtained by shifting this, as in a circulant Toeplitz matrix. Similarly we could compute the change of coordinate matrix the opposite way, $P_{\mathcal{C}_1 \leftarrow \boldsymbol{\phi}_1}$. We will explain how this can be done in the next section. The function ψ is plotted in Figure 5.15.

Example 5.43. Let us also plot the detail/error in the test audio file `castanets.wav` for different resolutions for our alternative wavelet, as we did in Example 5.21. The result is shown in Figure 5.16. Again, when comparing with Figure 5.10 we see much of the same. It is difficult to see an improvement from this figure. However, this figure also clearly shows a smaller error than the wavelet of



(a) $m = 1$

(b) $m = 2$

Figure 5.16: The error (i.e. the contribution from $W_0 \oplus W_1 \oplus \cdots \oplus W_{m-1}$) in the sound file `castanets.wav`, for different values of m.

the preceding section. A partial explanation is that the wavelet we now have constructed has two vanishing moments.

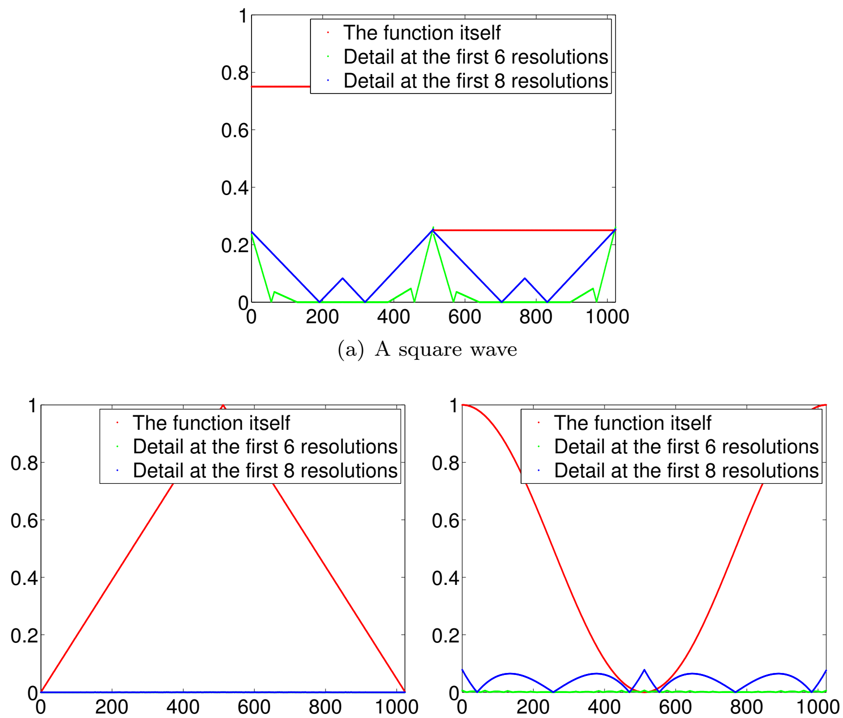
Example 5.44. Let us also repeat Exercise 5.22 for our alternative wavelet, where we plotted the detail/error at different resolutions, for the samples of a mathematical function. Figure 5.17 shows the new plot. Again for the square wave there is an error, which seems to be slightly lower than for the previous wavelet. For the second function we see that there is no error, as before. The reason is the same as before, since the function is piecewise constant. With the third function there is an error. The error seems to be slightly lower than for the previous wavelet, which fits well with the number of vanishing moments.

Exercises for Section 5.5

Ex. 1 — In this exercise we will show that there is a unique function on the form (5.54) which has two vanishing moments.

a. Show that, when $\hat{\psi}$ is defined by (5.54), we have that

$$\hat{\psi}(t) = \begin{cases} -\alpha t - \alpha & \text{for } -1 \leq t < 0 \\ (2+\alpha-\beta)t - \alpha & \text{for } 0 \leq t < 1/2 \\ (\alpha-\beta-2)t - \alpha + 2 & \text{for } 1/2 \leq t < 1 \\ \beta t - 2\beta & \text{for } 1 \leq t < 2 \\ 0 & \text{for all other } t \end{cases}$$



(a) A square wave

(b) $f(t) = 1 - 2|1/2 - t/N|$

(c) $f(t) = 1/2 + \cos(2\pi t/N)/2$

Figure 5.17: The error (i.e. the contribution from $W_0 \oplus W_1 \oplus \cdots \oplus W_{m-1}$) for $N = 1024$ for different functions $f(t)$, for different values of m.

b. Show that

$$\int_0^N \hat{\psi}(t)dt = \frac{1}{2} - \alpha - \beta$$
$$\int_0^N t\hat{\psi}(t)dt = \frac{1}{4} - \beta.$$

c. Explain why there is a unique function on the form (5.54) which has two vanishing moments, and that this function is given by Equation (5.55).

Ex. 2 — In the previous exercise we ended up with a lot of calculations to find α, β in Equation (5.54). Let us try to make a program which does this for us, and which also makes us able to generalize the result.

a. Define

$$a_k = \int_{-1}^{1} t^k(1 - |t|)dt$$
$$b_k = \int_0^2 t^k(1 - |t-1|)dt$$
$$e_k = \int_0^1 t^k(1 - 2|t-1/2|)dt,$$

for $k \geq 0$. Explain why finding α, β so that we have two vanishing moments in Equation 5.54 is equivalent to solving the following equation:

$$\begin{pmatrix} a_0 & b_0 \\ a_1 & b_1 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} e_0 \\ e_1 \end{pmatrix}$$

