Wavelets, their friends, and what they can do for you

María Cristina Pereyra[*] Martin J. Mohlenkamp[†]

June 9, 2004

Preface

These notes were created by María Cristina Pereyra for the short course *Wavelets: Theory and Applications*, at the I Panamerican Advanced Studies Institute in Computational Science and Engineering (PASI), Universidad Nacional de Córdoba, Córdoba, Argentina, June 24–July 5, 2002. They were modified and extended by Martin J. Mohlenkamp for the short course *Wavelets and Partial Differential Equations*, at the II Panamerican Advanced Studies Institute in Computational Science and Engineering, Universidad Nacional Autónoma de Honduras, Tegucigalpa, Honduras, June 14–18, 2004. These courses are similar but not identical, so the notes do not exactly represent either course. We would like to thank the organizers for the invitations to participate in these institutes. We hope that these notes will help you to begin your adventures with wavelets.

Contents

[*]Department of Mathematics and Statistics, MSC03 2150, 1 University of New Mexico, Albuquerque NM 87131-0001 USA; http://www.math.unm.edu/∼crisp, crisp@math.unm.edu.
[†]Department of Mathematics, Ohio University, 321 Morton Hall, Athens OH 45701 USA; http://www.math.ohiou.edu/∼mjm, mjm@math.ohiou.edu.

1 Introduction

There are myriads of processes that need to be analyzed. From the begining of time we have invented alphabets, music scores, and languages to express our thoughts. In today's computer age being able to code, extract and transmit efficiently information is crucial. This will be the main theme of these lectures.

We present an introduction to the theory of wavelets with an emphasis in applications. The theory of wavelets is the latest commer to the world of signal processing (more than 20 years now). It grew and brewed in different areas of science. Harmonic analysts had developed powerful time/frequency tools, electrical engineers were busy with subband codding, quantum physicists were trying to understand coherent states. They were not aware of each others progress until the late 80's when a synthesis of all these ideas came to be, and what is now called wavelet theory contains all those ideas and much more. Wavelets is not one magical transform that solves all problems. It is a library of bases that is appropriate for a large number of situations where the traditional tools of for example, Fourier analysis, are not so good. There are many other problems which can not be optimally treated with either of the known tools, therefore new ones have to be designed.

Wavelets have gone from what some thought would be a short-lived fashion, to become part of the standard curriculum in electrical engineering, statistics, physics, mathematical analysis and applied mathematics. It has become part of the toolbox of statisticians, signal and image processors, medical imaging, geophysicists, speech recognition, video coding, internet communications, etc. Among the most spectacular and well-known applications are the wavelet based FBI standard for storing, searching and retrieving fingerprints, and the wavelet based JPEG-2000 standard, for image compression and transmission used widely for example in the internet.

2 Time-frequency analysis

Most things that we perceive directly are represented by functions in space or time (signals, images, velocity of a fluid). In the physical sciences, however, the frequency content of these processes has been shown to be of extreme importance for understanding the underlying laws. The ability to go from the time domain to the frequency domain and viceversa, and to have a dictionary that allows us to infer properties in one domain from information in the other domain, has made Fourier analysis an invaluable tool since its inception in the early nineteenth century. With the advent of computers and the rediscovery of the fast Fourier transform (FFT) in the early 1960's, the Fourier transform became ubiquitous in the field of signal and image processing. However this type of analysis has it limits, and certain problems are best handled using more delicate time-frequency analysis of the functions.

In this section we sketch the basics of Fourier analysis, its cousin the windowed Fourier (Gabor) analysis, local trigonometric expansions, and finally the multiresolution analysis achieved using wavelets. We strive to make the advantages and disadvantages of each of these approaches clear.

2.1 Fourier analysis

The key idea of harmonic analysis is to represent signals as a superposition of simpler functions that are well understood. Traditional Fourier series represent periodic functions as a sum of pure harmonics (sines and cosines),

$$f(t) \sim \sum_{n\in\mathbb{Z}} a_n e^{2\pi int}.$$

Fourier's statement in the early 1800's that *any periodic function* could be expanded in such a series, where the coefficients (amplitudes) for each frequency n are calculated by

$$a_n = \int_0^1 f(x)e^{-2\pi inx}dx, \tag{1}$$

revolutionized mathematics. It took almost 150 years to settle exactly what this meant. Only in 1966, Lenhart Carleson in a remarkable paper [**?**], showed that for square integrable functions on $[0,1]$ the Fourier partial sums converge pointwise a.e., and as a consequence the same holds for continuous functions (this was unknown until then).

Thus, a suitable periodic function is captured completely by its Fourier coefficients. The conversion from $f(t)$ to its Fourier coefficients $\hat{f}(n) = a_n$ is called the **Analysis phase**. Reconstructing the function $f(t)$ from the Fourier coefficients is called the **Synthesis phase**, and is accomplished by the inverse Fourier transform $f(t) = \sum_{n\in\mathbb{Z}} \hat{f}(n)e^{2\pi int}$. One useful property is the Plancherel identity $\|f\|_2^2 = \int_0^1 |f(x)|^2dx = \sum_n |\hat{f}(n)|^2$, which means that the analysis phase is a unitary transformation and thus is **Energy preserving**. In summary: *The exponential functions* $\{e^{2\pi int}\}_{n\in\mathbb{Z}}$ *form an orthonormal basis in* $L^2([0,1])$.

Notice that when computing the Fourier coefficients in (1), the sinusoidal waves exist at all times, hence when integrating against the function f, all values of the function are taken into account for each frequency. This means that a local change in the signal will affect *all Fourier coefficients*. We can interpret the exponentials as being well-localized in frequency, but global in space. The corresponding phase-plane decomposition is shown in Figure 1.

In the non-periodic setting one has a Fourier transform

$$\hat{f}(\xi) = \int_{\mathbb{R}} f(t)e^{-i\xi t}dt. \tag{2}$$

For square integrable functions the inverse transform is now an integral instead of a sum,

$$f(t) = \frac{1}{2\pi}\int_{\mathbb{R}} \hat{f}(\xi)e^{i\xi t}d\xi.$$

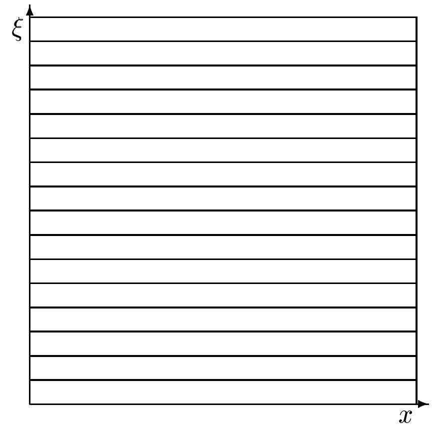

Figure 1: The Fourier phase plane.

The Fourier transform interacts very nicely with a number of operations. In particular differentiation is transformed into polynomial multiplication and viceversa, which “explains” the immense success of the Fourier transform techniques in the study of differential equations (certainly in the linear case). Also, convolutions are transformed into products, which explains the success in signal processing, since filtering (convolution) is one of the most important signal processing tools. A *time/frequency dictionary* is given in Table 1.

Time/Space		Frequency	
derivative	$f'(t)$	polynomial	$(f')^\wedge(\xi) = i\xi\hat{f}(\xi)$
convolution	$f * g(t) = \int f(t-x)g(x)dx$	product	$(f*g)^\wedge(\xi) = \hat{f}(\xi)\hat{g}(\xi)$
translation/shift	$\tau_s f(t) = f(t-s)$	modulation	$(\tau_s f)^\wedge(\xi) = e^{-is\xi}\hat{f}(\xi)$
rescaling/dilation	$f_s(t) = 1/s f(t/s)$	rescaling	$(f_s)^\wedge(\xi) = \hat{f}(s\xi)$
conjugate flip	$\tilde{f}(t) = \overline{f(-t)}$	conjugate	$(\tilde{f})^\wedge(\xi) = \overline{\hat{f}(\xi)}$

Table 1: A time-frequency dictionary

A consequence of the property $(f')^\wedge(\xi) = i\xi\hat{f}(\xi)$ is that *smoothness of the function implies fast decay of the Fourier transform at infinity.* One can show this by an integration by parts argument, or by noting that if $f'(x) \in L^2(\mathbb{R})$ then so is $i\xi\hat{f}(\xi)$, which means that $\hat{f}(\xi)$ must decay fast enough to compensate for the growth in the factor $i\xi$. Band-limited functions (compactly supported Fourier transform) have the most dramatic decay at infinity, hence they correspond to infinitely differentiable functions (C^∞). The unitary nature of the Fourier transform allows us to decide whether a function is in $L^2([0,1])$ just from knowledge about the size of the Fourier coefficients. However, this is not true about other size measurements, for example the $L^p([0,1])$ norms ($\|f\|_p = (\int_0^1 |f(t)|^p dt)^{1/p}$). It turns out that the phase of the coefficients is also needed to decide if a function belongs to L^p, $p \neq 2$. The difference between $p = 2$ and $p \neq 2$ is that the exponentials form an *unconditional basis* in the first case, but not in the other cases. Wavelet bases constitute unconditional bases for a large class of functional spaces: $L^p(\mathbb{R})$ for $1 < p < \infty$, Sobolev, Lipschitz or Hölder, Besov, Triebel-Lizorkin, bounded variation.

There is a parallel theory which is truly discrete, meaning it deals with finite vectors in $\mathbb{R}^N$ or $\mathbb{C}^N$. The Fourier transform in this context is nothing more than a particular change of basis, expressed by an unitary matrix. In this context all we encounter is linear algebra. This is of course what computers deal with, finite data. More precisely, an N-dimensional vector is given by $(z(0),\ldots,z(N-1))$ and its Fourier transform is defined by:

$$\hat{z}(m) = \frac{1}{\sqrt{N}}\sum_{n=0}^{N-1} z(n)e^{-2\pi inm/N}. \quad (3)$$

This is a symmetric orthogonal linear transformation, whose matrix F_N has entries $f_{n,m} = e^{-2\pi inm/N}/\sqrt{N}$. The columns are orthonormal vectors. The inverse transform is given by $(F_N)^{-1} = \overline{F_N}$. In the language of $\boxed{\text{bases}}$, the vectors $f_m(n) = f_{n,m}$ $m, n = 0, 1, \ldots, N-1$ form an orthonormal basis of N-dimensional space.

The exponentials are eigenfunctions of time/shift-invariant linear transformations. In the discrete case these transformations are circular convolutions, their matrix representation in the standard basis are circulant matrices, and they can be viewed also as polynomials in the shift operator $Sz(n) = z(n-1)$ (where the signals have been extended periodically to $\mathbb{Z}$). All such time-invariant linear transformations T (i.e. $TS = ST$) are diagonalizable in the Fourier basis. This means that the Fourier basis manages to completely decorrelate the coefficients of such transformations, hence providing the best matrix representation for them. As long as we are only interested in time-invariant linear phenomena, the Fourier transform provides simple answers to most questions (stationary processes, etc). But as soon as we move into *transient* phenomena (a note played at a particular time, an apple in the left corner of an image), the tool starts showing its limitations.

2.1.1 Algorithm: The Fast Fourier Transform

A key to the practical success of the discrete Fourier Transform (3) is a fast algorithm to accomplish it. The $N \times N$ matrix F_N is full, with entries $f_{n,m} = e^{-2\pi inm/N}/\sqrt{N}$. Applying it therefore involves N^2 complex multiplications. In the 1960's Cooley and Tukey [?] rediscovered an algorithm, which Gauss had found in 1805, that reduces the number of multiplications to order $N\log_2 N$. This improvement revolutionized digital signal processing (and of course at Gauss's time it was completely unforseen!). This is the famous *Fast Fourier Transform* (FFT). For a lively account of the success and far reaching consequences of the FFT see the article by Cipra [web:3]. There are faster algorithms, see for example an adaptive version, called Fastest Fourier Transform in the West (FFTW) [web:8] described in [?].

Mathematically, the FFT is based on a factorization of the Fourier matrix into a collection of sparse matrices. The process is easiest to explain, and implement, when $N = 2^n$. It begins with a factorization that reduces F_N to two copies of $F_{N/2}$. This reduction continuous for $\log_2 N = n$ steps, until the original matrix is written as a product of $2\log_2 N$ sparse matrices, half of which can be collapsed to a single permutation matrix. The total number of multiplications required to apply F_N becomes the $N\log_2 N$ instead of the brute force N^2.

For certain applications (e.g. in geophysics), it is required to calculate unequally spaced FFTs. These are transformations of the form

$$\sum_{n=0}^{N-1} z(n)e^{-2\pi i\lambda_n m/N},$$

where λ_n is the location of point number n. An efficient algorithm was developed by Dutt and Rokhlin [?]; see some applications in Beylkin's webpage [web:2].

2.2 Windowed Fourier transform

The continuous transform (2) provides a tool for analyzing a function, but the exponentials are not an orthonomal basis in $L^2(\mathbb{R})$. A simple way to get an orthonormal basis is to split the line into unit segments indexed by k, $[k, k+1]$, and on each segment use the periodic Fourier basis. That will give us the *windowed Fourier transform*.

The functions $g_{n,k}(t) = e^{2\pi int}\chi_{[k,k+1]}(t)$, $n, k \in \mathbb{Z}$, form an orthonormal basis for $L^2(\mathbb{R})$. However, the sharp windows effectively introduce discontinuities into the function, thereby ruining the decay in the Fourier coefficients. If you try to reconstruct with only the low frequency components, this will produce artifacts at the edges. For these reasons smoother windows are desirable. The sharp windows $\chi_{[0,1]}$ and its translates, are replaced by a smooth window g and their integer translates

$$g_{n,k}(t) = g(t-k)e^{2\pi int}. \tag{4}$$

Gabor considered in 1946 systems of this type and proposed to utilize them in communication theory [**?**]. Notice that their Fourier transforms can be calculated using the time/frequency dictionary:

formula below has changed!

$$(g_{n,k})^\wedge(\xi) = \hat{g}(\xi-n)e^{-2\pi ik(\xi-n)} = \hat{g}(\xi-n)e^{-2\pi ik(\xi)}. \tag{5}$$

This paragraph is new!

It is a fact that if $\{\psi_n\}$ is an orthonormal basis in $L^2(\mathbb{R})$, then so is the set of their Fourier transforms $\{(\psi_n)^\wedge\}$. Formula (5) tells us that if g generates a *Gabor basis* (that is, $\{g_{n,k}\}_{n,k\in\mathbb{Z}}$ defined by (4), is an o.n. basis), then so will $(g)^\wedge$. In particular, if $g(t) = \chi_{[0,1]}(t)$, then

$$(g)^\wedge(t)(\chi_{[0,1]})^\wedge(t) = e^{-it/2}\frac{\sin(t/2)}{(t/2)}$$

generates a Gabor basis. This provides an example of a smooth window which does not have compact support, but decays at infinity like $1/|t|$.

It will be nice to have pictures of these two windows with: the one that gives FWT and the one that has a sinc fucntion as window... like Figure 2 for local cosines

We would like the window to be well-localized in space/time. We can measure how spread out a function is by computing its center

$$\bar{t}(f) = \int t|f(t)|^2dt$$

and then its spread

$$\sigma_t(f) = \left(\int (t-\bar{t})^2|f(t)|^2dt\right).$$

Similarly, we can compute its frequency center

$$\bar{\xi}(f) = \int \xi|\hat{f}(\xi)|^2d\xi$$

and then its frequency spread

$$\sigma_\xi(f) = \left(\int (\xi-\bar{\xi})^2|\hat{f}(\xi)|^2d\xi\right).$$

The *Heisenberg box* of f is the box in the (t,ξ)-phase plane with center at $(\bar{t}(f),\bar{\xi}(f))$ and size $\sigma_t(f)$ by $\sigma_\xi(f)$. The uncertainty principle, whose proof can be found in many textbooks, tells us how much we can localize in both time and frequency.

Theorem 2.1 (Heisenberg's Uncertainty Principle) *Given f with $\|f\|_2 = 1$, we have*

"y" becomes t in formula below

$$\sigma_t^2\sigma_\xi^2 \geq \frac{1}{16\pi^2}.$$

It is thus impossible to find a function which is simultaneously perfectly localized in time and frequency. That is one cannot construct a function which is simultaneously compactly supported in time and band-limited (compactly supported in frequency).

The uncertainty principle tells us the best that we can ever do at localization. For the windowed Fourier transform, we also require our function $g \in L^2(\mathbb{R})$ to generate an orthonormal basis by the procedure in (4). It turns out that this requirement forces g to be poorly localized in either time or frequency.

Theorem 2.2 (Balian-Low) $\boxed{\textit{Suppose } g \in L^2(\mathbb{R}) \textit{ generates a Gabor basis}}$ *then either*

$$\int t^2 |g(t)|^2 dt = \infty \quad or \quad \int \xi^2 |\hat{g}(\xi)|^2 d\xi = \int |g'(t)|^2 dt = \infty.$$

A proof can be found in [HW, p.7].

Example 2.3 $\boxed{g(t) = \chi_{[0,1]}(t) \text{ generates a Gabor basis.}}$ The first integral is finite, but the second is not. $\boxed{\text{removed formula (it is in the new paragraph above) and adjusted phrasing}}$

Example 2.4 $\boxed{g(t) = e^{-1t/2} \frac{\sin(t/2)}{(t/2)} \text{ generates a Gabor basis.}}$ This time the second integral is finite but not the first. The Fourier transform of $\boxed{g \text{ is}}$ the characteristic function of the interval $[0, 1]$.