Write a program which sets up and solves this system of equations, and use this program to verify the values for α, β we previously have found. Hint: recall that you can integrate functions in Matlab with the function `quad`. As an example, the function $\phi(t)$, which is nonzero only on $[-1, 1]$, can be integrated as follows:

```
quad(@(t)t.^k.*(1-abs(t)),-1,1)
```

b. The procedure where we set up a matrix equation in a. allows for generalization to more vanishing moments. Define

$$\hat{\psi} = \psi_{0,0} - \alpha\phi_{0,0} - \beta\phi_{0,1} - \gamma\phi_{0,-1} - \delta\phi_{0,2}. \quad (5.60)$$

We would like to choose $\alpha, \beta, \gamma, \delta$ so that we have 4 vanishing moments. Define also

$$g_k = \int_{-2}^{0} t^k(1 - |t+1|)dt$$
$$d_k = \int_1^3 t^k(1 - |t-2|)dt$$

for $k \geq 0$. Show that $\alpha, \beta, \gamma, \delta$ must solve the equation

$$\begin{pmatrix} a_0 & b_0 & g_0 & d_0 \\ a_1 & b_1 & g_1 & d_1 \\ a_2 & b_2 & g_2 & d_2 \\ a_3 & b_3 & g_3 & d_3 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \\ \gamma \\ \delta \end{pmatrix} = \begin{pmatrix} e_0 \\ e_1 \\ e_2 \\ e_3 \end{pmatrix},$$

and solve this with Matlab.

c. Plot the function defined by (5.60), which you found in b.
Hint: If `t` is the vector of t-values, and you write `(t>=0).*(t<=1).*(1-2*abs(t-0.5))`, you get the points $\phi_{1,1}(t)$.

d. Explain why the coordinate vector of $\hat{\psi}$ in the basis $\boldsymbol{\phi}_0 \oplus \boldsymbol{\psi}_0$ is

$$[\hat{\psi}]_{\boldsymbol{\phi}_0 \oplus \boldsymbol{\psi}_0} = (-\alpha, -\beta, -\delta, 0, \ldots, 0 - \gamma) \oplus (1, 0, \ldots, 0).$$

Hint: you can also compare with Equation (5.58) here. The placement of $-\gamma$ may seem a bit strange here, and has to with that $\phi_{0,-1}$ is not one of the basis functions $\{\phi_{0,n}\}_{n=0}^{N-1}$. However, we have that $\phi_{0,-1} = \phi_{0,N-1}$, i.e. $\phi(t+1) = \phi(t - N + 1)$, since we always assume that the functions we work with have period N.

e. Sketch a more general procedure than the one you found in b., which can be used to find wavelet bases where we have even more vanishing moments.

Ex. 3 — It is also possible to add more vanishing moments to the Haar wavelet. Define

$$\hat{\psi} = \psi_{0,0} - a_0\phi_{0,0} - \cdots - a_{k-1}\phi_{0,k-1}.$$

Define also $c_{r,l} = \int_l^{l+1} t^r dt$, and $e_r = \int_0^1 t^r \psi(t) dt$.

a. Show that $\hat{\psi}$ has k vanishing moments if and only if $a_0, \ldots, a_{k-1}$ solves the equation

$$\begin{pmatrix} c_{0,0} & c_{0,1} & \cdots & c_{0,k-1} \\ c_{1,0} & c_{1,1} & \cdots & c_{1,k-1} \\ \vdots & \vdots & \vdots & \vdots \\ c_{k-1,0} & c_{k-1,1} & \cdots & c_{k-1,k-1} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{k-1} \end{pmatrix} = \begin{pmatrix} e_0 \\ e_1 \\ \vdots \\ e_{k-1} \end{pmatrix} \tag{5.61}$$

b. Write a function

```
function a=vanishingmomshaar(k)
```

which solves Equation 5.61, and returns $a_0, a_1, \ldots, a_{k-1}$ in the vector `a`.

5.6 Wavelets and filters

Up to now we have seen three different examples of wavelet bases: One for piecewise constant functions, and two for piecewise linear functions. In each case it turned out that the change of coordinate matrices $P_{\boldsymbol{\phi}_m \leftarrow \mathcal{C}_m}$, $P_{\mathcal{C}_m \leftarrow \boldsymbol{\phi}_m}$ had a special structure: They were obtained by repeating the first two columns in a circulant way, similarly to how we did in a circulant Toeplitz matrix. The matrices were not exactly circulant Toeplitz matrices, however, since there are two different columns repeating. The change of coordinate matrices occuring in the stages in a DWT are thus not digital filters, but they seem to be related. Let us start by giving these new matrices names:

Definition 5.45 (MRA-matrices)**.** An $N \times N$-matrix T, with N even, is called an MRA-matrix if the columns are translates of the first two columns in alternating order, in the same way as the columns of a circulant Toeplitz matrix.

From our previous calculations it is clear that, once ϕ and ψ are given through an MRA, the corresponding change of coordinate matrices will always be MRA-matrices. The MRA-matrices is our connection between matrices and wavelets. We would also like to state a similar connection with filters, i.e. show how the DWT could be implemented in terms of filters. We start with the following definition:

Definition 5.46. We denote by H_0 the (unique) filter with the same first row as $P_{\mathcal{C}_m \leftarrow \boldsymbol{\phi}_m}$, and by H_1 the (unique) filter with the same second row as $P_{\mathcal{C}_m \leftarrow \boldsymbol{\phi}_m}$.