The first example is perfectly localized in time but not in frequency, and the second example is the opposite. In particular the slow decay of the Fourier transform reflects the lack of smoothness. Thus the Balian-Low theorem tells us that a Gabor window cannot be simultaneously compactly supported and smooth.

There is a $\boxed{\textit{continuous Gabor transform}}$ as well. Let g be a real and symmetric window, normalized so that $\|g\|_2 = 1$, let

$$g_{\xi,u}(t) = g(t-u)e^{i\xi t}, \quad u, \xi \in \mathbb{R}.$$

The Gabor transform is then

$$Gf(\xi, u) = \int_{-\infty}^{\infty} f(t)g(t-u)e^{-i\xi t} dt.$$

The multiplication by the translated window localizes the Fourier integral in a neighborhood of u. The Gabor transform is essentially an isometry, hence invertible in $L^2(\mathbb{R})$, namely:

$$f(t) = \frac{1}{2\pi} \int_{\mathbb{R}} \int_{\mathbb{R}} Gf(\xi, u)g(t-u)e^{it\xi} d\xi du.$$

and $\|f\|_2^2 = \frac{1}{2\pi} \|Gf\|_{L^2(\mathbb{R}^2)}$.

2.3 Local Trigonometric Expansions

The disappointing consequences of Theorem 2.2, can be avoided, however, if the exponentials are replaced by appropriate sines and cosines. One can then obtain a Gabor-type basis with smooth bell functions $\boxed{\text{compactly supported!}}$, called the *local cosine basis.* They were first discovered by Malvar [**?**] (from Brazil), introduced independently by Coifman and Meyer [**?**], and are discussed further in [**?**, **?**].

A standard Local Cosine basis is constructed as follows. We begin with a sequence of points on the line (or interval, or circle) $\cdots a_i < a_{i+1} \cdots$. Let $I_i = [a_i, a_{i+1}]$. We construct a set of compatible bells, indexed by their interval, $\{b_i(x)\}$, with the properties that

- $b_i(x)b_{i-1}(x)$ is an even function about a_i,
- $b_i(x)b_{i'}(x) = 0$ if $\boxed{i' \neq i, i \pm 1}$, and
- $\sum_i b_i^2(x) = 1$.

We also take them to be real and non-negative. On each interval we have a set of cosines of the proper scaling and shift, denoted

$$\left\{ c_i^j(x) = \sqrt{\frac{2}{a_{i+1} - a_i}} \cos\left(\frac{(j + 1/2)\pi(x - a_i)}{a_{i+1} - a_i} \right) \right\}_{j=0}^{\infty}. \tag{6}$$

The set $\{b_i(x)c_i^j(x)\}$ forms an orthonormal basis for the line. An example of a $b_i(x)c_i^j(x)$ is given in Figure 2.

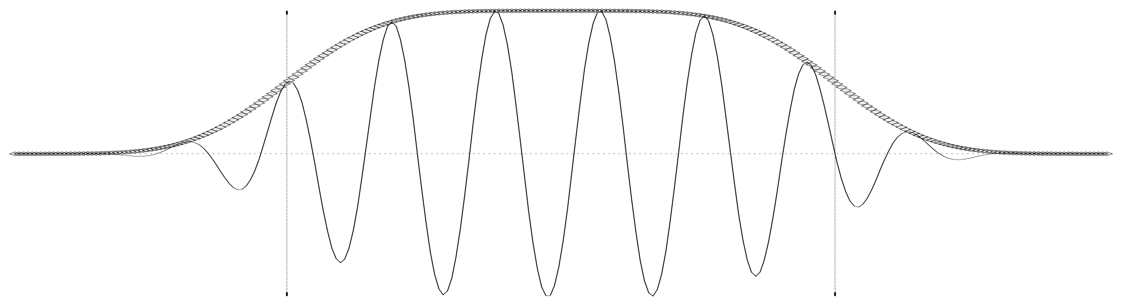

Figure 2: A local cosine basis function (light line) and its associated bell (dark line).

To show that they are indeed orthonormal, we check several cases. If $|i - i'| \geq 2$ the two elements are orthogonal because of disjoint support. If $i' = i + 1$, then

$$\begin{aligned} \langle b_i(x)c_i^j(x), b_{i+1}(x)c_{i+1}^{j'}(x)\rangle &= \int [b_i(x)b_{i+1}(x)]c_i^j(x)c_{i+1}^{j'}(x)dx \\ &= \int [\text{even}] \cdot (\text{even}) \cdot (\text{odd}) = \int (\text{odd about } a_{i+1}) = 0. \end{aligned} \tag{7}$$

If $i = i'$, $j \neq j'$ the properties of the bell allow us to reduce to the orthogonality of cosines of different frequencies i.e.,

$$\langle b_i c_i^j, b_i c_i^{j'}\rangle = \int b_i^2 c_i^j c_i^{j'}\, dx = \int_{a_i}^{a_{i+1}} c_i^j c_i^{j'}\, dx = 0. \tag{8}$$

If $i = i'$ and $j = j'$, then the integral evaluates to one. Completeness of the basis follows from the completeness of Fourier series.

The local cosine basis functions are localized in both time and frequency, within the limitations of the uncertainty principle. Intuitively, each is supported in a Heisenberg box, and together their boxes tile the phase plane. See Figure 3 for the general behavior of this intuition, and Figure 4 for an example of how it can be used. The local cosine basis can be very efficient at representing smooth functions with sustained high frequencies, such as the Associated Legendre functions that arise in spherical harmonics [?], one of which is shown in Figure 5.

The Local Cosine basis is based on trigonometric functions. With some simple preprocessing similar to that found in e.g. [?, Chapter 12] the local cosine expansion on each interval can be converted to a Fourier series expansion. The FFT from Section 2.1.1 can then be used to perform a fast Local Cosine transform.

2.4 Multiresolution Analysis and the Wavelet Transform

Although the local trigonometric basis is excellent in some situations, it is inappropriate for the many "natural" phenomena that have the property that high frequency "events" happen for short durations. Multiresolution methods that can handle such phenomena have a long history in numerical analysis, starting

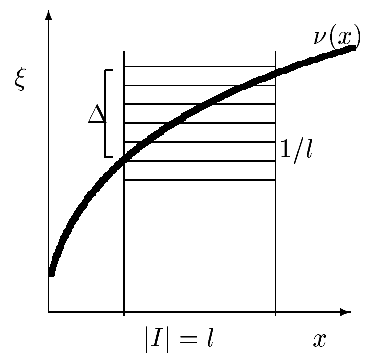

Figure 3: Phase Plane Intuition: If a function has 'instantaneous frequency' $\nu(x)$, then it should be represented by those Local Cosine basis elements whose rectangles intersect $\nu(x)$. There are $\max\{\Delta l, 1\}$ of these.

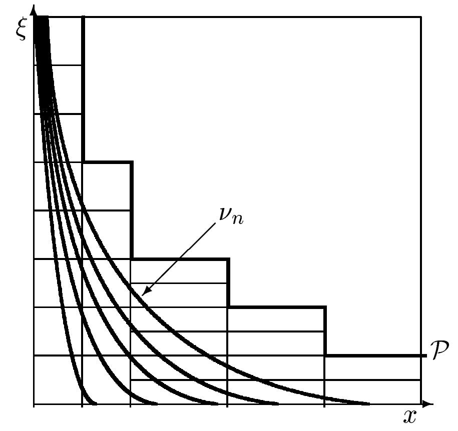

Figure 4: The local cosine boxes needed for the eigenfunctions of $-\Delta - C/x$, which have instantaneous frequency $\nu_n = \sqrt{C/x + \lambda_n}$.

with multigrid methods, and typically are fast iterative solvers based on hierarchical subdivisions. Wavelet methods are a multiresolution method, and have some other nice features. For a comparison of these methods and fast multipole methods see the invited address delivered by G. Beylkin at the 1998 International Mathematics Congress [**?**].

A $\boxed{\textit{(orthogonal) multiresolution analysis}}$ is a decomposition of $\mathbf{L}^2(\mathbb{R})$, into a chain of closed subspaces

$$\cdots \subset \mathbf{V}_{-2} \subset \mathbf{V}_{-1} \subset \mathbf{V}_0 \subset \mathbf{V}_1 \subset \mathbf{V}_2 \subset \cdots \subset \mathbf{L}^2(\mathbf{R})$$

such that

1. $\bigcap_{j\in\mathbf{Z}} \mathbf{V}_j = \{0\}$ and $\bigcup_{j\in\mathbf{Z}} \mathbf{V}_j$ is dense in $\mathbf{L}^2(\mathbf{R})$.
2. $f(x) \in \mathbf{V}_j$ if and only if $f(2x) \in \mathbf{V}_{j+1}$.
3. $f(x) \in \mathbf{V}_0$ if and only if $f(x-k) \in \mathbf{V}_0$ for any $k \in \mathbf{Z}$.
4. There exists a *scaling function* $\varphi \in \mathbf{V}_0$ such that $\{\varphi(x-k)\}_{k\in\mathbf{Z}}$ is an orthonormal basis of $\mathbf{V}_0$.

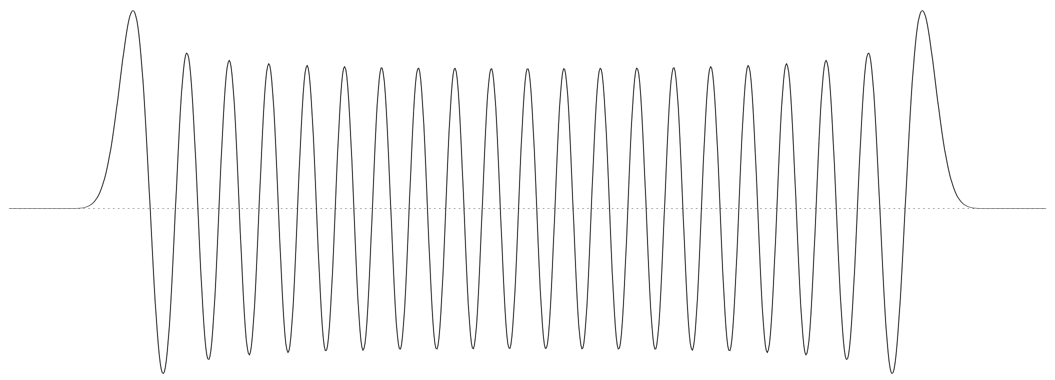

Figure 5: An Associated Legendre Function.

For notational convenience we define

$$\varphi_{j,k} = 2^{j/2}\varphi(2^j x - k), \tag{9}$$

and note that $\varphi_{j,k} \in \mathbf{V}_j$. We let P_j be the *orthogonal projection into* V_j, i.e.

$$P_j f(t) = \sum_{k\in\mathbb{Z}} \langle f, \varphi_{j,k}\rangle \varphi_{j,k}. \tag{10}$$

The function $P_j f(t)$ is an approximation to the original function at scale 2^{-j}. More precisely, it is the best approximation of f in the subspace $\mathbf{V}_j$.

How do we go from the approximation $P_j f$ to the better approximation $P_{j+1} f$? We simply add their difference. Letting

$$Q_j f = P_{j+1} f - P_j f,$$

we clearly have $P_{j+1} = P_j + Q_j$. This defines Q_j to be the $\boxed{\text{orthogonal}}$ projection onto a closed subspace, which we call $\mathbf{W}_j$. The space $\mathbf{W}_j$ is the orthogonal complement of $\mathbf{V}_j$ in $\mathbf{V}_{j+1}$, and $\mathbf{V}_{j+1}$ is the *direct sum* of $\mathbf{V}_j$ and $\mathbf{W}_j$:

$$\mathbf{V}_{j+1} = \mathbf{V}_j \oplus \mathbf{W}_j,$$

so that

$$\mathbf{L}^2(\mathbf{R}) = \bigoplus_{j\in\mathbf{Z}} \mathbf{W}_j. \tag{11}$$

One can show (Mallat [?]) that the scaling function φ, determines the *wavelet* ψ, such that $\{\psi(x-k)\}_{k\in\mathbf{Z}}$ is an orthonormal basis of $\mathbf{W}_0$. Since $\mathbf{W}_j$ is a dilation of $\mathbf{W}_0$, we can define

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

and have

$$\mathbf{W}_j = \operatorname{span}(\{\psi_{j,k}(x)\}_{k\in\mathbf{Z}}).$$

A calculation shows that

$$Q_j f = \sum_{k\in\mathbb{Z}} \langle f, \psi_{j,k}\rangle \psi_{j,k}.$$

The wavelet transform involves translations (like the Gabor basis) and scalings. This provides a zooming mechanism which is behind the multiresolution structure of these bases. The *orthogonal wavelet transform* is given by

$$Wf(j,k) = \langle f, \psi_{j,k}\rangle = \int_{\mathbb{R}} f(t)\overline{\psi_{j,k}(t)}dt,$$

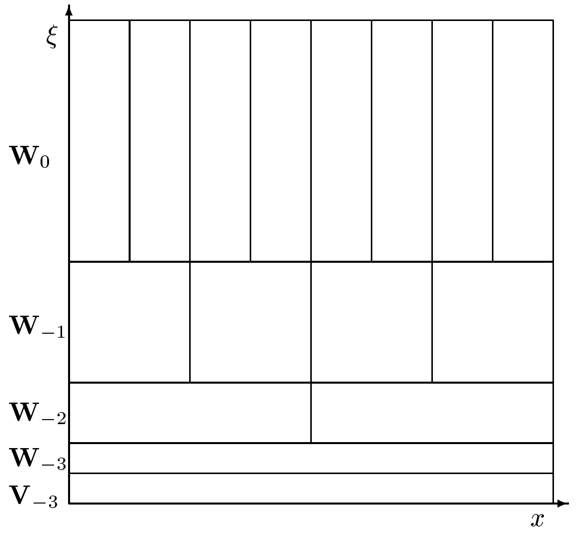

Figure 6: The wavelet phase plane.

and the reconstruction formula is,

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

Notice that on the Fourier side,

$$(\psi_{j,0})^\wedge(\xi) = 2^{-j/2}\hat{\psi}(2^{-j}\xi).$$

The Heisenberg boxes change with the scale parameter j, and remain constant along translations k. For $j = 0$ they are essentially squares of area 1, for other j the dimensions are $2^j \times 2^{-j}$. The wavelet transform divides the phase plane differently than either the Fourier or local cosine bases. The wavelet phase plane is given in Figure 6. For wavelets this decomposition is intuitive but not as rigorous as in the local cosine case. For the functions decomposed in the local cosine basis in Figure 4, wavelets give the decomposition in Figure 7. Note that whereas the local cosine basis had to adapt to account for the location of the singularity, the wavelet basis naturally matches it.

2.4.1 Haar Wavelets

Before considering how to construct scaling functions and wavelets, we will consider a simple example that predates the main development of wavelets. It was introduced by Haar in 1910 [**?**].

Let the scaling function be

$$\varphi(x) = \begin{cases} 1 & \text{for } 0 < x < 1 \\ 0 & \text{elsewhere} \end{cases}.$$

Then $\mathbf{V}_0 = \mathrm{span}(\{\varphi(x-k)\}_{k\in\mathbf{Z}})$ consists of piecewise constant functions with jumps only at integers. The wavelet is

$$\psi(x) = \begin{cases} 1 & \text{for } 0 < x < 1/2 \\ -1 & \text{for } 1/2 \le x < 1 \\ 0 & \text{elsewhere} \end{cases}.$$

The subspace $\mathbf{W}_0 = \mathrm{span}(\{\psi(x-k)\}_{k\in\mathbf{Z}})$ are piecewise constant functions with jumps only at half-integers, and average 0 between integers.

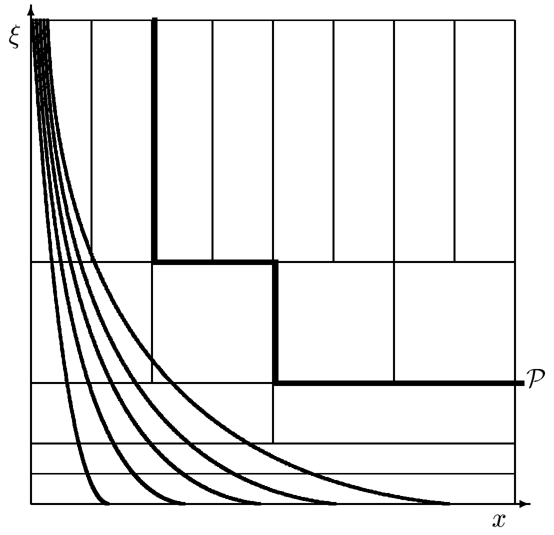

Figure 7: The wavelet boxes needed for the eigenfunctions of $-\Delta - C/x$, which have instantaneous frequency $\nu_n = \sqrt{C/x + \lambda_n}$.

We next consider an example of how to decompose a function into its projections onto the subspaces. In practice we select a coarsest scale $\mathbf{V}_{-n}$ and finest scale $\mathbf{V}_0$, truncate the chain to

$$\mathbf{V}_{-n} \subset \cdots \subset \mathbf{V}_{-2} \subset \mathbf{V}_{-1} \subset \mathbf{V}_0$$

and obtain

$$\mathbf{V}_0 = \mathbf{V}_{-n} \bigoplus_{j=1}^{n} \mathbf{W}_{-j}. \tag{12}$$

We will go through the decomposition process in the text using vectors, but it more enlightening to look at the graphical version in Figure 8.