Using this definition it is clear that

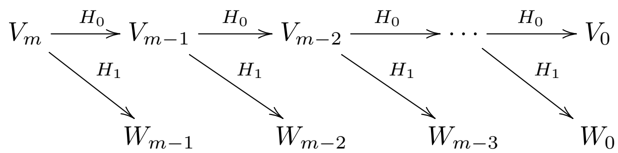
$$
\begin{aligned}
(P_{\mathcal{C}_m \leftarrow \boldsymbol{\phi}_m} \boldsymbol{c}_m)_k &= (H_0 \boldsymbol{c}_m)_k && \text{when } k \text{ is even} \\
(P_{\mathcal{C}_m \leftarrow \boldsymbol{\phi}_m} \boldsymbol{c}_m)_k &= (H_1 \boldsymbol{c}_m)_k && \text{when } k \text{ is odd}
\end{aligned}
$$

since the left hand side depends only on row k in the matrix $P_{\mathcal{C}_m \leftarrow \boldsymbol{\phi}_m}$, and this is equal to row k in H_0 (when k is even) or row k in H_1 (when k is odd). This means that $P_{\mathcal{C}_m \leftarrow \boldsymbol{\phi}_m} \boldsymbol{c}_m$ can be computed with the help of H_0 and H_1 as follows:

Theorem 5.47 (DWT expressed in terms of filters)**.** Let $\boldsymbol{c}_m$ be the coordinates in $\boldsymbol{\phi}_m$, and let H_0, H_1 be defined as above. Any stage in a DWT can ble implemented in terms of filters as follows:

1. Compute $H_0 \boldsymbol{c}_m$. The even-indexed entries in the result are the cordinates $\boldsymbol{c}_{m-1}$ in $\boldsymbol{\phi}_{m-1}$.
2. Compute $H_1 \boldsymbol{c}_m$. The odd-indexed entries in the result are the coordinates $\boldsymbol{w}_{m-1}$ in $\boldsymbol{\psi}_{m-1}$.

Note that this corresponds to applying two filters, and throwing away half of the results (since we only keep even-indexed and odd-indexed entries, respectively). In practice we do not compute the full application of the filter due to this. We can now complement Figure 5.3.1 by giving names to the arrows as follows:



Let us make a similar anlysis for the IDWT, and let us first make the following definition:

Definition 5.48. We denote by G_0 the (unique) filter with the same first column as $P_{\boldsymbol{\phi}_m\leftarrow\mathcal{C}_m}$, and by G_1 the (unique) filter with the same second column as $P_{\boldsymbol{\phi}_m\leftarrow\mathcal{C}_m}$.

These filters are uniquely determined, since any filter is uniquely determined from one of its columns. We can now write

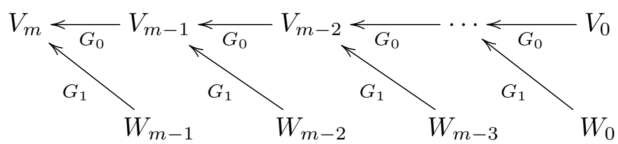
$$
\begin{aligned}
P_{\boldsymbol{\phi}_m\leftarrow\mathcal{C}_m}\begin{pmatrix} c_{m-1,0} \\ w_{m-1,0} \\ c_{m-1,1} \\ w_{m-1,1} \\ \cdots \\ c_{m-1,2^{m-1}N-1} \\ w_{m-1,2^{m-1}N-1} \end{pmatrix}
&= P_{\boldsymbol{\phi}_m\leftarrow\mathcal{C}_m}\left(\begin{pmatrix} c_{m-1,0} \\ 0 \\ c_{m-1,1} \\ 0 \\ \cdots \\ c_{m-1,2^{m-1}N-1} \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ w_{m-1,0} \\ 0 \\ w_{m-1,1} \\ \cdots \\ 0 \\ w_{m-1,2^{m-1}N-1} \end{pmatrix}\right) \\
&= P_{\boldsymbol{\phi}_m\leftarrow\mathcal{C}_m}\begin{pmatrix} c_{m-1,0} \\ 0 \\ c_{m-1,1} \\ 0 \\ \cdots \\ c_{m-1,2^{m-1}N-1} \\ 0 \end{pmatrix} + P_{\boldsymbol{\phi}_m\leftarrow\mathcal{C}_m}\begin{pmatrix} 0 \\ w_{m-1,0} \\ 0 \\ w_{m-1,1} \\ \cdots \\ 0 \\ w_{m-1,2^{m-1}N-1} \end{pmatrix} \\
&= G_0\begin{pmatrix} c_{m-1,0} \\ 0 \\ c_{m-1,1} \\ 0 \\ \cdots \\ c_{m-1,2^{m-1}N-1} \\ 0 \end{pmatrix} + G_1\begin{pmatrix} 0 \\ w_{m-1,0} \\ 0 \\ w_{m-1,1} \\ \cdots \\ 0 \\ w_{m-1,2^{m-1}N-1} \end{pmatrix}.
\end{aligned}
$$

Here we have split a vector into its even-indexed and odd-indexed elements, which correspond to the coefficients from $\boldsymbol{\phi}_{m-1}$ and $\boldsymbol{\psi}_{m-1}$, respectively. In the last equation, we replaced with G_0, G_1, since the multiplications with $P_{\boldsymbol{\phi}_m\leftarrow\mathcal{C}_m}$ depend only on the even and odd columns in that matrix (due to the zeros inserted), and these columns are equal in G_0, G_1. We can now state the following characterization of the inverse Discrete Wavelet transform:

Theorem 5.49 (IDWT expressed in terms of filters)**.** Let G_0, G_1 be defined as above. Any stage in an IDWT can be implemented in terms of filters as follows:

$$\boldsymbol{c}_m = G_0 \begin{pmatrix} c_{m-1,0} \\ 0 \\ c_{m-1,1} \\ 0 \\ \cdots \\ c_{m-1,2^{m-1}N-1} \\ 0 \end{pmatrix} + G_1 \begin{pmatrix} 0 \\ w_{m-1,0} \\ 0 \\ w_{m-1,1} \\ \cdots \\ 0 \\ w_{m-1,2^{m-1}N-1} \end{pmatrix}. \tag{5.62}$$

We can now also complement Figure 5.3.1 for the IDWT with named arrows as follows:



Note that the filters G_0, G_1 were defined in terms of the columns of $P_{\boldsymbol{\phi}_m \leftarrow \mathcal{C}_m}$, while the filters H_0, H_1 were defined in terms of the rows of $P_{\mathcal{C}_m \leftarrow \boldsymbol{\phi}_m}$. This difference is seen from the computations above to come from that the change of coordinates one way splits the coordinates into two parts, while the inverse change of coordinates performs the opposite.