We begin with a vector of $8 = 2^3$ "samples" of a function, which we assume to be the average value of the function on 8 intervals of length 1, so that our function is supported on the interval $[0, 8]$. We will use the vector

$$v_0 = [6, 6, 5, 3, 0, -2, 0, 6]$$

to represent our function in $\mathbf{V}_0$. To construct the projection onto $\mathbf{V}_{-1}$ we average pairs of values, to obtain

$$v_{-1} = [6, 6, 4, 4, -1, -1, 3, 3].$$

The difference is in $\mathbf{W}_{-1}$, so we have

$$w_{-1} = [0, 0, 1, -1, 1, -1, -3, 3].$$

By repeating this process, we obtain

$$\begin{aligned} v_{-2} &= [5, 5, 5, 5, 1, 1, 1, 1], \\ w_{-2} &= [1, 1, -1, -1, -2, -2, 2, 2], \\ v_{-3} &= [3, 3, 3, 3, 3, 3, 3, 3], \quad \text{and} \\ w_{-3} &= [2, 2, 2, 2, -2, -2, -2, -2]. \end{aligned}$$

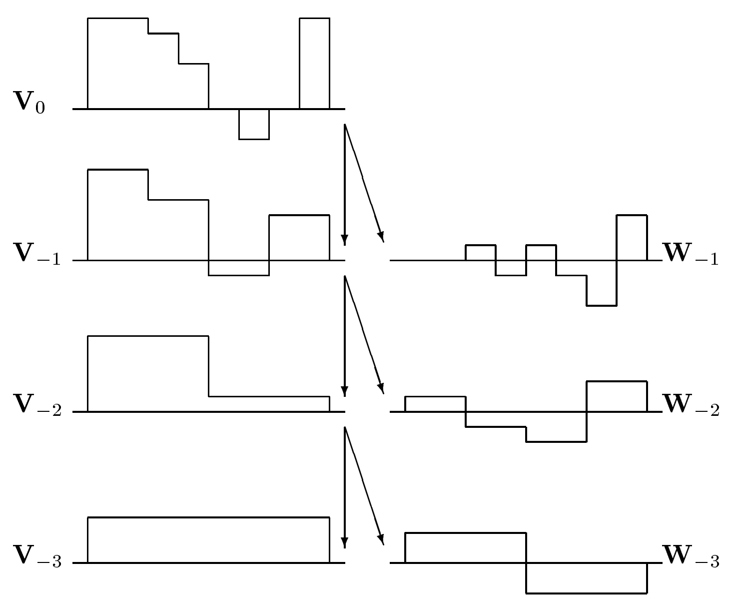

Figure 8: A wavelet decomposition. $\mathbf{V}_0 = \mathbf{V}_{-3} \oplus \mathbf{W}_{-3} \oplus \mathbf{W}_{-2} \oplus \mathbf{W}_{-1}$.

For the example in Figure 8, the scaling function subspaces are shown in Figure 9 and the wavelet subspaces are shown in Figure 10. To compute the coefficients of the expansion (10), we need to compute the inner product $\langle f, \varphi_{j,k} \rangle$ for the function (9). In terms of our vectors, we have, for example

$$\langle f, \varphi_{0,3} \rangle = \langle [6,6,5,3,0,-2,0,6], [0,0,0,1,0,0,0,0] \rangle = 3$$

and

$$\langle f, \varphi_{1,1} \rangle = \langle [6,6,5,3,0,-2,0,6], [0,0,1/\sqrt{2},1/\sqrt{2},0,0,0,0] \rangle = 8/\sqrt{2}$$

The scaling function φ satisfies the *two-scale recurrence equation*,

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

which tells us $\varphi_{j,k} = (\varphi_{j-1,2k} + \varphi_{j-1,2k+1})/\sqrt{2}$, and thus

$$\langle f, \varphi_{j,k} \rangle = \frac{1}{\sqrt{2}}(\langle f, \varphi_{j-1,2k} \rangle + \langle f, \varphi_{j-1,2k+1} \rangle).$$

Thus we can also compute

$$\langle f, \varphi_{1,1} \rangle = \frac{1}{\sqrt{2}}(5+3).$$

The coefficients $\langle f, \varphi_{j,k} \rangle$ for j fixed are called the *averages* of f at scale j, and commonly denoted $a_{j,k}$.

Similarly, the wavelet satisfies the *two-scale difference equation*,

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

and thus we can recursively compute

$$\langle f, \psi_{j,k} \rangle = \frac{1}{\sqrt{2}}(\langle f, \varphi_{j-1,2k} \rangle - \langle f, \varphi_{j-1,2k+1} \rangle).$$

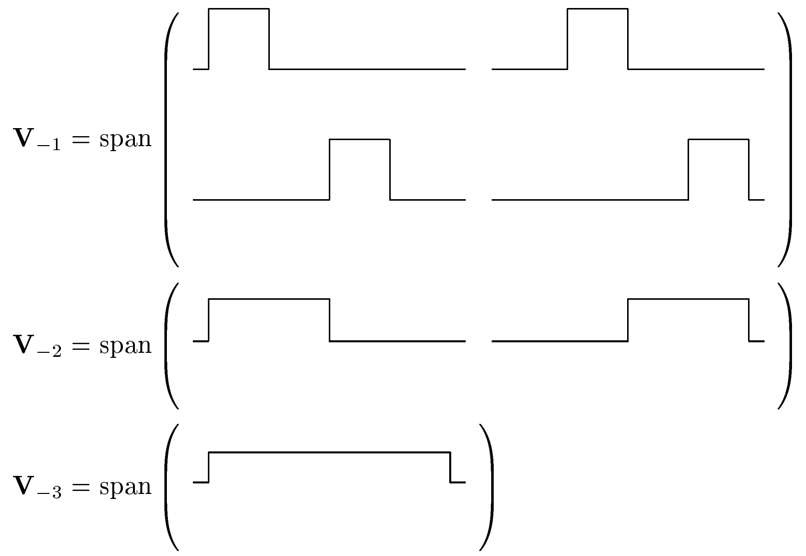

Figure 9: The scaling function subspaces used in Figure 8.

The coefficients $\langle f, \psi_{j,k} \rangle$ for j fixed are called the *differences* of f at scale j, and commonly denoted $d_{j,k}$. Evaluating the whole set of Haar coefficients $d_{j,k}$ and averages $a_{j,k}$ requires $2(N-1)$ additions and $2N$ multiplications. The discrete wavelet transform can be performed using a similar *cascade algorithm* with complexity $\mathcal{O}(N)$, where N is the number of data points.

We can view the averages at resolution j like successive approximations to the original signal, and the *details*, necessary to move from level j to the next level $(j+1)$, are encoded in the Haar coefficients at level j. Starting at a low resolution level, we can obtain better and better resolution by adding the details at the subsequent levels. As $j \to \infty$, the resolution is increased, that is the steps get smaller (length 2^{-j}), and the approximation converges to $f \in L^2(\mathbb{R})$ a.e. and in L^2-norm. Clearly the subspaces are nested, that is, $\mathbf{V}_j \subset \mathbf{V}_{j+1}$, and their intersection is the trivial subspace containing just the zero function. Therefore the Haar scaling function generates an orthogonal MRA.

2.4.2 Algorithm: The Fast Wavelet Transform

Given an orthogonal MRA with scaling function φ, then $\varphi \in \mathbf{V}_0 \subset \mathbf{V}_1$ and the functions $\varphi_{1,k}(t) = \sqrt{2}\varphi(2t-k)$ for $k \in \mathbb{Z}$ form an orthonormal basis of $\mathbf{V}_1$. This means that the following *scaling equation* holds:

$$\varphi(t) = \sum_{k \in \mathbb{Z}} h_k \varphi_{1,k}(t) = \sqrt{2} \sum_{k \in \mathbb{Z}} h_k \varphi(2t - k).$$

In general this sum is not finite, but whenever it is, the scaling function φ has *compact support*, and so will the wavelet. In the case of the Haar MRA, we have $h_0 = h_1 = 1/\sqrt{2}$, and all other coefficients vanish. The sequence $H = \{h_k\}$ is the so called *lowpass filter*. We will assume that the lowpass filter has length L (it turns out that such filters will always have even lengths $L = 2M$). The *refinement mask* is given by $H(z) = \frac{1}{\sqrt{2}} \sum h_k z^k$. Note the abuse in notation in that H is a sequence, but is also a function in $z = e^{-2\pi i \xi}$, or, abusing even further, it can be viewed as a periodic function of period one in the frequency variable ξ, whose Fourier coefficients are $\hat{H}(n) = h_n/\sqrt{2}$. We expect that it will be clear from the context what we mean by H in each case.

Notice that the wavelet ψ that we are seeking is an element of $\mathbf{W}_0 \subset \mathbf{V}_1$. Therefore it will also be a

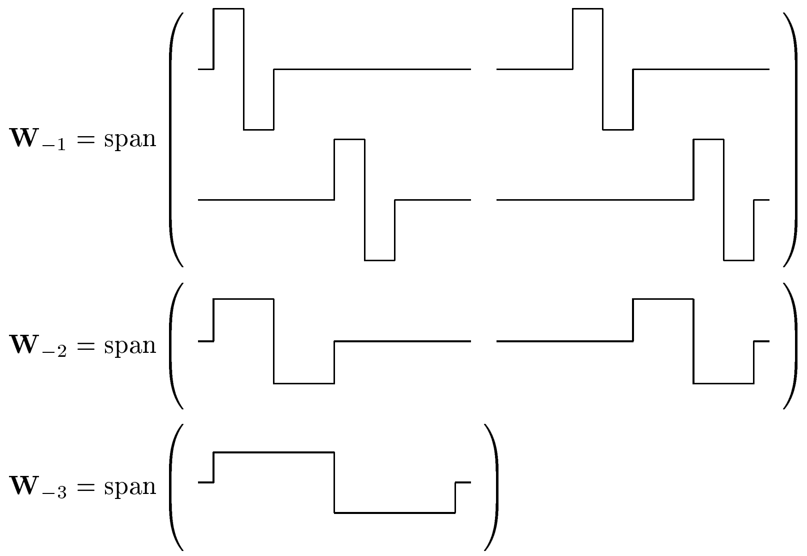

Figure 10: The wavelet subspaces used in Figure 8.

superposition of the basis $\{\varphi_{1,k}\}_{k\in\mathbb{Z}}$ of $\mathbf{V}_1$. Define the highpass filter $G = \{g_k\}$ by

$$g_k = (-1)^{k-1}\overline{h_{1-k}}.$$

Now define

$$\psi(t) = \sum_{k\in\mathbb{Z}} g_k \varphi_{1,k}(t).$$

In Section 3.1 we will discuss why this process yields the wavelet ψ.

A cascade algorithm, similar to the one described for the Haar basis, can be implemented to provide a fast wavelet transform. Given the coefficients $\{a_{J,k} = \langle f, \varphi_{J,k}\rangle\}_{k=0,1,\ldots,N-1}$, $N = 2^J$ "samples" of the function f defined on the interval $[0,1]$ and extended periodically on the line. If the function f is not periodic, then such periodization will create artifacts at the boundaries. In that case it is better to use wavelets adapted to the interval; see Section 4.5. The coefficients $a_{j,k} = \langle f, \varphi_{j,k}\rangle$ and $d_{j,k} = \langle f, \psi_{j,k}\rangle$ for scales $j < J$ can be calculated in order N operations, via

$$a_{j,k} = \sum_{n=0}^{N2^{-j}-1} \overline{h_n}\, a_{j+1,n+2k} = [\tilde{H} * a_{j+1}](2k),$$

$$d_{j,k} = \sum_{n=0}^{N2^{-j}-1} \overline{g_n}\, a_{j+1,n+2k} = [\tilde{G} * a_{j+1}](2k),$$

where $a_j = \{a_{j,k}\}$ and $d_j = \{d_{j,k}\}$ are viewed as periodic sequences with period $N2^{-j} > L$ (effectively in each sum there are only L non-zero terms). The filter $\tilde{H} = \{\tilde{h}_k = \overline{h_{-k}}\}$ is the conjugate flip of the filter H, and similarly for $\tilde{G}$. Thus we can obtain the approximation and detail coefficients at a rougher scale j by convolving the approximation coefficients at the finer scale $j+1$ with the low and highpass filters $\tilde{H}$ and $\tilde{G}$, and *downsampling* by a factor of 2. More precisely the *downsampling operator* takes an N-vector and maps it into a vector half as long by discarding the odd entries, $Ds(n) = s(2n)$; it is denoted by the symbol $\downarrow 2$. In electrical engineering terms this is the *analysis phase* of a *subband filtering scheme*, which can be represented schematically by:

$$a_{j+1} \rightarrow \begin{array}{l} \nearrow \\ \searrow \end{array} \begin{array}{l} \boxed{*\tilde{H}} \rightarrow \boxed{\downarrow 2} \rightarrow a_j \\ \\ \boxed{*\tilde{G}} \rightarrow \boxed{\downarrow 2} \rightarrow d_j \end{array} .$$

Computing the coefficients can be represented by the following tree or *pyramid scheme*,

$$\begin{array}{ccccccccc} a_J & \rightarrow & a_{J-1} & \rightarrow & a_{J-2} & \rightarrow & a_{J-3} & \cdots \\ & \searrow & & \searrow & & \searrow & & \\ & & d_{J-1} & & d_{J-2} & & d_{J-3} & \cdots \end{array} .$$

A graphical version of this algorithm is given in Figure 11. The reconstruction of the "samples" at level j

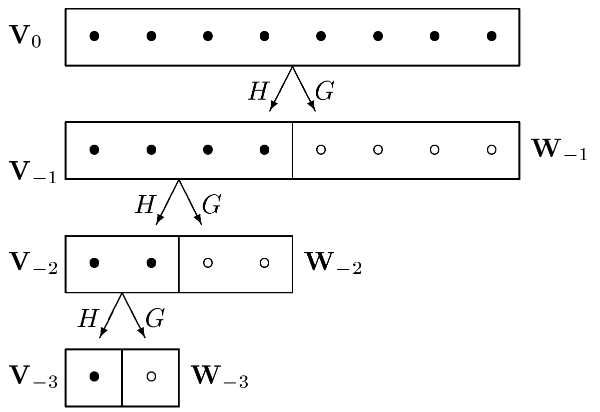

Figure 11: The cascade algorithm for the fast wavelet transform.

from the samples and details at the previous level, is also an order N algorithm,

$$a_{j+1,2k} = \sum_{n=1}^{L/2=M} h_{2k} a_{j,n-k} + \sum_{n=1}^{L/2=M} g_{2k} d_{j,n-k},$$

$$a_{j+1,2k-1} = \sum_{n=1}^{L/2=M} h_{2k-1} a_{j,n-k} + \sum_{n=1}^{L/2=M} g_{2k-1} d_{j,n-k}.$$

This can also be rewritten in terms of circular convolutions:

$$a_{j+1,m} = \sum_n \left[h_{m-2n} a_{j,n} + g_{m-2n} d_{j,n} \right] = H * Ua_j(m) + G * Ud_j(m),$$

where here we first upsample the approximation and detail coefficients to restore the right dimensions, then convolve with the filters and finally add the outcomes. More precisely the *upsampling operator* takes an N-vector and maps it to a vector twice as long, by intertwining zeros,

$$Us(n) = \begin{cases} s(n/2) & \text{if } n \text{ is even} \\ 0 & \text{if } n \text{ is odd} \end{cases} ;$$

it is denoted by the symbol $\uparrow 2$. This corresponds to the synthesis phase of the *subband filtering scheme*, which can be represented schematically by:

$$\begin{array}{lll} a_j \to \boxed{\uparrow 2} \to \boxed{*H} & & \\ & \searrow & \\ & \nearrow & \oplus \to a_{j+1}\,. \\ d_j \to \boxed{\uparrow 2} \to \boxed{*G} & & \end{array}$$

Notice that both up and downsampling have nice representations on Fourier domain,

$$(\downarrow 2z)^{\wedge}(\xi) = \frac{\hat{z}(\xi/2) + \hat{z}(\xi/2 + 1/2)}{2},$$

$$(\uparrow 2z)^{\wedge}(\xi) = \hat{z}(2\xi).$$

The reconstruction can also be illustrated with a pyramid scheme,

$$\begin{array}{ccccccc} a_n & \to & a_{n+1} & \to & a_{n+2} & \to & a_{n+3} \quad \cdots \\ & \nearrow & & \nearrow & & \nearrow & \\ d_n & & d_{n+1} & & d_{n+2} & & \cdots \end{array}.$$

Note that once the lowpass filter H is chosen, then everything else, the highpass filter G, the scaling function φ and the wavelet ψ, are completely determined, and so is the MRA. In practice one never computes the values of φ and ψ. All the manipulations are performed with the filters G and H, even if they involve calculating quantities associated to φ or ψ, like moments or derivatives.

The total cost of this cascade is $2 \cdot N \cdot L$, where L is the length of the filter. Generally $L \ll N$, but if it is not, then the FFT can be used to perform the convolutions.

2.4.3 The Continuous Wavelet transform

As in the Gabor case, there is a *continuous wavelet transform* which can be traced back to the famous *Calderón reproducing formula*. In this case we have continuous translation and scaling parameters, $s \in \mathbb{R}^+$ (that is $s > 0$), $u \in \mathbb{R}$. A family of *time-frequency atoms* is obtained by rescaling by s and shifting by u a wavelet $\psi \in L^2(\mathbb{R})$ with zero average ($\int \psi = 0$), and normalized ($\|\psi\|_2 = 1$),

$$\psi_{s,u}(t) = \frac{1}{\sqrt{s}}\psi\left(\frac{t-u}{s}\right).$$

The continuous wavelet transform is then defined by,

$$Wf(s,u) = \langle f, \psi_{s,u}\rangle = \int_{\mathbb{R}} f(t)\overline{\psi_{s,u}(t)}dt.$$

If ψ is real valued, then the wavelet transform measures the variation of f near u at scale s (in the orthonormal case, $u = k2^{-j}$ and $s = 2^{-j}$). As the scale s goes to zero (j goes to infinity), the decay of the wavelet coefficients characterizes the regularity of f near u (in the discrete case $k2^{-j}$).