There are two reasons why it is smart to express a wavelet transformation in terms of filters. First of all, it enables us to reuse theoretical results from the world of filters in the world of wavelets. Secondly, and perhaps most important, it enables us to reuse efficient implementations of filters in order to compute wavelet transformations. A lot of work has been done in order to establish efficient implementations of filters, due to their importance.

In Example 5.21 we argued that the elements in V_{m-1} correspond to frequencies at lower frequencies than those in V_m, since $V_0 = \text{Span}(\phi_{0,n})$ should be interpreted as content of lower frequency than the $\phi_{1,n}$, with $W_0 = \text{Span}(\psi_{0,n}$ the remaining high frequency detail. To elaborate more on this, we have that have that

$$\phi(t) = \sum_{n=0}^{2N-1} (G_0)_{n,0}\phi_{1,n}(t) \tag{5.63}$$

$$\psi(t) = \sum_{n=0}^{2N-1} (G_1)_{n,1}\psi_{1,n}(t)., \tag{5.64}$$

where $(G_k)_{i,j}$ are the entries in the matrix G_k. Similar equations are true for $\phi(t-k), \psi(t-k)$. Due to (5.63), the filter G_0 should have lowpass filter characteristics, since it extracts the information at lower frequencies. G_1 should have highpass filter characteristics du to (5.64). Let us verify this for the different wavelets we have defined up to now.

5.6.1 Frequency response for the Haar Wavelet

For the Haar wavelet we saw that, in $P_{\phi_m \leftarrow \mathcal{C}_m}$, the matrix

$$\begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix}$$

repeated along the diagonal. From this it is clear that

$$\begin{aligned} G_0 &= \{\underline{1/\sqrt{2}}, 1/\sqrt{2}\} \\ G_1 &= \{1/\sqrt{2}, \underline{-1/\sqrt{2}}\}. \end{aligned}$$

We have seen these filters previously: G_0 is a movinge average filter (of two elements), while G_1 is a bass-reducing filter (up to multiplication with a constant). We compute their frequency response as

$$\begin{aligned} \lambda_{G_0}(\omega) &= \frac{1}{\sqrt{2}} + \frac{1}{\sqrt{2}} e^{-i\omega} = \sqrt{2} e^{-i\omega/2} \cos(\omega/2) \\ \lambda_{G_1}(\omega) &= \frac{1}{\sqrt{2}} e^{i\omega} - \frac{1}{\sqrt{2}} = \sqrt{2} i e^{i\omega/2} \sin(\omega/2). \end{aligned}$$

The magnitude of these are plotted in Figure 5.18, where the lowpass/highpass characteristics are clearly seen. The two frequency responses seem also to be the same, except for a shift by π in frequency. We will show later that this is not coincidental. In this case we also have that

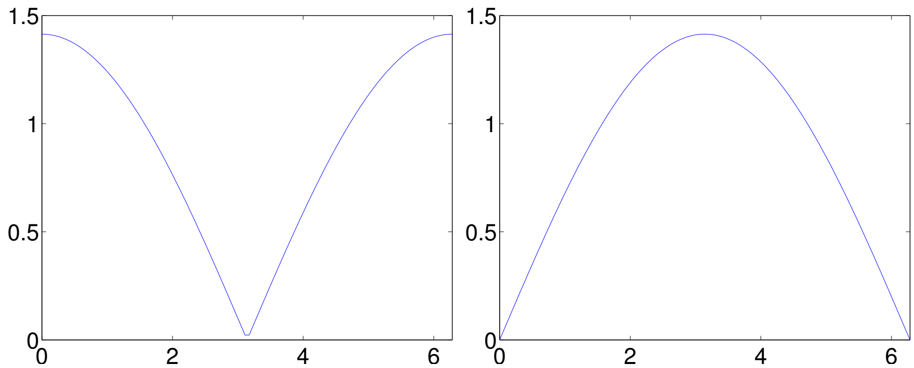
$$\begin{aligned} H_0 &= \{\underline{1/\sqrt{2}}, 1/\sqrt{2}\} \\ H_1 &= \{1/\sqrt{2}, \underline{-1/\sqrt{2}}\}, \end{aligned}$$

so that the frequency responses for the DWT have the same lowpass/highpass characteristics.

5.6.2 Frequency responses for wavelets of piecewise linear functions

For the first wavelet for piecewise linear functions we looked at in the previous section, Equation (5.50) gives that

$$\begin{aligned} G_0 &= \frac{1}{\sqrt{2}} \{1/2, \underline{1}, 1/2\} \\ G_1 &= \frac{1}{\sqrt{2}} \{\underline{1}\}. \end{aligned} \tag{5.65}$$



(a) $\lambda_{G_0}(\omega)$ (b) $\lambda_{G_1}(\omega)$

Figure 5.18: The frequency responses for the MRA of piecewise constant functions.