Under very mild assumptions on the wavelet ψ we obtain a reconstruction formula. For any $f \in L^2(\mathbb{R})$,

$$f(t) = \frac{1}{C_\psi}\int_0^\infty \int_{-\infty}^{+\infty} Wf(s,u)\psi_{s,u}(t)\frac{duds}{s^2},$$

and $\|f\|_2 = \frac{1}{\sqrt{C_\psi}}\|Wf\|_{L^2(\mathbb{R}^+\times\mathbb{R})}$; provided that ψ satisfies *Calderón's admissibility condition* [?]

$$C_\psi = \int_0^\infty \frac{|\hat{\psi}(\xi)|^2}{\xi}d\xi < \infty.$$

One can heuristically discretize the above integrals over the natural time-frequency tiling of the upper-half plane given by the parameters j, k to obtain the discrete reconstruction formulas, and corresponding admissibility condition:

$$\sum_{j\in\mathbb{Z}} |\hat{\psi}(2^j\xi)|^2 = 1, \quad \text{for almost every } \xi.$$

It turns out that this condition is necessary and sufficient for an orthonormal system $\{\psi_{j,k}\}$ to be complete. Analogous admissibility conditions for general dilations has been recently shown to be necessary and sufficient by R. Laugesen [**?**].

3 Basic Wavelets

When we introduced wavelets through the MRA in Section 2.4, we deliberately avoided the issue of how to find a MRA and its associated wavelet. Currently, the preferred method is “look in the literature”. A vast number of different types of wavelets have now been constructed, and one should only consider constructing one's own if one has a very, very good reason.

In this section we sketch the theory an give examples for the “plain” wavelets, namely those that do come from an orthogonal MRA as described in Section 2.4. In Section 4 we will discuss other types of wavelets, and related objects.

3.1 Daubechies Style Wavelets

In this section we discuss the wavelets that grew mainly from the work of Daubechies [?]. These wavelets are compactly supported and can be designed with as much smoothness as desired. But as smoothness increases so does the length of the support, which is an example of the trades-off one often encounters in the design of wavelets. In fact, there are no infinitely differentiable and compactly supported orthogonal wavelets.

3.1.1 Filter banks

In Section 2.4.2 we introduced the lowpass filter $H = \{h_k\}_{k\in\mathbb{Z}}$ associated to an MRA with scaling function ϕ. Remember the scaling equation, which is just a reflection of the fact that $\phi \in \mathbf{V}_1$,

$$\phi(t) = \sum_{k\in\mathbb{Z}} h_k\phi_{1,k}(t), \quad H(z) = \frac{1}{\sqrt{2}}\sum_{k\in\mathbb{Z}} h_k z^k. \tag{13}$$

Similarly, the fact that the wavelet $\psi \in \mathbf{V}_1$, leads to a similar equation defining the highpass filter G,

$$\psi(t) = \sum_{k\in\mathbb{Z}} g_k\phi_{1,k}(t), \quad G(z) = \frac{1}{\sqrt{2}}\sum_{k\in\mathbb{Z}} g_k z^k. \tag{14}$$

¿From the knowledge of the lowpass filter H, we chose in Section 2.4.2, in a seemingly arbitrary fashion, the highpass filter $G = \{g_k\}_{k\in\mathbb{Z}}$,

$$g_k = (-1)^{k-1}\overline{h_{k-1}}, \tag{15}$$

and claimed it will indeed produce the desired wavelet ψ by equation (14). We will try to justify here why such choice is acceptable, and clarify some of the necessary conditions on the filters H and G so that there is a solution ϕ to the equation (13) that generates an orthogonal MRA.

The first observation is that if ϕ is given by the scaling equation (13), and $\{\phi(t-k)\}_{k\in\mathbb{Z}}$ is an orthonormal set, then necessarily the lowpass filter H must satisfy the following condition for all $z = e^{-2\pi i\xi}$,

$$H(z)\overline{H(z)} + H(-z)\overline{H(-z)} = \sqrt{2}. \tag{16}$$

This would be recognized in the engineering community as a *quadrature mirror filter* (QMF) condition, necessary for exact reconstruction for a pair of filters, and introduced in 1985 by Smith and Barnwell [?] for subband coding (however the first subband coding schemes without aliasing date back to Esteban and Galan [?]). There is a good discussion in Daubechies' classic book [?, pp. 156-163] about connections with subband filtering, as well as books dedicated to the subject such as the one authored by Vetterli and Kovacevic [?].

Since ψ satisfies (14) and its integer shifts are also supposed to be orthonormal, we conclude that the highpass filter G must also satisfy a QMF condition, namely, for all $z = e^{-2\pi i\xi}$,

$$G(z)\overline{G(z)} + G(-z)\overline{G(-z)} = \sqrt{2}. \tag{17}$$

In fact, if we define G via (15), and H is a QMF, then so will be G. The orthogonality between W_0 and V_0 imply that for all $z = e^{-2\pi i\xi}$

$$H(z)\overline{G(z)} + H(-z)\overline{G(-z)} = 0, \tag{18}$$

and this condition is automatically satisfied for our choice (15) of G.

The two QMF conditions (16), (17) plus condition (18) can be summarized in one phrase, the 2×2 matrices

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

are *unitary* for all $z = e^{-2\pi i\xi}$. (Remember that a matrix is unitary if its columns form an orthonormal set, which implies that its rows are also orthonormal.) Notice that at $z = 1$ ($\xi = 0$), $H(1) = 1$ implies $G(1) = 0$ (equivalently $\sum_k g_k = 0$), which in turn implies $\hat{\psi}(0) = \int \psi(t)dt = 0$. This is a standard cancellation property for wavelets. Also at $z = -1$, $H(-1) = 0$, and $|G(-1)| = 1$. This explains the denomination low and highpass filters.

These conditions on finite filters (FIR) are necessary and sufficient to guarantee the validity of the following *perfect reconstruction filter bank*, which we already discussed in Section 2.4.2,

$$\begin{array}{ccccc} & & \boxed{*\tilde{H}} \to \boxed{\downarrow 2} \to a_{j-1} \to \boxed{\uparrow 2} \to \boxed{*H} & & \\ & \nearrow & & \searrow & \\ a_j \to & & & & \oplus \to a_j. \\ & \searrow & & \nearrow & \\ & & \boxed{*\tilde{G}} \to \boxed{\downarrow 2} \to d_{j-1} \to \boxed{\uparrow 2} \to \boxed{*G} & & \end{array}$$

A general filter bank is any sequence of convolutions and other operations. The study of such banks is an entire subject in engineering called *multirate signal analysis*, or *subband coding*. The term *filter* is used to denote a convolution operator because such operator can cut out various frequencies if the corresponding Fourier multiplier vanishes (or is very small) at those frequencies. You can consult Strang and Nguyen's textbook [**?**], or Vetterli and Kovacevic's one [**?**].

The simplest 2×2 unitary matrices are the identity matrix and the matrix $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$. Choosing them will lead us to the Haar basis. It is not hard to characterize 2×2 unitary matrices, that will lead to a variety of orthonormal filters potentially associated to MRAs. Not all of them will actually work since these conditions on the filters are not sufficient to guarantee the existence of a solution of the scaling equation.

The existence of a solution of the scaling equation can be expressed in the language of fixed point theory or fractal interpolating functions. Given a lowpass filter H, we define a transformation $T\varphi(t) = \sqrt{2}\sum_k h_k \varphi(2t - k)$, and then try to determine if it has a fixed point ϕ. If yes, then the fixed point is a solution to the scaling equation, however we still have to make sure that such ϕ generates an orthogonal MRA.

On the Fourier side the scaling equation becomes

$$\hat{\varphi}(\xi) = H(\xi/2)\hat{\varphi}(\xi/2),$$

where $H(\xi/2) = H(z^{1/2})$, or in general, $H(s\xi) = H(z^s)$ ($z = e^{-2\pi i\xi}$). We can iterate this formula to obtain

$$\hat{\varphi}(\xi) = \left(\Pi_{j=0}^{N} H(\xi/2^j)\right)\hat{\varphi}(\xi/2^N).$$

Provided $\hat{\varphi}$ is continuous at $\xi = 0$, $\hat{\varphi}(0) \neq 0$, and the infinite product converges, then a solution to the scaling equation will exist. It turns out that to obtain orthonormality of the set $\{\varphi_{0,k}\}_{k\in\mathbb{Z}}$ we must have $|\hat{\varphi}(0)| = 1$, and one usually normalizes to $\hat{\varphi}(0) = \int \varphi(t)dt = 1$, which happens to be useful in numerical implementations of the wavelet transform.

By now the conditions on the filter H that will guarantee the existence of a solution φ to the scaling equation are well understood, see [**?**] for more details. For example, necessarily $H(\xi = 0) = 1$ ($z = 1$) or equivalently $\sum_{k=0}^{L-1} h_k = \sqrt{2}$. As already mentioned, the orthonormality of the integer shifts of the scaling function imply the QMF condition for H. It turns out that for finite filters H the conditions listed above are sufficient to guarantee the existence of a solution φ to the scaling equation. For infinite filters an extra decay assumption is necessary. However, it is not sufficient to guarantee the orthonormality of the integer shifts of φ. But, if for example $\inf_{|\xi|\leq\pi/2}|H(\xi)|>0$ is also true, then $\{\varphi_{0,k}\}$ is an orthonormal set in $L^2(\mathbb{R})$. See for example Frazier's book [?, Ch. 5] for more details, or more advanced textbooks.

Once ϕ has been found that generates an orthogonal MRA, then Mallat's Theorem [?] guarantees that ψ exists, furthermore it satisfies the equation on Fourier side

$$\hat{\psi}(\xi)=G(\xi/2)\hat{\varphi}(\xi/2),$$

where $G(z)=z\overline{H(-z)}$, is given by our choice (15).

Hopefully the connection to filter banks is now clear. Suffices to say here that compactly supported scaling functions imply finite impulse response (FIR) filters. And FIR filters that satisfy some relatively easy to verify conditions lead to compactly supported scaling functions and wavelets. The connections to filter bank theory and the possibility of implementing efficiently FIR filters opened the doors to wide spread use of wavelets in applications.

3.1.2 Competing attributes

Wavelet theory can sometimes be thought as the search of smoother wavelets. Haar is perfectly localized in time but not in frequency. The goal is to find functions which are "simultaneously" localized in time and frequency (within the limits imposed by the Heisenberg principle).

Most of the applications of wavelets exploit their ability to approximate functions as efficiently as possible, that is with as few coefficients as possible. For different applications one wishes the wavelets to have various properties. Some of them are competing against each other, so it is up to the user to decide which one is more important.

Wavelets are designed through properties of their "quadrature mirror filters" $\{H,G\}$.

Compact support: We have already stressed that compact support is important for numerical purposes (implementation of the FIR and fast wavelet transform in Section 2.4.2). Also in terms of detecting point singularities, it is clear that if the signal f has a singularity at t_0 then if t_0 is inside the support of $\psi_{j,n}$, the corresponding coefficient could be large. If ψ has support of length l, then at each scale j there will be l wavelets interacting with the singularity (that is their support contains t_0). The shorter the support the less wavelets interacting with the singularity.

We have already mentioned that compact support of the scaling function coincides with FIR, moreover if the lowpass filter is supported on $[N_1,N_2]$, so is φ, and it is not hard to see that ψ will have support of the same length (N_2-N_1) but centered at $1/2$.

Smoothness: The regularity of the wavelet has effect on the error introduced by thresholding or quantizing the wavelet coefficients. Suppose an error ϵ is added to the coefficient $\langle f,\psi_{j,k}\rangle$, then we will add an error of the form $\epsilon\psi_{j,k}$ to the reconstruction. Smooth errors are often less *visible* or *audible*. Often better quality images are obtained when the wavelets are smooth.

The smoother the wavelet, the longer the support.

The uniform Lipschitz regularity α of φ and ψ is related to the number of zeros of the refinement mask at $z=-1$. The more zeros the smoother.

There is no orthogonal wavelet that is C^∞ and has exponential decay. Therefore there is no hope of finding an orthogonal wavelet that is C^∞ and has compact support.

Hint for the user: Do not use more derivatives than your function typically has.

Vanishing moments: A function ψ has M vanishing moments if

$$\int_{-\infty}^{+\infty}\psi(x)x^m\,dx=0,\qquad m=0,\ldots,M-1, \tag{19}$$

This automatically implies that ψ is orthogonal to polynomials of degree $M-1$.

Wavelets are usually designed with vanishing moments, which makes them orthogonal to the low degree polynomials, and so tend to compress non-oscillatory functions. For example, we can expand an M-differentiable function f in a Taylor series

$$f(x) = f(0) + f'(0)x + \cdots + f^{(M-1)}(0)\frac{x^{M-1}}{(M-1)!} + f^{(M)}(\xi(x))\frac{x^M}{M!}$$

and conclude, if ψ has M vanishing moments, that

$$|\langle f, \psi\rangle| \leq \max_x \left| f^{(M)}(\xi(x))\frac{x^M}{M!}\right|.$$

On the filter side, the fact that a wavelet ψ has M vanishing moments is a consequence of the following identities being valid for the filter G,

$$\sum_k G(k)k^m = 0, \qquad m = 0, \ldots, M-1.$$

The number of vanishing moments of ψ is related to the number of vanishing derivatives of $\hat{\psi}(\xi)$ at $\xi = 0$ (by the time/frequency dictionary), which in turn is related to the number of vanishing derivatives of the refinement mask $H(\xi)$ at $\xi = 1/2$. These relations imply that if ψ has M vanishing moments, then the polynomials of degree $M-1$ are reproduced by the scaling functions, this is often referred as the *approximation order* of the MRA. Haar has $M = 1$, so only constant functions are reproduced by the scaling functions.

The constraints imposed on orthogonal wavelets imply that if ψ has M moments then its support is at least of length $2M-1$. Daubechies wavelets have minimum support length for a given number of vanishing moments. So there is a trade-off between length of the support and vanishing moments. If the function has few singularities and is smooth between singularities, then we might as well take advantage of the vanishing moments. If there are many singularities, we might prefer to use wavelets with shorter supports.

Both smoothness and vanishing moments are related to the zeros of the refinement mask at $z = -1$. It turns out that the more vanishing moments the more regular the wavelet. However in terms of the amplitude coefficients, it is the number of vanishing moments that affect their size at fine scales, not the regularity.

Hint for the user: Use one vanishing moment per digit desired (truncation level).

Symmetry: It is impossible to construct compactly supported symmetric orthogonal wavelets except for Haar. However symmetry is often useful for image and signal analysis. Some wavelets have been designed to be nearly symmetric (Daubechies symmlets, for example). It can be obtained at the expense of one of the other properties. If we give up orthogonality, then there are compactly supported, smooth and symmetric *biorthogonal* wavelets, which we will describe in Section 4.2. If we use *multiwavelets*, we can construct them to be orthogonal, smooth, compactly supported and symmetric, which we will describe in Section 4.3.

Cost equation: You can get

- higher M
- more derivatives
- closer to symmetric

- closer to interpolating (coiflets)

if you pay by increasing the filter length, which causes

- longer (overlapping) supports, and so worse localization.
- slower transforms

The cost is linear (in M etc.).

3.1.3 Examples and Pictures

The cascade algorithm can be used to produce very good approximations for both ψ and φ, and this is how pictures of the wavelets and the scaling functions are obtained. For the scaling function φ, it suffices to observe that $\langle \varphi, \varphi_{1,k} \rangle = h_k$ and $\langle \varphi, \psi_{j,k} \rangle = 0$ for all $j \geq 1$ (the first because of the scaling equation, the second because $\mathbf{V}_0 \subset \mathbf{V}_j \perp \mathbf{W}_j$ for all $j \geq 1$), that is what we need to initialize and iterate as many times as we wish (say n iterations) the synthesis phase of the filter bank,

$$H \to \boxed{\uparrow 2} \to \boxed{*H} \to \boxed{\uparrow 2} \to \boxed{*H} \to \cdots \to \boxed{\uparrow 2} \to \boxed{*H} \to \{\langle \varphi, \varphi_{n+1,k} \rangle\}.$$

The output after n iterations are the approximation coefficients at scale $j = n + 1$. After multiplying by a scaling factor one can make precise the statement that

$$\varphi(k2^{-j}) \sim 2^{-j/2} \langle \varphi, \varphi_{j,k} \rangle.$$

It works similarly for the wavelet ψ. Notice that this time $\langle \psi, \varphi_{1,k} \rangle = g_k$ and $\langle \psi, \psi_{j,k} \rangle = 0$ for all $j \geq 1$. The cascade algorithm now will produce the approximation coefficients at scale j after $n = j - 1$ iterations,

$$G \to \boxed{\uparrow 2} \to \boxed{*H} \to \boxed{\uparrow 2} \to \boxed{*H} \to \cdots \to \boxed{\uparrow 2} \to \boxed{*H} \to \{\langle \psi, \varphi_{j,k} \rangle\}.$$

In Section 2.4.1 we saw the example of the **Haar Wavelet**. The Haar wavelet is perfectly localized in time, poorly localized in frequency, discontinuous, and symmetric. It has the shortest possible support, but only one vanishing moment. It is not well adapted to approximating smooth functions. As a reminder, the scaling function and wavelet are shown in Figure 12. The characteristic function of the unit interval

Haar

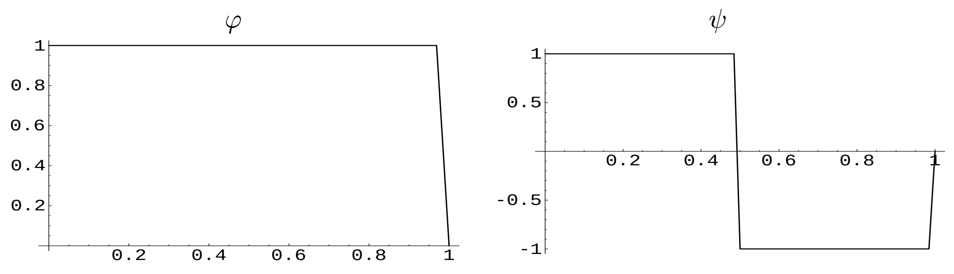

Figure 12: The Haar scaling function and wavelet.

generates an orthogonal MRA (Haar MRA). The non-zero lowpass filter coefficients are

$$H = [h_0, h_1] = \left[\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right].$$

Consequently the non-zero highpass coefficients are

$$G = [g_0, g_1] = \left[\frac{-1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right].$$

Therefore the Haar wavelet is $\psi(t) = \varphi(2t-1) - \varphi(2t)$. The refinement masks are $H(z) = (1+z^{-1})/2$ and $G(z) = (z-1)/2$.

The **Daubechies compactly supported wavelets** [?] have compact support of minimal length for any given number of vanishing moments. More precisely, if ψ has M vanishing moments, then the filters have length $\boxed{2M}$ (have $\boxed{2M}$ "taps"). For large M, the functions φ and ψ are uniformly Lipschitz α of the order $\alpha \sim 0.2M$. They are very asymmetric. When $M = 1$ we recover the Haar wavelet. For each integer $N \geq 1$ there is an orthogonal MRA with compactly supported wavelet minimally supported (length of the support $2N$), and the filters have $2N$-taps. They are denoted in Matlab by *dbN*. The *db*1 corresponds to the Haar wavelet. The coefficients corresponding to *db*2 are:

$$H = [h_0, h_1, h_2, h_3] = \left[\frac{1+\sqrt{3}}{4}, \frac{3+\sqrt{3}}{4}, \frac{3-\sqrt{3}}{4}, \frac{1-\sqrt{3}}{4}\right].$$

For $N > 2$ the values are not in closed form. The scaling function and wavelet for filter lengths 4, 8, and 12 are shown in Figure 13.

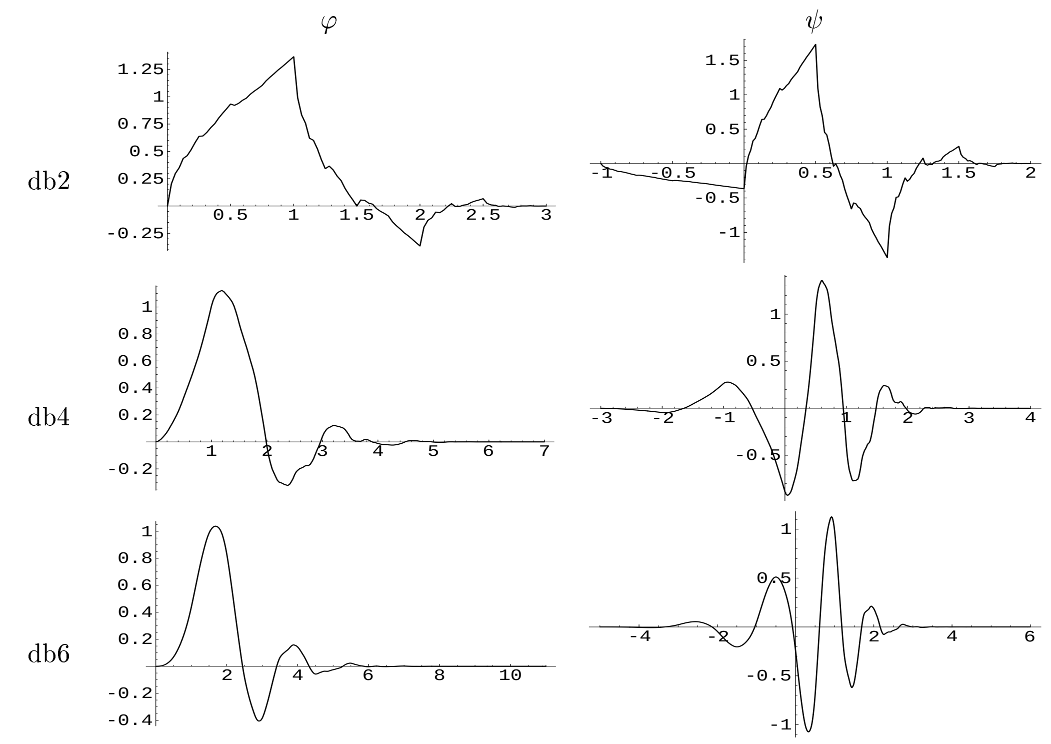

Figure 13: The Daubechies scaling function and wavelet for filters length 4, 8, and 12.

The **Coiflet** wavelet $\boxed{\text{[?, pp.258-261]}}$ ψ has M vanishing moments, $\boxed{\text{killed subsentence}}$ and φ has $M-1$ moments vanishing (from the second to the Mth moment, never the first since $\int \varphi = 1$. This extra property requires enlarging the support of ψ to length $(3M-1)$. This time if we approximate a regular function f by a Taylor polynomial, the approximation coefficients will satisfy,

$$2^{J/2}\langle f, \varphi_{J,k}\rangle \sim f(2^J k) + O(2^{-(k+1)J}).$$

Hence at fine scale J, the approximation coefficients are close to the signal samples. Rumor has it that the coiflets were constructed by Daubechies after Coifman requested them for the purpose of applications to almost diagonalization of singular integral operators [?]. The scaling function and wavelet for filter lengths 4, 8, and 12 are shown in Figure 14.

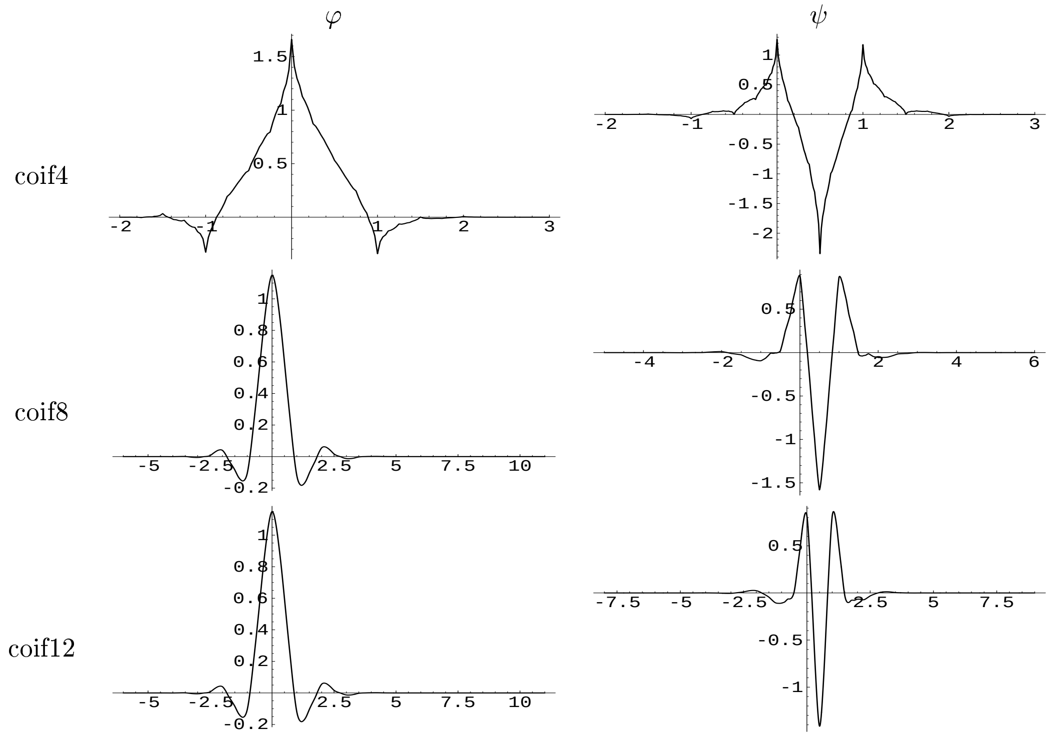

Figure 14: The coiflet scaling function and wavelet for filters length 4, 8, and 12.

Daubechies symmlets: M vanishing moments, minimum support of length $2M$, as symmetric as possible.

3.2 Other Plain Wavelets

Here we list other wavelets that are associated with an orthogonal MRA and indicate their properties.

Not all wavelets have compact support, for example the Meyer wavelet and the Mexican hat do not. However for applications it is a most desirable property since it corresponds to the FIR filter case.

Shannon wavelet: This wavelet does not have compact support, however it is C^{∞}. It is band-limited, but its Fourier transform is discontinuous, hence $\psi(t)$ decays like $1/|t|$ at infinity. $\hat{\psi}(\xi)$ is zero in a neighborhood of $\xi = 0$, hence all its derivatives are zero at $\xi = 0$, hence ψ has an infinite number of vanishing moments. **Shannon revisited:** This time $\hat{\varphi}(\xi) = \chi_{\{|\xi| \leq 1/2\}}(\xi)$ generates an orthogonal MRA. One can deduce that $H(\xi) = \chi_{\{|\xi| \leq 1/4\}}(\xi)$, hence $G(\xi) = e^{-2\pi i \xi} H(\xi)$, and $\hat{\psi}(\xi) = e^{-\pi i \xi} \chi_{\{1/4 \leq |\xi| \leq 1/2\}}(\xi)$. The family $\{\psi_{j,k}\}_{j,k \in \mathbb{Z}}$ is an orthonormal basis for $L^2(\mathbb{R})$. This basis is perfectly localized in frequency but not in time being the difference of two sinc functions, as opposed to the Haar basis which is perfectly localized in space but not in frequency.

The Shannon basis is connected to the celebrated *Shannon Sampling formula*. It states that band-limited functions can be recovered by appropriate samplings as coefficients of series of translated sinc functions. More precisely, **Shannon Sampling Formula:** *For $f \in L^1(\mathbb{R})$, with Fourier transform $\hat{f}$*

supported on the interval $[-1/2, 1/2]$*, then the following equality holds in the* L^2*-sense,*

$$f(x) = \sum_{n\in\mathbb{Z}} f(n)\mathrm{sinc}(x-n), \qquad \mathrm{sinc}(x) = \frac{\sin(\pi x)}{\pi x}.$$

The proof can be found in most of the textbooks; you can also check the original reference [?].

Meyer wavelet: This is a symmetric band-limited function whose Fourier transform is smooth, hence $\psi(t)$ has faster decay at infinity. The scaling function is also band-limited. Hence both ψ and φ are C^∞. The refinement mask is given by, $H(\xi) = \sqrt{2}\chi_{\{|\xi|\leq\pi/3\}}(\xi)$, hence it has vanishing derivatives of all orders at $\xi = 0$, therefore ψ has an infinite number of vanishing moments. (This wavelet was found by Strömberg in 1982 [?] J. O. Stromberg *A modified Franklin system and higher order spline systems on* $\mathbb{R}^n$ *as unconditional bases for Hardy spaces.* Conf. in honor of A. Zygmund, vol. II, W. Beckner et al. ed., Wadsworth math. series, pp. 475-493., and it went unnoticed for several years).

Battle-Lemarié spline wavelets: For splines of degree m, the refinement mask has m vanishing derivatives at $\xi = 1/2$. Hence ψ has $m+1$ vanishing moments. They do not have compact support, but they have exponential decay. Since they are polynomial splines of degree m, they are $m-1$ times continuously differentiable. For m odd, ψ is symmetric around 1/2. For m even it is antisymmetric around 1/2. The linear spline wavelet is the Franklin or Franklyn? wavelet.

4 Friends, Relatives, and Mutations of Wavelets

In this section we discuss various objects that are connected to wavelets, and try to organize them to give an overview. Three of these, the Fourier basis, the Gabor transform, and local trigonometric transforms, were already discussed in Section 2.

4.1 Relatives: Wavelets not from a MRA

The root meaning of "wavelet" is just "little wave", meaning a wave that does not last very long. Using this definition, there are wavelets that do not come from a Multiresolution Analysis. These are rare. If the wavelet has compact support then it does come from an MRA, see [?].

The following two wavelets do not have compact support and do not come from a MRA. They both have exponential decay and are appropriate for the continuous wavelet transform in Section 2.4.3. Neither are orthogonal to their shifts.

The **Mexican hat** has a closed formula involving second derivatives of the Gaussian:

$$\psi(t) = C(1-t^2)e^{t^2/2},$$

where the constant is chosen to normalize it in $L^2(\mathbb{R})$. According to Daubechies, this function is popular in "vision analysis".

The **Morlet wavelet** has a closed formula

$$\psi(t) = Ce^{-t^2/2}\cos(5t).$$

4.2 Mutation: Biorthogonal wavelets

Our first mutation is the loss of orthogonality. Orthogonality allows for straightforward calculation of the coefficients (via inner products with the basis elements). It guarantees that the energy is preserved. However sometimes it can be substituted by *biorthogonality* where there is an auxiliary set of functions that are used to compute the coefficients by taking inner products against them; also the energy is almost preserved.

Consider this simple example in two dimensions. The vectors $\vec{v}_1 = (1,0)$ and $\vec{v}_2 = (1,1)$ are linearly independent but not orthogonal. They form a basis in $\mathbb{R}^2$, that is given any $\vec{v} = (x,y) \in \mathbb{R}^2$ there exist unique coefficients a, b such that:

$$\vec{v} = a\vec{v}_1 + b\vec{v}_2.$$

How do we compute the coefficients? In the orthonormal case, we compute the inner product with the basis vectors. One can go through the simple algebra and write a, b in terms of x, y. It turns out that there are ***dual vectors*** such that $\langle \vec{v}_i, \vec{v}_j^* \rangle = \delta_{i,j}$, in particular this implies that $\vec{v}_1^*, \vec{v}_2^*$, such that $\vec{v}_1^* \perp \vec{v}_2$ and $\vec{v}_2^* \perp \vec{v}_1$. Inner product against these dual vectors will produce the corresponding coefficients

$$a = \langle \vec{v}, \vec{v}_1^* \rangle, \quad b = \langle \vec{v}, \vec{v}_2^* \rangle. \tag{20}$$

See Figure 15. Because $\vec{v}_2$ and $\vec{v}_1^*$ are orthogonal, then it should be clear that the orthogonal projection $\boxed{\mathrm{Proj}_{\vec{v}_1^*}(\vec{v})}$ of $\vec{v}$ onto $\vec{v}_1^*$ is equal to $\boxed{\mathrm{Proj}_{\vec{v}_1^*}(a\vec{v}_1)}$. We know how to compute orthogonal projections, $\boxed{\mathrm{Proj}_{\vec{v}_1^*}(\vec{v}) = \frac{\langle \vec{v}, \vec{v}_1^* \rangle}{\|\vec{v}_1^*\|} \vec{v}_1^*}$ and $\boxed{\mathrm{Proj}_{\vec{v}_1^*}(a\vec{v}_1) = \frac{\langle a\vec{v}_1, \vec{v}_1^* \rangle}{\|\vec{v}_1^*\|} \vec{v}_1^* = \frac{a}{\|\vec{v}_1^*\|} \vec{v}_1^*}$ (here we used the biorthogonality condition). These projections coincide, and similarly $\boxed{\mathrm{Proj}_{\vec{v}_2^*}(\vec{v}) = \mathrm{Proj}_{\vec{v}_2^*}(b\vec{v}_2)}$, we can deduce (20) from these identities. This is the simplest possible example of a ***biorthogonal basis*** .

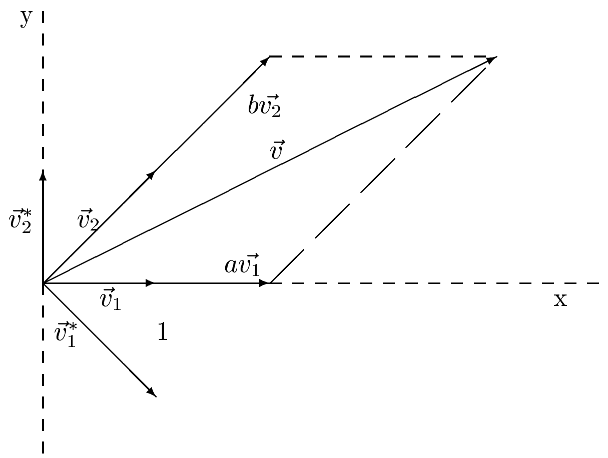

Figure 15: Biorthogonal vectors in $\mathbb{R}^2$.