G_0 is again a filter we have seen before: Up to multiplication with a constant, it is the treble-reducing filter with values from row 2 of Pascal's triangle. The frequency responses are thus

$$\begin{aligned}\lambda_{G_0}(\omega) &= \frac{1}{2\sqrt{2}}e^{i\omega} + \frac{1}{\sqrt{2}} + \frac{1}{2\sqrt{2}}e^{-i\omega} = \frac{1}{\sqrt{2}}(\cos\omega + 1)\\ \lambda_{G_1}(\omega) &= \frac{1}{\sqrt{2}}.\end{aligned}$$

$\lambda_{G_1}(\omega)$ thus has magnitude $\frac{1}{\sqrt{2}}$ at all points. The magnitude of $\lambda_{G_0}(\omega)$ is plotted in Figure 5.19. Comparing with Figure 5.18 we see that here also the frequency response has a zero at π. The frequency response seems also to be flatter around π. For the DWT, Equation (5.51) gives us

$$\begin{aligned}H_0 &= \sqrt{2}\{\underline{1}\}\\ H_1 &= \sqrt{2}\{-1/2, \underline{1}, -1/2\}.\end{aligned} \tag{5.66}$$

We see that, up to a constant, H_1 is obtained from G_0 by adding an alternating sign. We know from before that this turns a lowpass filter into a highpass filter, so that H_1 is a highpass filter (it is a bass-reducing filter with values taken from row 2 of Pascals triangle).

Let us compare with the alternative wavelet we used for piecewise linear functions. In Equation (5.59) we wrote down the first two columns in $P_{\phi_m \leftarrow \mathcal{C}_m}$. This gives us that the two filters are

$$\begin{aligned}G_0 &= \frac{1}{\sqrt{2}}\{1/2, \underline{1}, 1/2\}\\ G_1 &= \frac{1}{\sqrt{2}}\{-1/8, -1/4, \underline{3/4}, -1/4, -1/8\}.\end{aligned} \tag{5.67}$$

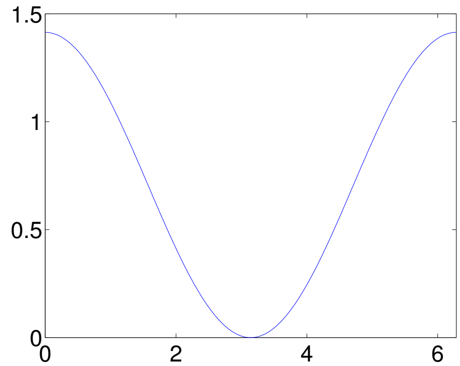


Figure 5.19: The frequency response $\lambda_{G_0}(\omega)$ for the first choice of wavelet for piecewise linear functions

Here G_0 was as before since we use the same scaling function, but G_1 was changed. We also have to find the filters H_0, H_1. It can be shown that (although we do not prove this), if $g_{0,n}$, $g_{1,n}$, $h_{0,n}$, $h_{1,n}$ are the filter coefficients for the filters, then

$$\begin{aligned} h_{0,n} &= \alpha(-1)^n g_{1,n} \\ h_{1,n} &= \alpha(-1)^n g_{0,n}, \end{aligned} \tag{5.68}$$

where $\alpha = \frac{1}{\sum_n g_{0,n} g_{1,n}}$. In other words, the filters are the same as G_0, G_1 up to multiplication by a constant, and an alternating sign. This means, more generally, that H_1 is a highpass filter when G_0 is a lowpass filter, and that H_0 is a lowpass filter when G_1 is a highpass filter. In this case, this means that

$$\alpha = \frac{1}{\frac{1}{2}\left(-\frac{1}{2}\left(-\frac{1}{4}\right) + 1\cdot\frac{3}{4} - \frac{1}{2}\left(-\frac{1}{4}\right)\right)} = 2,$$

so that

$$\begin{aligned} H_0 &= \sqrt{2}\{-1/8, 1/4, \underline{3/4}, 1/4, -1/8\} \\ H_1 &= \sqrt{2}\{-1/2, \underline{1}, -1/2\}. \end{aligned} \tag{5.69}$$

We now have that

$$\begin{aligned} \lambda_{G_1}(\omega) &= -1/(8\sqrt{2})e^{2i\omega} - 1/(4\sqrt{2})e^{i\omega} + 3/(4\sqrt{2}) - 1/(4\sqrt{2})e^{-i\omega} - 1/(8\sqrt{2})e^{-2i\omega} \\ &= -\frac{1}{4\sqrt{2}}\cos(2\omega) - \frac{1}{2\sqrt{2}}\cos\omega + \frac{3}{4\sqrt{2}}. \end{aligned}$$

The magnitude of $\lambda_{G_1}(\omega)$ is plotted in Figure 5.20. Clearly, G_1 now has highpass characteristics, while the lowpass characteristic of G_0 has been preserved. The filters G_0, G_1, H_0, H_1 are particularly important in applications: Apart from the

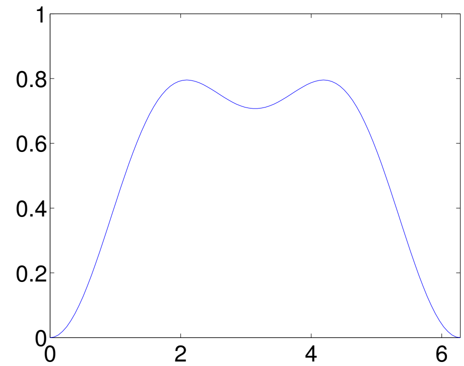


Figure 5.20: The frequency response $\lambda_{G_1}(\omega)$ for the alternative wavelet for piecewise linear functions.

scaling factors $1/\sqrt{2}$, $\sqrt{2}$ in front, we see that the filter coefficients are all dyadic fractions, i.e. they are on the form $\beta/2^j$. Arithmetic operations with dyadic fractions can be carried out exactly on a computer, due to representations as binary numbers in computers. These filters are thus important in applications, since they can be used as transformations for lossless coding. The same argument can be made for the Haar wavelet, but this wavelet had one less vanishing moment.

5.6.3 Filter-based algorithm for the DWT and the IDWT

From the analysis in this section, we see that we can implement DWT/IDWT based on expressions for the corresponding filters. This opens up for other opportunities also, in that we can start altogether by providing filters G_0, G_1, H_0, H_1, and construct MRA-matrices with the same even/odd-indexed rows/-columns as these filters (as in Theorem 5.47 and Theorem 5.49). If we can find such filters so that the corresponding MRA-matrices invert each other, we can use them in implementations of the DWT/IDWT, even though we have no idea what the underlying functions ϕ, ψ may be, or if such functions exist at all.