Biorthogonal wavelets are constructed so that they will be a *Riesz basis* in $L^2(\mathbb{R})$. This means, first that any function in $L^2(\mathbb{R})$ has a unique representation as a superposition of basis elements, i.e. $f = \sum_n a_n \psi_n$. Second, it means that there is an appropriate substitute for the energy preserving property in the orthogonal case, namely, there exist positive constants c, C such that $\boxed{\text{formula below changed}}$

$$c \sum_n |a_n|^2 \leq \|f\|_2^2 \leq C \sum_k |a_n|^2.$$

$\boxed{\text{(a subsentence has been killed)}}$ The relative size of the constants c, C is important in applications. Their ratio corresponds to the condition number of a non-orthogonal change of basis, and the further it deviates from one, the worse the numerical performance. As a consequence of the Riesz Representation Theorem, there is always a dual basis $\{\psi_n^*\}$, in the sense that $\langle \psi_n, \psi_k^* \rangle = \delta_{n,k}$, and any square integrable function f can be expressed as

$$f = \sum_n \langle f, \psi_n^* \rangle \psi_n = \sum_n \langle f, \psi_n \rangle \psi_n^*.$$

A *biorthogonal multiresolution analysis* consists of two dual multiresolution analysis with scaling function φ, and φ^*, that is two decomposition of the space $L^2(\mathbb{R})$ into a nested sequence of closed subspaces are given

$$\cdots \subset \mathbf{V}_{-2} \subset \mathbf{V}_{-1} \subset \mathbf{V}_0 \subset \mathbf{V}_1 \subset \mathbf{V}_2 \subset \ldots,$$

$$\cdots \subset \mathbf{V}_{-2}^* \subset \mathbf{V}_{-1}^* \subset \mathbf{V}_0^* \subset \mathbf{V}_1^* \subset \mathbf{V}_2^* \subset \ldots,$$

such that their intersection are the trivial subspace, their union is dense in $L^2(\mathbb{R})$. The integer translates of φ form a Riesz basis for the subspace $\mathbf{V}_0$, the integer translates of φ^* form a Riesz basis for the subspace $\mathbf{V}_0^*$. They are dual bases in the sense that $\langle \phi_{0,m}, \phi_{0,k}^* \rangle = \delta_{k,m}$, and if $f \in \mathbf{V}_0$, then $f = \sum_k \langle f, \phi_{0,k}^* \rangle \phi_{0,k} = \sum_k \langle f, \phi_{0,k} \rangle \phi_{0,k}^*$. To move across scales the following scaling properties hold:

$$f(t) \in \mathbf{V}_j \Leftrightarrow f(2t) \in \mathbf{V}_{j+1},$$

$$f(t) \in \mathbf{V}_j^* \Leftrightarrow f(2t) \in \mathbf{V}_{j+1}^*.$$

Due to the biorthogonality, we can write $\mathbf{V}_{j+1}$ as a direct sum (not orthogonal, although we use the same symbol) of $\mathbf{V}_j$ and $\mathbf{W}_j$, $\mathbf{V}_{j+1} = \mathbf{V}_j \oplus \mathbf{W}_j$; similarly $\mathbf{V}_{j+1}^* = \mathbf{V}_j^* \oplus \mathbf{W}_j^*$. This time

$$\mathbf{V}_j \perp \mathbf{W}_j^*, \qquad \mathbf{V}_j^* \perp \mathbf{W}_j.$$

Furthermore, $L^2(\mathbb{R}) = \overline{\oplus_{j\in\mathbb{Z}} \mathbf{W}_j} = \overline{\oplus_{j\in\mathbb{Z}} \mathbf{W}_j^*}$.

Under certain conditions there will be dual wavelets ψ, ψ^* that generate a biorthogonal basis in $L^2(\mathbb{R})$. One can use either of the wavelets to find the coefficients and the other for reconstruction purposes, that is,

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

The wavelets will usually have different properties, e.g. one has short support while the other is smoother, or one has several vanishing moments while the other is smoother, etc. Depending on the application it would be preferable to use one for the synthesis of the coefficients, the other for the reconstruction.

The *basic scaling equations* read

$$\varphi(x) = \sum_{n\in\mathbb{Z}} h_n \varphi(2x-n), \quad \varphi^*(x) = \sum_{n\in\mathbb{Z}} h_n^* \varphi^*(2x-n),$$

where the filters $H = \{h_n\}$ and $H^* = \{h_n^*\}$ are zero except for a finite number (this will again imply compact support for the scaling functions). On Fourier side these equations become

$$(\varphi)^\wedge(\xi) = H(\xi/2)(\varphi)^\wedge(\xi/2), \quad (\varphi^*)^\wedge(\xi) = H^*(\xi/2)(\varphi^*)^\wedge(\xi/2),$$

where $H(\xi) = H(z)$, $H^*(\xi) = H^*(z)$ for $z = e^{-2\pi i\xi}$, are the *refinement masks* corresponding to the multi-scaling functions φ, φ^* respectively:

$$H(z) = \frac{1}{\sqrt{2}} \sum_{n\in\mathbb{Z}} h_n z^n, \quad H^*(z) = \frac{1}{\sqrt{2}} \sum_{n\in\mathbb{Z}} h_n^* z^n.$$

Iterating the equations we get

$$(\varphi)^\wedge(\xi) = H\big(\xi/2\big)H\big(\xi/4\big)(\varphi)^\wedge\big(\xi/4\big) = \cdots = \prod_{j=1}^{\infty} H\big(\xi/2^j\big)(\varphi)^\wedge(0). \tag{21}$$

Provided $(\varphi)^\wedge$ is continuous at $\xi = 0$. The same holds for $(\varphi^*)^\wedge$ and H^*.

In terms of the refinement masks, the biorthogonality condition at the level of the scaling functions becomes

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

We want to find a biorthogonal basis generated by wavelets so that for each j the collections $\{\psi_{j,k}\}_k$, $\{\psi^*_{j,k}\}_k$, form a biorthogonal basis for $\mathbf{W}_j$ and $\mathbf{W}^*_j$. Provided the filter H, H^* satisfy some mild conditions, the problem reduces to finding a pair of dual filters G, G^* such that the following biorthogonality conditions hold:

- $H(z)\overline{H^*(z)} + H(-z)\overline{H^*(-z)} = \sqrt{2}$
- $G(z)\overline{G^*(z)} + G(-z)\overline{G^*(-z)} = \sqrt{2}$
- $H(z)\overline{G^*(z)} + H(-z)\overline{G^*(-z)} = 0$
- $G(z)\overline{H^*(z)} + G(-z)\overline{H^*(-z)} = 0$

Given (H, H^*) lowpass filters corresponding to a biorthogonal MRA, let the highpass filters be

$$G(z) = z\overline{H^*(-z)}, \quad G^*(z) = z\overline{H(-z)}.$$

With this choice of dual highpass filters we will have that on Fourier side, the dual wavelets satisfy the scaling equations

$$(\psi)^\wedge(\xi) = G(\xi/2)(\varphi)^\wedge(\xi/2), \quad (\psi^*)^\wedge(\xi) = G^*(\xi/2)(\varphi^*)^\wedge(\xi/2).$$

The corresponding perfect reconstruction filter bank now reads,

$$a_{j+1} \rightarrow \begin{array}{c} \nearrow \\ \\ \searrow \end{array} \begin{array}{c} \boxed{*\tilde{H}^*} \rightarrow \boxed{\downarrow 2} \rightarrow a_j \rightarrow \boxed{\uparrow 2} \rightarrow \boxed{*H} \\ \\ \\ \boxed{*\tilde{G}^*} \rightarrow \boxed{\downarrow 2} \rightarrow d_j \rightarrow \boxed{\uparrow 2} \rightarrow \boxed{*G} \end{array} \begin{array}{c} \searrow \\ \\ \nearrow \end{array} \oplus \rightarrow a_{j+1}.$$

There is also a *fast biorthogonal wavelet transform*.

Let us consider an example. The **linear spline biorthogonal wavelets** are generated by the linear spline (the hat function),

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

This φ satisfies the following scaling equation, see Figure 16:

$$\varphi(t) = \frac{1}{2}\varphi(2t+1) + \varphi(2t) + \frac{1}{2}\varphi(2t-1);$$

and clearly the subspaces $\boxed{\mathbf{V}_j}$ satisfy all the properties described above. One can locate compactly supported linear spline ψ and dual functions φ^*, ψ^* The dual wavelets can be chosen so they have increasing vanishing moments and supports. This is the **Daubechies biorthogonal family**. The functions are all symmetric, continuous and compactly supported. You can see pictures of the scaling and wavelet functions for different biorthogonal completions of the MRA generated by the hat function in Daubechies book [**?**, pp. 273-274]. All these properties are incompatible in the orthogonal case (Haar is the closest example, but is discontinuous). The filters have rational dyadic coefficients which make them very attractive for numerical computations. There is a trick for orthonormalization (not the usual Gram-Schmidt)), that allows to construct a new scaling function which will generate the same MRA but now the integer shifts will be orthonormal, however this function will not have compact support.

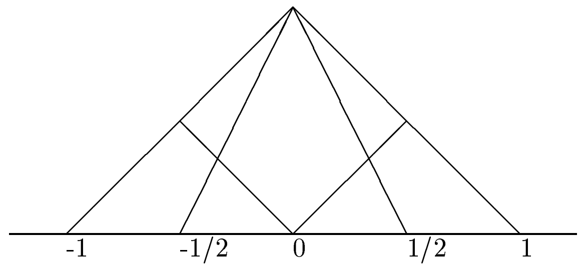

Figure 16: Scaling equation for the hat function.

The **higher-order spline biorthogonal wavelets** are also compactly supported. There are two positive integer parameters N, N^*. N^* determines the scaling function φ^*, it is a spline of order $[N^*/2]$. The other scaling function and both wavelets depend on both parameters. ψ^* is a compactly supported piecewise polynomial of order $N^* - 1$, which is C^{N^*-2} at the knots, and increasing support with N. ψ has also support increasing with N and vanishing moments as well. Their regularity can differ notably. The filter coefficients are dyadic rationals, which makes them very attractive for numerical purposes. The functions ψ are known explicitly. The dual filters are very unequal in length, which could be a nuisance when performing image analysis for example. See [**?**, pp. 271-278]

4.3 Mutation: Multiwavelets

A second mutation is to use more than one scaling function to span $\mathbf{V}_0$, and more than one wavelet to span $\mathbf{W}_0$. An MRA of multiplicity r, is an ordinary MRA but instead of one scaling functions there will be r of them, encoded in the scaling vector $\Phi = (\varphi^1, \ldots, \varphi^r)$. There is a nested sequence of subspaces $\mathbf{V}_j \subset \mathbf{V}_{j+1}$ with all the MRA properties, the only change is that the family $\{\varphi^i(t-k)\}_{k\in\mathbb{Z}, i=1,\ldots,r}$ is an orthonormal basis of $\mathbf{V}_0$. The basic scaling equation reads

$$\Phi(t) = \sum_{k\in\mathbb{Z}} \mathbf{H}_k \Phi_{1,k}(t),$$

where the lowpass filter coefficients $\mathbf{H}_k$ are now $r \times r$ matrices. On Fourier side we have a similar equation,

$$\hat{\Phi}(\xi) = \mathbf{H}(\xi/2)\hat{\Phi}(\xi/2),$$

where $\mathbf{H}(z) = \frac{1}{\sqrt{2}} \sum_k \mathbf{H}_k z^k$, $z = e^{-2\pi i\xi}$. Iterating this equation, and being careful with the order in which the matrices are multiplied,

$$\hat{\Phi}(\xi) = \mathbf{H}(\xi/2)\mathbf{H}(\xi/4)\cdots\mathbf{H}(\xi/2^J)\hat{\Phi}(\xi/2^J).$$

The equation will have a solution if the product converges and $\hat{\Phi}$ is continuous at $\xi = 0$, and moreover, $\hat{\Phi}(0)$ is an eigenvector with eigenvalue one of $\mathbf{H}(\xi = 0)$ (i.e. $H(0)\hat{\Phi}(0) = \hat{\Phi}(0)$). Another necessary condition on the filter to guarantee the existence of a solution to the scaling equation, so that its integer translates form an orthonormal basis of $\mathbf{V}_0$, is a QMF condition: $\boxed{\text{formula below changed: boldface I (identity matrix)}}$

$$\mathbf{H}(z)\overline{\mathbf{H}^t(z)} + \mathbf{H}(-z)\overline{\mathbf{H}^t(-z)} = \sqrt{2}\,\mathbf{I},$$

where here $\mathbf{A}^t$ stands for the transpose matrix of a given matrix $\mathbf{A}$. If the refinement mask $\mathbf{H}(z)$ has a simple eigenvalue $\lambda = 1$, and all its other eigenvalues are strictly less than one in absolute value, then uniform convergence of the infinite product is guaranteed, see [**?**].

A vector $\Psi = (\psi^1, \ldots, \psi^r)$ is an orthonormal multiwavelet if the family $\{\psi^i_{j,k}\}_{j,k\in\mathbb{Z},i=1,\ldots,r}$ is an orthonormal basis of $L^2(\mathbb{R})$. As in the "uniwavelet" case, the existence of a multiresolution analysis in the background greatly symplifies the search for multiwavelets. In the case of uniwavelets it is simple how to find the wavelets once an MRA is given. In the multiwavelet case it is not so straightforward. The theory grew out of specific examples. The first multiwavelet construction is due to Alpert and Hervé. Their functions were piecewise linear, but discontinuous. Using fractal interpolation, Geronimo, Hardin and Massopust [?] succeeded in constructing orthogonal biscaling functions with short support, symmetry and second approximation order, they are called the *GHM biwavelets*. In the mean time Strela, a student of G. Strang at MIT, developed a more systematic theory for constructing multiwavelets given a multi-MRA, his approach being to complete a perfect reconstruction filter bank given the lowpass filter $\mathbf{H}$, see [?], in particular find the highpass filters $\mathbf{G}$. A necessary condition for the completion of the filter bank is

$$\mathbf{H}(z)\overline{\mathbf{G}^t(z)} + \mathbf{H}(-z)\overline{\mathbf{G}^t(-z)} = 0.$$

There is software available for multiwavelets. In particular Strela has built a Matlab based package available in the internet at [web:1]. All these softwares allow you to enter your own wavelets or multiwavelets. All you need is the correct filter coefficients.

4.3.1 Cross Mutation: Biorthogonal Multiwavelets

One can develop a parallel theory of biorthogonal multiwavelets. There are certain complications when one tries to implement a fast multiwavelet transform. In the uniwavelet case, given N samples of a function, one assumes in practice that they are the approximation coefficients at a given scale. This is much less natural in the multiwavelet case. When the multiscaling functions differ substantially, this direct mapping from samples to scaling coefficients introduces high frequency artifacts. A *preprocessing* step is necessary, that will take the samples as input and it will output r vectors of same length or smaller than the input.

There are a couple of preprocessings available:

Oversampling preprocessing: in the biwavelet case it outputs two copies of the data, one of them a scalar multiple, where the scalar has been chosen so that some cancellation occurs. A solution is guaranteed because of the approximation properties. But higher approximation is not preserved.

Matrix preprocessing: In the biwavelet case, the output produces two vectors of half the length of the data (critical rate). Approximation properties can be preserved this way.

Lifting preprocessing: the preprocessing step reduces to a simple polyphase split.

Modified GHM biwavelets: Compactly supported biorthogonal biwavelets this means we have two vector wavelets and two vector scaling functions, each of dimension two. Supported on $[-1, 1]$, in fact one of them is supported on $[0, 1]$. One is symmetric the other antisymmetric. There is a tuning parameter s, and the dual parameter $s^* = \frac{2s+1}{5s-2}$, in the sense that given one matrix filter $\boxed{\mathbf{H}_s}$ the dual filter is given by $\boxed{\mathbf{H}_{s^*}}$. Regularity is governed by s. They have approximation order 2 (or 2 vanishing moments). When $s = -1/5$ they correspond to the orthogonal GHM biwavelet, the first example of a smooth compactly supported biwavelet [?].

4.3.2 Mutated Relative: Alpert Multiwavelets

The Alpert multiwavelet [?] is a compactly supported, orthogonal, piecewise polynomial multiwavelet. Although it is associated with an MRA, it can be constructed without the use of filters.

We will describe the construction on the interval $[0, 1]$. Fix $k \in \mathbf{N}$ and $n > 0$, and let $\mathbf{V}_n$ be the space of (polynomial) functions that are polynomials of degree less than k on the intervals $(2^{-n}j, 2^{-n}(j+1))$ for $j = 0, \ldots, 2^n - 1$, and 0 elsewhere. On each subinterval, $\mathbf{V}_n$ is spanned by k scaling functions, namely the Legendre polynomials up to degree k. By including their shifts we span all of $\mathbf{V}_n$. By the definition of a (orthogonal) MRA, $\mathbf{W}_n$ consists of functions that are polynomials of degree less than k on the intervals $(2^{-n}j, 2^{-n}(j+1))$, and are orthogonal to polynomials of degree less than k on the intervals $(2^{-n+1}j, 2^{-n+1}(j+1))$. We can construct them by a simple Gramm-Schmidt orthogonalization. There is a choice in which set of multiwavelets to choose to span $\mathbf{W}_n$. For example, one could try to give some of them the maximum smoothness across the center of the interval, or have maximal number of vanishing moments. By construction, the wavelets are orthogonal to polynomials of degree $k-1$, so they have k vanishing moments, and so the same sparsity properties as ordinary wavelets.

The wavelets are discontinuous, like the Haar function. This would seem to be a disadvantage, but is actually an advantage for doing PDEs, because it allows weak formulations of the derivative, and better treatment of boundaries. See Section 5.6.2 and [?].

4.4 Wavelets in 2-D

There is a standard procedure to construct a basis in 2-D space from a given basis in 1-D, the *tensor product*. In particular, given a wavelet basis $\{\psi_{j,k}\}$ in $L^2(\mathbb{R})$, the family of tensor products

$$\psi_{j,k;i,n}(x,y) = \psi_{j,k}(x)\psi_{i,n}(y), \quad j,k,i,n \in \mathbb{Z},$$

is an orthonormal basis in $L^2(\mathbb{R}^2)$. Unfortunately we have lost the multiresolution structure. Notice that we are mixing up scales in the above process, that is the scaling parameters i, j can be anything.

We would like to use this idea but at the level of the approximation spaces $\mathbf{V}_j$ in the MRA. For each scale j, the family $\{\varphi_{j,k}\}_k$ is an orthonormal basis of $\mathbf{V}_j$. Consider the tensor products of these functions, $\varphi_{j,k,n}(x,y) = \varphi_{j,k}(x)\varphi_{j,n}(y)$, then let $\mathcal{V}_j$ be the closure in $L^2(\mathbb{R}^2)$ of the linear span of those functions (i.e. $\mathcal{V}_j = \mathbf{V}_j \otimes \mathbf{V}_j$). Notice that we are not mixing scales at the level of the MRA. It is not hard to see that the spaces $\mathcal{V}_j$ form an MRA in $L^2(\mathbb{R}^2)$ with scaling function $\varphi(x,y) = \varphi(x)\varphi(y)$. This means that the integer shifts $\{\varphi(x-k, y-n) = \varphi_{0,k,n}\}_{k,n\in\mathbb{Z}}$ form an orthonormal basis of $\mathcal{V}_0$, consecutive approximation spaces are connected via scaling by 2 on both variables, and the other conditions are clear.

The orthogonal complement of $\mathcal{V}_j$ in $\mathcal{V}_{j+1}$ is the space $\mathcal{W}_j$ which can be seen is the direct sum of three orthogonal tensor products, namely,

$$\mathcal{W}_j = (\mathbf{W}_j \otimes \mathbf{W}_j) \oplus (\mathbf{W}_j \otimes \mathbf{V}_j) \oplus (\mathbf{V}_j \otimes \mathbf{W}_j).$$

Therefore three wavelets are necessary to span the detail spaces,

$$\psi^d(x,y) = \psi(x)\psi(y), \quad \psi^v(x,y) = \psi(x)\varphi(y), \quad \psi^h(x,y) = \varphi(x)\psi(y),$$

where d stands for diagonal, v for vertical, and h for horizontal (the reason for such names is that each of the subspaces will somehow favor details in those directions).

As an example consider the **2-D Haar basis**. The scaling function is the characteristic function of the unit square,

$$\varphi(x,y) = \chi_{[0,1]^2}(x,y).$$

The following pictures should suffice to understand the nature of the 2-D Haar wavelets and scaling function:

1	1
1	1

$\varphi(x,y)$

-1	1
1	-1

$\psi^d(x,y)$

1	-1
1	-1

$\psi^h(x,y)$

-1	-1
1	1

$\psi^v(x,y)$

This construction has the advantage that the basis are separable, and so implementing the fast two dimensional wavelet transform is not difficult. In fact it can be done by successively applying the one dimensional FWT. The disadvantage is that the analysis is very axis dependent, which might not be desirable for certain applications. In higher dimensions the same construction works. There will be one scaling function, and $2^n - 1$ wavelets, where n is the dimension.

There are *non-separable* two dimensional MRAs. The most famous one corresponds to an analogue of the Haar basis. The scaling function is the characteristic function of a two dimensional set. It turns out that the set has to be rather complicated, in fact it is a self-similar set with fractal boundary, the so-called *twin dragon*. There is an applet for viewing the twin dragon at [web:12].

There is nothing sacred about the dilation factor 2. In n-dimensions one can think of a dilation matrix, which in the tensor product case corresponds to the matrix $2^n I$. One can read from the dilation matrix the number of wavelet functions that will be necessary. Moreover the lattice $\mathbb{Z}^n$ can be replaced by any general lattice in $\mathbb{R}^n$. Necessary and sufficient conditions in terms of the refinement mask and the filter for the existence of an associated wavelet basis are given by Carlos Cabrelli and Maria Luisa Gordillo [**?**]. They prove these results for orthonormal regular multiwavelets in any dimension and for an arbitrary dilation matrix $\mathbf{A}$ and lattice Γ. These wavelets are associated to an MRA of multiplicity r (i.e. r scaling functions). They show that when such wavelet basis exists, then it is necessary to have $(\det(\mathbf{A})-1)r$ wavelet functions. Moreover, if $2r(\det(\mathbf{A})-1) \geq n$, then the necessary and sufficient conditions hold, and the wavelets are at least as regular as the scaling functions.

Remember that an arbitrary lattice Γ in $\mathbb{R}^n$ is given as the image under any invertible $n \times n$ matrix S of the usual integer lattice $\mathbb{Z}^n$. $\mathbf{A}$ is a dilation matrix for Γ if $\mathbf{A}(\Gamma) \subset \Gamma$, and every eigenvalue λ of $\mathbf{A}$ is strictly larger than one. At the level of the MRA, one should adapt the scaling and translation scheme to this setting. Namely, $g(x) \in \mathcal{V}_j \leftrightarrow g(\mathbf{A}x) \in \mathcal{V}_{j+1}$ for each $j \in \mathbb{Z}$; and for an MRA of multiplicity r, there exist scaling functions $\varphi_1, \ldots, \varphi_r \in L^2(\mathbb{R}^n)$ such that the collection of lattice translates $\{\varphi_i(x-k)\}_{k\in\Gamma, i=1,\ldots,r}$ forms an orthonormal basis for $\mathcal{V}_0$.

4.4.1 Mutation: Wavelets for Image Processing

Images can be quite complicated. Edges and textures can exist in all possible locations, directions and scales. Wavelets do not have a good angular resolution. Wavelet packets provide more flexibility, see Section 4.7. However the tensor product wavelets introduce artifacts. *Directionally oriented filter banks* have been used for image processing, but they do not allow for an arbitrary partition of the Fourier plane. *Steerable filters* with arbitrary orientation have been designed, but they are overcomplete, and not orthogonal. Several wavelet type solutions have been studied, in an ever expanding zoo of "objectlets".

Coifman and F. Meyer (the son of Yves Meyer) introduced **brushlets** [**?**] in order to obtain better angular resolution than wavelet packets. The idea is to expand the Fourier plane into windowed Fourier bases. A *brushlet* is a function reasonably well localized with only one peak in frequency (tensor product bases have always two peaks). The brushlets are complex valued, and their phase provides information about the orientation. One can adaptively select the size and locations of the brushlets in order to obtain good compression ratios.

The **wedgelets** [**?**] are a collection of dyadically organized indicator functions with a variety of locations, scales, and orientations. They are used to estimate the location of a smooth edge in a noisy image.

The **beamlets** [**?**] dictionary is a dyadically organized collection of line segments, occupying a range of dyadic locations and scales, and occuring at a range of orientations. The beamlet transform of an image $f(x, y)$ is a collection of integrals of f over each segment in the beamlet dictionary; the resulting information is stored in a beamlet pyramid. The beamlet graph is a graph structure with pixel corners as vertices and beamlets as edges; a path through this graph corresponds to a polygon in the original image. By exploiting the first four components of the beamlet framework, they can formulate beamlet-based algorithms which are able to identify and extract beamlets and chains of beamlets with special properties. These algorithms can be shown in practice to have surprisingly powerful and apparently unprecedented capabilities, for example in detection of very faint curves in very noisy data.

The **ridgelets** [**?**] are a smooth, orthonormal basis of $L^2(\mathbb{R}^2)$ that is designed to efficiently represent functions that have singularities/discontinuities that lie along ridges. They are constructed as the inverse Radon transform of specially modified radially symmetric wavelets. For kill a the theoretical basis behind ridgelets, see the the PhD Thesis of Donoho's former student, Emanuel Candés, [**?**], and a more recent results in [**?**].

Donoho and his student E. Candès have recently constructed a tight frame of **curvelets** [**?**] which provides stable, efficient and near-optimal representation of smooth objects having discontinuities along smooth curves. By naively thresholding the curvelet transform, they obtain approximation rates comparable to complex adaptive schemes which try to track the discontinuity set. For a descriptive account see [**?**].

4.5 Wavelets on the interval

For many problems one would like to have wavelet bases on $L^2([0,1])$. Images live in bounded rectangles for instance. The Haar functions that live in the interval provide an orthonormal wavelet basis of $L^2([0,1])$ if one includes the constant function on the interval. Another way of thinking of this is to notice that the restriction to the unit interval of a basis on $L^2(\mathbb{R})$ will be a basis in $L^2([0,1])$. In the Haar case, orthonormality is preserved, but not in general. There are several approaches to this problem, more information and pointers to the literature can be found in [**?**, Section 7.5].

Periodic wavelets: Wavelets are periodized using the usual periodization trick,

$$f^{\mathrm{per}}(t) = \sum_{k\in\mathbb{Z}} f(t+k)$$

and restricting to the unit interval. For each scale $j > 0$ there are 2^j periodized wavelets indexed by $0 \le n < 2^j$. The restriction to $[0,1]$ preserves those wavelets whose support was contained in $[0,1]$, and modifies those whose support overlapped the boundaries $t = 0$ or $t = 1$. Those periodized wavelets together with the periodized scaling functions at the coarsest scale ($j = 0$ in our case means that only one scaling function needs to be periodized, like in the Haar example) form an orthonormal basis of $L^2([0,1])$. Another way of seeing that this is indeed a basis is extending the function f to be zero outside of $[0,1]$, performing its wavelet transform in $L^2(\mathbb{R})$, then periodizing, and this shows that the periodized wavelets are indeed a basis in $L^2([0,1])$.

Periodic wavelet basis have the disadvantage of creating large amplitude coefficients near $t = 0$ and $t = 1$, because the boundary wavelets have separate components with no vanishing moments. If $f(0) \neq f(1)$ the wavelet coefficients behave as if the signal were discontinuous at the boundaries. On the other hand, implementations are very simple, so this method is often used.

Folded wavelets: Here one folds the function defined on $[0,1]$ and extended to be zero outside, to avoid discontinuities in the boundaries, and extends it into a 2-periodic function:

$$f^{\mathrm{fold}}(t) = \sum_{k\in\mathbb{Z}} \big[f(t-2k) + f(2k-t)\big].$$

This function coincides with the original function on $[0,1]$. f^{fold} is continuous on the boundary, but even if f is continuously differentiable its derivative will be discontinuous on the boundary, unless $f'(0) = f'(1) = 0$. Decomposing f^{fold} with a wavelet basis is the same as decomposing f in the folded wavelet basis. One can verify that

$$\int_0^1 f(t)\psi_{j,k}^{\mathrm{fold}}(t)dt = \int_{\mathbb{R}} f^{\mathrm{fold}}(t)\psi_{j,k}(t)dt.$$

Because of the continuity at the boundaries, the boundary wavelet coefficients are smaller than in the periodized case. But the discontinuity of the derivative still makes them bigger than the interior coefficients.

To construct a basis of $L^2([0,1])$ of folded wavelets, it is sufficient for ψ to be either symmetric or antisymmetric with respect to $t = 1/2$. Unfortunately the Haar basis is the only real symmetric compactly supported orthogonal basis. On the other hand we can obtain compactly supported basis if we drop the orthogonality assumption, and content ourselves with biorthogonal basis. We could increase the multiplicity and consider multiwavelets, and then again, we can find them with all the desired properties.

Modified filter banks can be implemented, where the boundaries are a little bit more complicated to handle than in the periodic case. For more details, see [**?**, pp. 284-286].

Boundary wavelets: The previous methods created boundary wavelets that have at most one vanishing moment. This means that the boundary coefficients can be large even if the signal is very smooth near the boundary. Wavelets adapted to "life in the interval" are required. Boundary wavelets that have as many vanishing moments as the original wavelet were first introduced by Yves Meyer and later refined by Cohen, Daubechies and Vial [**?**].

The idea behind this construction is to construct a multiresolution analysis in $L^2([0,1])$, $\{\mathbf{V}_j^{int}\}_{j\geq 0}$, from an orthogonal MRA in $L^2(\mathbb{R})$ such that the corresponding wavelet has M vanishing moments. We want that property to be preserved. Notice that wavelets have M vanishing moments if they are orthogonal to polynomials of degree $M-1$. Since the wavelets at scale 2^{-j} in the interval are orthogonal to $\mathbf{V}_j^{int}$, if one can guarantee that polynomials of degree $M-1$ are in $\mathbf{V}_j^{int}$, then that will ensure the vanishing moments of the wavelets.

One can construct such an MRA on the interval starting with Daubechies compactly supported wavelet with M vanishing moments. It has support of length $2M-1$. At scale $2^{-j} \leq 1/2M$ there are $2^j - 2M$ scaling functions with support completely inside $[0,1]$. Those are not touched. To construct an approximation space $\mathbf{V}_j^{int}$ of dimension 2^j, M functions are added with support on the left boundary, and M with support on the right boundary, in such a way that it is guaranteed that the restrictions to $[0,1]$ of polynomials of degree $M-1$ are in $\mathbf{V}_j^{int}$. The scaling functions are specified by discrete filters, which are adjusted at the boundaries. With some more work the wavelets can be constructed, again at scale 2^{-j} there will be $2^j - 2M$ interior wavelets, and M right and M left wavelets. The filter coefficients can be computed as well and a fast transform can be implemented. For those coefficients that correspond to truly interior wavelets, the cascade algorithm is used. At the boundary the filters must be replaced by the boundary filters. The implementation is more complicated than in the periodized or folding cases, but it does not require more computations.

Multiwavelets can be adapted to life in the interval. For further information can see the work of Dahmen et al [**?**, **?**] in the spirit of the (CDV) boundary wavelets. You can also see the work of Lakey and Pereyra [**?**], whose biwavelets are minimally supported and symmetric, and allow one to construct boundary wavelets by just truncating the wavelets in the line, in the spirit of the folding technique.

Construction of wavelet-like basis on bounded domains of $\mathbb{R}^n$ is an active area. In terms of characterizing Besov spaces on domains, see the work of Cohen, Dahmen and De Vore [**?**]. Second generation wavelets on irregular point sets in one and two-dimensions are discussed by Daubechies, Guskov, Schröder and Sweldens [**?**].

4.6 Mutation: Lifting Schemes

The filter banks described earlier relied heavily on the Fourier transform for understanding their convergence properties. Also for implementations, convolutions are completely decorrelated on Fourier side (products), and both up and downsampling have nice representations on Fourier domain. However, there are many situations were the Fourier transform is not available because the context is not euclidean (translation and shift invariant), for example:

- Curves and surfaces,
- Bounded domains,
- Weighted spaces,

- Non-uniform samplings.

It is desirable to have algorithms that are independent of the Fourier transform, where all the action occurs in the time domain instead of on the frequency domain. Wim Sweldens' *lifting algorithm* accomplishes exactly that [?]. This algorithm speeds up the wavelet transform [?] and allows one to build *second generation wavelets* [?]. The transform works for images of arbitrary size with correct treatment of the boundaries. Also, all computations can be done in-place.

The basic idea is simple, one starts with a trivial wavelet called the *Lazy wavelet*. Step by step a new wavelet is built gradually improving its properties. It all amounts to a clever matrix factorization which has been long known to algebraists. For a *five minute tour of lifting* check [?] Wim Sweldens, *Wavelets and the lifting scheme: A 5 minute tour* Zeitschrift fur Angewandte Mathematik und Mechanik, vol. 76 (Suppl. 2), pp. 41-44, 1996.

both references I found in his webpage, I cut and pasted and some funny german characters went into the file unnoticed by me ;-) I just tried to check reference in MATHREVIEW but netscape crashed...

Peter Schröder and Wim Sweldens [?] P. Schröder, W. Sweldens, *Spherical wavelets: Efficiently representing functions on a sphere.* Computer Graphics Proceedings (SIGGRAPH 95), pp. 161-172, 1995 have used lifting to develop biorthogonal **wavelets bases on the sphere** with various properties. According to them, the bases are very easy to implement and allow fully adaptive subdivisions. They give examples of functions defined on the sphere, such as topographic data, bi-directional reflection distribution functions, and illumination, and show how they can be efficiently represented with spherical wavelets.

For more information on lifting, see Wim Sweldens webpage [web:6], there you can retrieve electronically most of the papers cited in this section. Sweldens was the editor of the *The Wavelet Digest*, a free monthly electronic newsletter which contains all kinds of information concerning wavelets: announcement of conferences, preprints, software, questions, etc. To receive copies of the digest you can register online at [web:4].

There is software available: Liftpack in C; Claypoole at dsp.rice for matlab code. There is also Manuel Leon lifting software that can be retrieved from [web:5].

Strela et al [?] have shown how to use lifting in the multiwavelet case.

4.7 Relative: Wavelet packets

A function with a sustained high frequency, such as that shown in Figure 5, is a problem for wavelets, since the number of significant coefficients will be proportional to the number of oscillations . To enable wavelets to handle such functions, "wavelet packets" were developed.

To perform the wavelet transform we iterate at the level of the lowpass filter (approximation). The details (wavelet coefficients) are not touched. In principle this is an arbitrary choice, and one could iterate at the highpass filter level, or any desirable combination. If we iterate both the high and lowpass n times, then the resulting binary tree encodes information for 2^{2^n} different bases. Denote the spaces by $\mathbf{W}_{j,n}$, where j is the scale as before, and n determines the "frequency". The full wavelet packet binary tree with 3 levels is

$$\begin{array}{ccccc} & & & \nearrow & \mathbf{W}_{-2,3} \\ & \nearrow & \mathbf{W}_{-1,1} & & \\ & & & \searrow & \mathbf{W}_{-2,2} \\ \mathbf{W}_{0,0} & & & & \\ & & & \nearrow & \mathbf{W}_{-2,1} \\ & \searrow & \mathbf{W}_{-1,0} & & \\ & & & \searrow & \mathbf{W}_{-2,0} \end{array} \quad . \tag{22}$$

Each of the spaces is generated by the integer shifts of a wavelet function at scale j and frequency n. More precisely, let $\omega_{j,k,n}(t) = 2^{j/2}\omega_n(2^j t - k)$, where $n \in \mathbb{N}$, $j, k \in \mathbb{Z}$, and

$$\omega_{2n}(t) = \sqrt{2}\sum h(k)\omega_n(2t-k), \quad \omega_0 = \varphi;$$
$$\omega_{2n+1}(t) = \sqrt{2}\sum g(k)\omega_n(2t-k), \quad \omega_1 = \psi.$$

Then $\mathbf{W}_{j,n} = \mathrm{span}\{\omega_{j,k,n} : k \in \mathbb{Z}\}$. For a graphical view of the possible filtering steps, see Figure 17.