This approach will be made in the following. To be more precise, we will provide filters G_0, G_1, H_0, H_1 which are used in practice, and where it is known that the corresponding MRA-matrices invert each other, and apply these in the algorithms sketched in Theorem 5.47 and Theorem 5.49. We will restrict ourself to the case where the filters G_0, G_1, H_0, H_1 all are symmetric. As can be seen from the filter expressions, this was the case for all filters we have looked at, except the Haar wavelet. We have already implemented the Haar wavelet, however. Symmetric filters are also very common in practice. A reason for this is that MRA-matrices based on symmetric filters also can be shown to preserve symmetric vectors, so that they share some of the desirable properties of symmetric filters (which lead us to the definition of the DCT).

The algorithm for the DWT/IDWT is essentially a matrix/vector multiplication, and this can be computed entry by entry once we have the rows of the matrices. With the DWT we have these rows, since the rows in $P_{\mathcal{C}_m\leftarrow\phi_m}$ are given by the rows of H_0, H_1. In Exercise 3.4.3, we implemented a symmetric filter, which took the filter coefficients as input. Another thing we need is to extend this implementation so that it works for the case where the first two rows, instead of only the first row, are given. In Exercise 3 we take you through the steps in finding the formulas for this. This enables us to write an algorithm for multiplying with an MRA matrix based on symmetric filters. You will be spared writing this algorithm, and you can assume that the function `y=rowsymmmratrans(a0,a1,x)` performs this task. Here `x` represents the vector we want to multiply with the MRA-matrix, and `y` represents the result. The parameters `a0,a1` require som explanation: They represent the filter coefficients for the filters which have the same first/second rows as the MRA-matrix, respectively. This is most easily explained for the MRA-matrix for the DWT, since here these filters are H_0 and H_1. Since H_0, H_1 are symmetric filters, they can be written on the form

$$
\begin{aligned}
H_0 &= \{h_{0,-k_1},\ldots,h_{0,-1}\underline{h_{0,0}},h_{0,1},\ldots,h_{0,k_0}\}\\
H_1 &= \{h_{1,-k_1},\ldots,h_{1,-1}\underline{h_{1,0}},h_{1,1},\ldots,h_{1,k_1}\}
\end{aligned}
$$

Since the negative-indexed filter coefficients are equal to the positive-indexed filter coefficients, there is no need to provide them to the function `rowsymmmratrans`. It will therefore be assumed that the input to `rowsymmmratrans` is

$$
\begin{aligned}
\texttt{a0} &= (h_{0,0},h_{0,1},\ldots,h_{0,k_0})\\
\texttt{a1} &= (h_{1,0},h_{1,1},\ldots,h_{1,k_1}),
\end{aligned}
$$

i.e. only the filter coefficients with index ≥ 0 are included in `a0` and `a1`. For the IDWT the situation is different: here only the columns of $P_{\phi_m\leftarrow\mathcal{C}_m}$ are given (and in terms of the columns of G_0, G_1). We therefore first need a step where we translate the column representation of $P_{\phi_m\leftarrow\mathcal{C}_m}$ to a row representation of the same matrix. This is a straightforward task, however a bit tedious. You will be spared writing this code also, and can take for granted that the function `[a0,a1]=changecolumnrows(g0,g1)` performs this task. The parameters `g0,g1` are best explained in terms of the IDWT: If G_0, G_1 are the symmetric filters in the IDWT, they are first written on the form

$$
\begin{aligned}
G_0 &= \{g_{0,-l_1},\ldots,g_{0,-1}\underline{g_{0,0}},g_{0,1},\ldots,g_{0,l_0}\}\\
G_1 &= \{g_{1,-l_1},\ldots,g_{1,-1}\underline{g_{1,0}},g_{1,1},\ldots,g_{1,l_1}\},
\end{aligned}
$$

and we write as above

$$
\begin{aligned}
\texttt{g0} &= (g_{0,0},g_{0,1},\ldots,g_{0,l_0})\\
\texttt{g1} &= (g_{1,0},g_{1,1},\ldots,g_{1,l_1}).
\end{aligned}
$$

The vectors a0,a1 returned by changecolumnrows are then the row representation which is accepted by the unction rowsymmmratrans.

Once we have the functions rowsymmmratrans and changecolumnrows, the DWT and the IDWT can easily be computed. Exercises 4 and 5 will talk you through the steps in this process. There is a very good reason for encapsulating the filtering operations inside the function rowsymmmratrans: it can hide the details of highly optimized implementations of different types of filters.

Example 5.50. In Exercise 8 you will be asked to implement a function playDWTfilterslower which plays the low-resolution approximations to our audio test file, for any type of wavelet, using the functions we have described. With this function we can play the result for all the wavelets we have considered up to now, in succession, and at a given resolution, with the following code:

```
function playDWTall(m)
disp('Haar wavelet');
playDWTlower(m);
disp('Wavelet for piecewise linear functions');
playDWTfilterslower(m,[sqrt(2)],...
    [sqrt(2) -1/sqrt(2)],...
    [1/sqrt(2) 1/(2*sqrt(2))],...
    [1/sqrt(2)]);
disp('Wavelet for piecewise linear functions, alternative  version');
playDWTfilterslower(m,[3/(2*sqrt(2)) 1/(2*sqrt(2)) -1/(4*sqrt(2))],...
   [sqrt(2) -1/sqrt(2)],...
   [1/sqrt(2) 1/(2*sqrt(2))],...
   [3/(4*sqrt(2)) -1/(4*sqrt(2)) -1/(8*sqrt(2))]);
```

The call to playDWTlower first plays the result, using the Haar wavelet. The code then moves on to the piecewise linear wavelet. From Equation (5.66) we first see that

$$\texttt{h0} = (h_{0,0}, h_{0,1}, \ldots, h_{0,k_0}) = (\sqrt{2}) \tag{5.70}$$
$$\texttt{h1} = (h_{1,0}, h_{1,1}, \ldots, h_{1,k_1}) = (\sqrt{2}, -\sqrt{2}/2), \tag{5.71}$$

and from Equation (5.65) we see that

$$\texttt{g0} = (g_{0,0}, g_{0,1}, \ldots, g_{0,l_0}) = (1/\sqrt{2}, 1/(2\sqrt{2})) \tag{5.72}$$
$$\texttt{g1} = (g_{1,0}, g_{1,1}, \ldots, g_{1,l_1}) = (1/\sqrt{2}). \tag{5.73}$$

These explain the parameters to the call to playDWTfilterslower for the piecewise linear wavelet. The code then moves to the alternative piecewise linear wavelet. From Equation (5.69) we see that

$$\texttt{h0} = (h_{0,0}, h_{0,1}, \ldots, h_{0,k_0}) = (3\sqrt{2}/4, \sqrt{2}/4, -\sqrt{2}/8)$$
$$\texttt{h1} = (h_{1,0}, h_{1,1}, \ldots, h_{1,k_1}) = (\sqrt{2}, -\sqrt{2}/2),$$

and from Equation (5.67) we see that

$$\begin{aligned}
\texttt{g0} &= (g_{0,0}, g_{0,1}, \ldots, g_{0,l_0}) = (1/\sqrt{2}, 1/(2\sqrt{2})) \\
\texttt{g1} &= (g_{1,0}, g_{1,1}, \ldots, g_{1,l_1}) = (3/(4\sqrt{2}), -1/(4\sqrt{2}), -1/(8\sqrt{2})).
\end{aligned}$$

These explain the parameters to the call to `playDWTfilterslower` for the alternative piecewise linear wavelet.

Exercises for Section 5.6

Ex. 1 — Find two symmetric filters, so that the corresponding MRA-matrix, constructed with alternating rows from these two filters, is not symmetric.

Ex. 2 — Assume that an MRA-matrix is symmetric. Show hat the corresponding filters are also symmetric.

Ex. 3 — Assume that G is an MRA-matrix where the rows repeated are $\boldsymbol{a}^{(0)}$, $\boldsymbol{a}^{(1)}$ (symmetric around 0). Assume that their supports are $[-E_0, E_0]$ and $[-E_1, E_1]$, respectively. Show that $y_n = (G\boldsymbol{x})_n$ can be computed as follows, depending on n:

a. n even: The formulas (3.33)-(3.35) you derived in Exercise 3.4.3 can be used, with $T_{0,k}$ replaced with $\boldsymbol{a}^{(0)}$, E replaced by E_0.

b. n odd: The formulas (3.33)-(3.35) you derived in Exercise 3.4.3 can be used, with $T_{0,k}$ replaced with $\boldsymbol{a}^{(1)}$, E replaced by E_1.

Ex. 4 — Write a function

```
function xnew=DWTImpl(h0,h1,x,m)
```

which takes a signal `x` of length N, computes the transforms $F_1, ..., F_{m-1}$, and computes the coordinate of `x` in the basis $V_0 \oplus W_0 \oplus W_1 \oplus \cdots \oplus W_{m-1}$. Your function should call the function `rowsymmmratrans` to achieve this. Remember that you have to sort the even and odd outputs after calling that function, before you apply the next step. You can assume that the signal `x` has length 2^m.

Ex. 5 — Write a function

```
function x=IDWTImpl(g0,g1,xnew,m)
```

which recovers the coordinates in the basis V_m from those in the basis $V_0 \oplus W_0 \oplus W_1 \oplus \cdots \oplus W_{m-1}$. Your function should call the function `changecolumnrows`, and the function `rowsymmmratrans`.

Ex. 6 — In this exercise we will practice setting up the parameters `h0,h1,g0,g1` which are used in the calls to `DWTImpl` and `IDWTImpl`.

a. Assume that one stage in a DWT is given by the MRA-matrix

$$P_{\mathcal{C}_1 \leftarrow \phi_1} = \begin{pmatrix} 1/5 & 1/5 & 1/5 & 0 & 0 & 0 & \cdots & 0 & 1/5 & 1/5 \\ -1/3 & 1/3 & -1/3 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 1/5 & 1/5 & 1/5 & 1/5 & 1/5 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & -1/3 & 1/3 & -1/3 & 0 & \cdots 0 & 0 & & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix}$$

Write down the compact form for the corresponding filters H_0, H_1, and compute and plot the frequency responses. Are the filters symmetric? If so, also write down the parameters `h0,h1` you would use for this matrix in a call to `DWTImpl`.

b. Assume that one stage in the IDWT is given by the MRA-matrix

$$P_{\phi_1 \leftarrow \mathcal{C}_1} = \begin{pmatrix} 1/2 & -1/4 & 0 & 0 & \cdots \\ 1/4 & 3/8 & 1/4 & 1/16 & \cdots \\ 0 & -1/4 & 1/2 & -1/4 & \cdots \\ 0 & 1/16 & 1/4 & 3/8 & \cdots \\ 0 & 0 & 0 & -1/4 & \cdots \\ 0 & 0 & 0 & 1/16 & \cdots \\ 0 & 0 & 0 & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots \\ 1/4 & 1/16 & 0 & 0 & \cdots \end{pmatrix}$$

Write down the compact form for the filters G_0, G_1, and compute and plot the frequency responses. Are the filters symmetric? If so, also write down the parameters `g0,g1` you would use for this matrix in a call to `IDWTImpl`.

Ex. 7 — Let us also practice on writing down the change of coordinate matrices from the parameters `h0,h1,g0,g1`.

a. Assume that `h0=[3/8 1/4 1/16]` and `h1=[1/2 -1/4]`. Write down the compact form for the filters H_0, H_1. Plot the frequency responses and verify that H_0 is a lowpass filter, and that H_1 is a highpass filter. Also write down the change of coordinate matrix $P_{\mathcal{C}_1 \leftarrow \phi_1}$ for the wavelet corresponding to these filters.

b. Assume that `g0=[1/3 1/3]` and `g1=[1/5 -1/5 1/5]`. Write down the compact form for the filters G_0, G_1. Plot the frequency responses and verify that G_0 is a lowpass filter, and that G_1 is a highpass filter. Also write down the change of coordinate matrix $P_{\phi_1 \leftarrow \mathcal{C}_1}$ for the wavelet corresponding to these filters.

Ex. 8 — Write a function

```
function playDWTfilterslower(m,h0,h1,g0,g1)
```

which reimplements the function `playDWTlower` from Exercise 5.3.9 so that it takes as input the positive parts of the four different filters as in Example 5.50. Listen to the result using the different wavelets we have encountered and for different m, using the code from Example 5.50. Can you hear any difference from the Haar wavelet? If so, which wavelet gives the best sound quality?

Ex. 9 — In this exercise we will change the code in Example 5.50 so that it instead only plays the contribution from the detail spaces (i.e. $W_0 \oplus W_1 \oplus \cdots \oplus W_{m-1}$).

a. Reimplement the function you made in Exercise 8 so that it instead plays the contribution from the detail spaces. Call the new function `playDWTfilterslowerdifference`.

b. In Exercise 5.3.11 we implemented a function `playDWTlowerdifference` for listening to the detail/error when the Haar wavelet is used. In the function `playDWTall` from Example 5.50, replace `playDWTlower` and `playDWTfilterslower` with `playDWTlowerdifference` and `playDWTfilterslowerdifference`. Describe the sounds you hear for different m. Try to explain why the sound seems to get louder when you increase m.

Ex. 10 — Let us return to the piecewise linear wavelet from Exercise 5.5.2.

a. With $\hat{\psi}$ as defined as in Exercise 5.5.2 b., compute the coordinates of $\hat{\psi}$ in the basis $\boldsymbol{\phi}_1$ (i.e. $[\hat{\psi}]_{\boldsymbol{\phi}_1}$) with $N = 16$, i.e. compute the IDWT of

$$[\hat{\psi}]_{\boldsymbol{\phi}_0 \oplus \boldsymbol{\psi}_0} = (-\alpha, -\beta, -\delta, 0, 0, 0, 0, -\gamma) \oplus (1, 0, 0, 0, 0, 0, 0, 0),$$

which is the coordinate vector you computed in Exercise 5.5.2 d.. For this, you should use the function `IDWTImpl` from Exercise 5, with parameters being the filters G_0, G_1, given as described by `g0,g1` by equations (5.72)-(5.73) in Example 5.50.

b. If we redefine the basis $\mathcal{C}_1$ from $\{\phi_{0,0}, \psi_{0,0}, \phi_{0,1}, \psi_{0,1}, \ldots\}$, to $\{\phi_{0,0}, \hat{\psi}_{0,0}, \phi_{0,1}, \hat{\psi}_{0,1}, \ldots\}$, the vector you obtained in a. gives us an expression for the second column in $P_{\boldsymbol{\phi}_1 \leftarrow \mathcal{C}_1}$. After redefining the basis like this, the corresponding filter G_1 has changed from that of the piecewise linear wavelet we started with. Use Matlab to so state the new filter G_1 with our compact filter notation. Also, plot its frequency response. Hint: Here you are asked to find the unique filter with the same second column as $P_{\boldsymbol{\phi}_1 \leftarrow \mathcal{C}_1}$, i.e. the vector from a..

c. Write code which uses Equation (5.68) to find H_0, H_1 from G_0, G_1, and state these filters with our compact filter notation. Also, state the forms

`h0,h1`, which should be used in calls to `DWTImpl` for our new wavelet. These replace the forms from equations (5.70)-(5.71) in Example 5.50, which we found for the first piecewise linear wavelet.
Hint: Note that the filter G_0 is unchanged from that of the first piecewise linear wavelet (since ϕ is unchanged when compared to the other wavelets for piecewise linear functions).

d. The filters you have found above should be symmetric, so that we can follow the procedure from Example 5.50 to listen to sound which has been wavelet-transformed by this wavelet. Write a program which plays our audio test file as in Example 5.50 for $m = 1, 2, 3, 4$ (i.e. plays the part in V_0), as well as the difference as in Exercise 9 (i.e. play the part from $W_0 \oplus W_1 \oplus \cdots \oplus W_{m-1}$), where the new filters you have found are used. Listen to the sounds.

Ex. 11 — Repeat the previous exercise for the Haar wavelet as in exercise 3, and plot the corresponding frequency responses for $k = 2, 4, 6$.

5.7 Summary

We started this chapter by motivating the theory of wavelets as a different function approximation scheme, which solved some of the shortcomings of Fourier series. While one approximates functions with trigonometric functions in Fourier theory, with wavelets one instead approximates a function in several stages, where one at each stage attempts to capture information at a given resolution, using a function prototype. We first considered the Haar wavelet, which is a function approximation scheme based on piecewise constant functions. We then moved on to a scheme with piecewise linear functions, where we saw that we had several degrees of freedom in constructing wavelets. Just as the DFT and the DCT, we interpreted a wavelet transformation as a change of basis, and found that the corresponding change of coordinate matrices had a particular form, which we studied. We denoted the change of basis in a wavelet transformation by the Discrete Wavelet Transform (DWT), and we showed how we could interpret and implement the DWT in terms of filters in such a way that a wide range of usable wavelets could be used as input to this implementation. We will use this implementation in the coming sections, in order to analyze images.