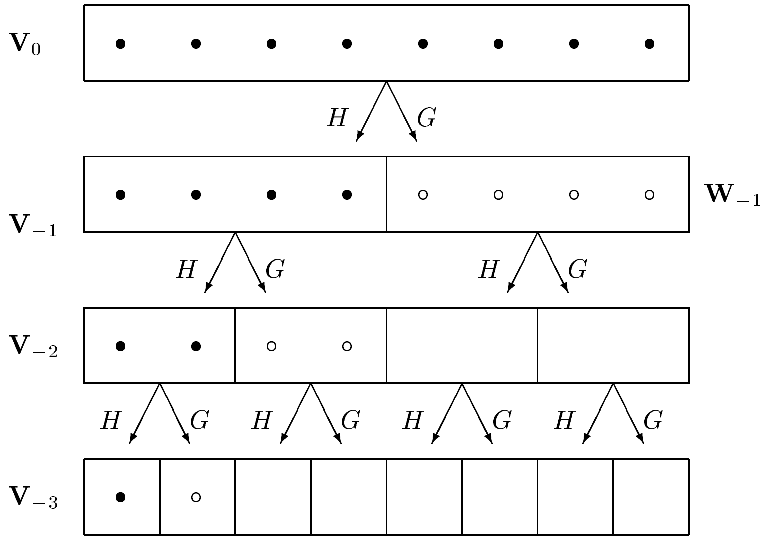

Figure 17: The filtering choices available for wavelet packets. Compare to Figure 11 for wavelets.

Notice that $\mathbf{W}_{0,0} = \mathbf{V}_0$, and more generally, $\mathbf{W}_{j,0} = \mathbf{V}_j$, and $\mathbf{W}_{j,1} = \mathbf{W}_j$. We also know that the spaces $\mathbf{W}_{j-1,2n}$ and $\mathbf{W}_{j-1,2n+1}$ are orthogonal and their direct sum is $\mathbf{W}_{j,n}$. Therefore the leaves of every connected binary subtree of the wavelet packet tree correspond to an orthogonal basis of the initial space. Graphically this means that any choice of decompositions that covers the interval gives a wavelet packet representation. Each of the bases encoded in the wavelet packet corresponds to a dyadic tiling of the phase plane in Heisenberg boxes of area one. They provide a much richer time/frequency analysis, so by choosing which spaces to filter, we can match the behavior of our target function. For example, the choices in Figure 18 gives the phase plane in Figure 19.

For the example of the Haar wavelet packet, the equations become

$$\begin{aligned} \omega_{2n}(t) &= \omega_n(2t) + \omega_n(2t-1) \\ \omega_{2n+1}(t) &= \omega_n(2t) - \omega_n(2t-1). \end{aligned}$$

The functions so obtained are the Walsh functions, which are sort of discretized versions of the sines and cosines. A good source of information for this topic is [**?**].

4.8 Mutant Orphans

The **Noiselets** [**?**] are a basis that is designed to appear to be noise to the Haar-Walsh wavelet packets. This means that they are incompressible in this basis. It also means that they do not interfere with the information in a Haar-Walsh signal, and thus it may be possible for example to transmit a Haar-Walsh signal and a noiselet signal at the same time without interference.

When talking about lifting, we mentioned that it was attractive because it allowed for construction of wavelet type basis on non-uniform grids. The local cosine and sine basis can be developed on non-uniform grids as well, which makes them attractive. Further developments to produce **squeezable bases on nonuniform grids** have been carried on by G. Donovan, J. Geronimo and D. Hardin [**?**] They exploit the minimal support and symmetry properties of multiwavelets for the purpose of constructing basis on non-uniform grids. They generate local orthogonal bases on arbitrary partitions of $\mathbb{R}$ from a given basis via what they call a *squeeze map*. They can control the squeeze map to get a non-uniform basis that preserves smoothness and/or accuracy.

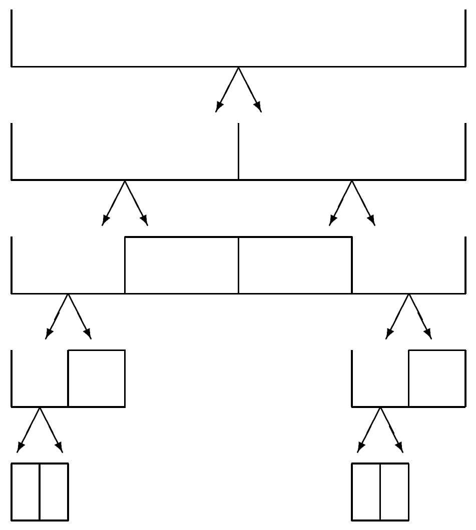

Figure 18: A possible choice for a wavelet packet decomposition

5 Assorted Applications

5.1 Basics of Compression

One of the main interests in signal and image processing is to be able to code the information with as little data as possible. That allows for rapid transmission, etc.

In traditional approximation theory there are to two possible methods, *linear* and *non-linear* approximation.

Linear approximation: This refers to selecting a priori N elements in the basis and projecting onto the subspace generated by those elements, regardless of the function that is being approximated, this is a linear scheme,

$$P_N^l f = \sum_{n=1}^{N} \langle f, \psi_n \rangle \psi_n.$$

Non-linear approximation: This approach chooses the basis elements depending on the function, for example choose the N basis elements so that the coefficients are the largest in size for the particular function. This time the chosen basis elements will depend on the particular function to be approximated,

$$P_N^{nl} f = \sum_{n=1}^{N} \langle f, \psi_{n,f} \rangle \psi_{n,f}.$$

The non-linear approach has proven quite successful. There is a lot more information about these issues in [**?**, Chapters 9 and 10]. The basic steps are to:

- Transform the data, finding coefficients in a given basis.
- Threshold the coefficients. (Essentially one keeps the large one and discards the small ones, information is lost in this step, so perfect reconstruction is not possible.)

The coefficients can then be transmitted, stored, etc. (In real applications, they would first be quantized and coded, but that is another issue.)

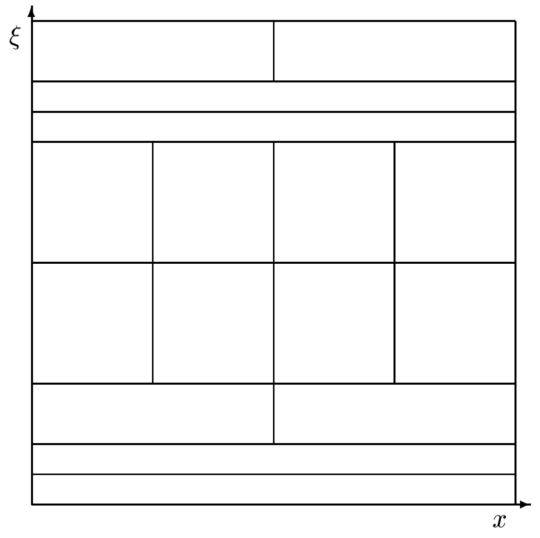

Figure 19: The wavelet packet phase plane corresponding to Figure 18.

5.2 Basics of Denoising

Wavelet basis are good for decorrelating coefficients. They are also good for denoising in the presence of *white noise*. The crudest approach would be to use the projection into an approximation space as your compressed signal, discarding all the details after certain scale j,

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

The noise is usually concentrated in the finer scales (higher frequencies!), so this approach would denoise but at the same time it will remove many of the sharp features of the signal that were encoded in the finer wavelet coefficients.

The basic philosophy behind wavelet denoising is that noise will be poorly approximated by wavelets, and so reside in the small coefficients. The signal, however, will be in the large coefficients. The process of removing or modifying the small coefficients is called *thresholding*. There are different thresholding techniques. The most popular are *hard thresholding* (it is a keep or toss thresholding), and *soft thresholding* (the coefficients are attenuated following a linear scheme). There is also the issue about thresholding individual coefficients or *block-thresholding* (more about this issue in the last lecture). How to select the threshold is another issue. In the denoising case, there are some thresholding selection rules which are justified by probability theory (basically the law of the large numbers), and are used widely by statisticians:

- Selection using Stein's unbiased risk estimate (SURE).
- Universal threshold by Donoho.
- Selection based on the minimax principle.

See [?] and the WAVELAB webpage [?].

So to denoise, you essentially compress, and then reconstruct with the remaining coefficients, and hope that you have obtained a good approximation to your original signal and have successfully denoised it.

5.3 Best Basis Searches

The wavelet packets in Section 4.7 can decompose a signal of length $N = 2^J$ in 2^N different ways, which is the number of binary subtrees of a complete binary tree of depth J. A tree search can find the "best basis" within all these possible bases. Furthermore the search can be performed in $O(N\log(N))$ operations. The citerion used for “best” needs to be specified.

Functionals verifying an additive-type property are well suited for this type of search. Coifman and Wickerhauser introduced a number of such functionals [?], among them some *entropy criteria*. Given a signal s and $(s_i)_i$ its coefficients in an orthonormal basis, the entropy E must be an additive cost function such that $E(0) = 0$ and $E(s) = \sum_i E(s_i)$. Matlab encodes four different entropy criteria:

- The Shannon entropy: $E_1(s_i) = -s_i^2 \log(s_i^2)$.
- The concentration in ℓ^p norm with $1 \le p \le 2$: $E_2(s_i) = |s_i|^p$.
- The logarithm of the “energy” entropy: $E_3(s_i) = \log(s_i^2)$.
- The threshold entropy: $E_4(s_i) = 1$ if $|s_i| > \epsilon$ and 0 otherwise. So $E_4(s)$ counts the number of coefficients that are above a given threshold.

By finding the best basis, we can do a more effective job at compressing, denoising, or meeting some other objective.

5.4 Calculating Derivatives using Biorthogonal Wavelets

Given a biorthogonal MRA, with compactly supported dual scaling functions, one of them smooth, then one can find another biorthogonal MRA related to the original one by differentiation and integration [?]. The dual wavelets are the derivative and antiderivative of the old ones up to a constant. The scaling functions are not exactly derivatives and antiderivatives of the $\boxed{\text{original}}$ ones, but are related by finite differences to them. More precisely, let φ, φ^* be the dual scaling functions and H, H^* the refinement masks of the initial MRA; and ψ, ψ^* the dual wavelets and G, G^* their masks. Let φ_-, φ_+^* be the new scaling functions and H_-, H_+^* their refinement masks; ψ_-, ψ_+^* the new dual wavelets and G_-, G_+^* their masks (the “+” indicates more smoothness, integration; the “−” less smoothness, differentiation). Table 2 has the formulas relating the old and new scaling and wavelets functions and their masks.

smoothened scaling	$H_+^*(z) = \frac{1+z}{2z} H^*(z)$	$\frac{d}{dx}\varphi_+^*(x) = \varphi^*(x+1) - \varphi^*(x)$
roughened scaling	$H_-(z) = \frac{2}{1+z} H(z)$	$\frac{d}{dx}\varphi(x) = \varphi_-(x) - \varphi^-(x-1) = \Delta_-\varphi$
smoothened wavelet	$F_+^*(z) = \frac{z}{2(z-1)} F^*(z)$	$\frac{d}{dx}\psi_+^*(x) = -\psi^*(x)$
roughened wavelet	$F_-(z) = 2(1-z)F(z)$	$\frac{d}{dx}\psi(x) = \psi_-(x)$

Table 2: Smoothened and roughened biorthogonal scalar MRAs

It is not hard to see that if the original filters satisfy the biorthogonality conditions, so will the new ones. Furthermore, the compact support of the wavelets is preserved, and for the scaling functions is still compact but one unit longer. Of course when playing this game one wavelet gains an order of smoothness but looses one vanishing moment, the other looses an order of smoothness and gains a vanishing moment.

One could apply this recipe to any known compactly supported wavelet as long as it has enough smoothness. We will record the scalar filters as polynomials (in z^n where $n \in Z$) whose coefficients are the data utilized by the wavelet toolbox. Remember also that z is in the unit disc, therefore, $\overline{z} = z^{-1}$. The highpass filters G, G^* are found from the lowpass filter H, H^* by the usual conjugate flip, more precisely: $G(z) = z\overline{H^*(-z)}$ and $F^*(z) = z\overline{H(-z)}$. Notice that given the new lowpass filter H_-, H_+^*, we could compute the highpass filters F_-, F_+^* using the conjugate flip trick and we will obtain the same filters as recorded in the table up to a factor of 4, more precisely,

$$F_-(z) = \frac{(1-z)}{2}F(z), \quad \frac{d}{dx}\psi(x) = 4\psi_-(x); \quad F_+^*(z) = \frac{2z}{(z-1)}\tilde{F}^*(z), \quad \frac{d}{dx}\psi_+^*(x) = -4\psi^*(x). \tag{23}$$

These last formulas were the ones obtained by Lemarié [?], and these are the formulas we will use when utilizing MATLAB, since MATLAB computes scalar highpass filters from the lowpass filters by the conjugate flip trick. We will use H^* and H_+^* as the decomposition filters and H and H_- as the reconstruction filters.

As an example, consider the *biorthogonal symmetric spline, Bior3.1*, with parameter $\tilde{N} = 3$, $N = 1$; as described in [?, pp. 271-278]. This particular class has the virtue that the coefficients are very simple dyadic fractions. These biorthogonal wavelets are symmetric and have relatively small support and enough smoothness so that we can differentiate and still get a continuous function. We record in Table 3 the lowpass filters and coefficients.

lowpass dec.	$\tilde{H}(z) = -(4z)^{-1}(1+z)(1-4z+z^2)$	$(0, -1/4, 3/4, 3/4, -1/4)$
lowpass rec.	$H(z) = (8z)^{-1}(1+z)^3$	$(0, 1/8, 3/8, 3/8, 1/8)$
smoothed lowpass dec.	$H_+(z) = -(8z^2)^{-1}(1+z)^2(1-4z+z^2)$	$(-1/8, 1/4, 3/4, 1/4, -1/8)$
roughened lowpass rec.	$H_-(z) = (4z)^{-1}(1+z)^2$	$(0, 1/4, 1/2, 1/4, 0)$

Table 3: Bior3.1/Lemarié lowpass filters and coefficients.

How can this machinery be used to compute derivatives, or antiderivatives? Decompose your function in the biorthogonal basis, formula below changed

$$f = \sum_k \langle f, \varphi^*_{J,k} \rangle \varphi_{J,k} + \sum_{j=J}^{\infty} \sum_k \langle f, \psi^*_{j,k} \rangle \psi_{j,k}.$$

Now calculate the derivative term by term, a factor of 2^j will appear in each summand because of the chain rule, and use Lemarié formulas to replace the derivatives of φ and ψ, formula below changed

$$f' = 4 \sum_k 2^J \langle f, \varphi^*_{J,k} \rangle \Delta_- \varphi^-_{J,k} + 4 \sum_{j=J}^{\infty} \sum_k \langle f, \psi^*_{j,k} \rangle \psi^-_{j,k}.$$

The coefficients for the details are just the old coefficients rescaled. As for the approximation part, one has to reorder, so the coefficients will be differences of consecutive ones rescaled. It means we can compute the derivative using the fast wavelet transform for the new MRA.

This construction can be done in the context of multiwavelets; see [?].

5.5 Divergence-free Wavelets and Multiwavelets

Lemarié [?] used the multiresolution analyses described in the previous section to construct *divergence-free biorthogonal (vector field) wavelet bases*. The one-dimensional wavelets described in Section 5.4 were used as building blocks to produce two or three-dimensional wavelets by appropriate tensor products, and then these were used to create the components of the two or three-dimensional divergence-free vector fields.

Lemarié proved that in dimension two, one could not create these bases so that the wavelets and their duals are compactly supported, have both some smoothness and one of them is divergence-free.

Lakey and Pereyra extended this result to multiwavelets and any dimension [?]. They were hoping to reconcile all the desired properties in the framework of multiwavelets. Unfortunately that was not the case, but in the process they came up with a family of biorthogonal multi-MRAs related by integration and differentiation whose multiwavelets and multiscaling functions have minimal support and various symmetries which turned out to be very useful for constructing wavelets on the interval with little work. Again by tensor products and appropriate choice of components these lead to the construction of divergence-free multiwavelets on the unit box [?]. The basic biorthogonal multiwavelet building blocks have been used for some applications in statistics [?] S. Efromovich, J. D. Lakey, M. C. Pereyra, N. Tymes Jr., *Data-driven and optimal denoising of a signal and recovery of its derivative using multiwavelets.* IEEE Transactions on Signal Processing, vol 52, no. 3, pp. 1-8 (2004).

The hope with all these constructions is that one could use these divergence-free bases to analyze uncompressible fluids, and turbulent flows. Some work has been done in that direction with these and other similar bases [?, ?].

5.6 Applications to Differential Equations

In this section we very briefly describe how wavelets could be used for the study of differential equations. There have been some results using wavelets to advance the theory of differential equations. For instance, Marco Cannone and Yves Meyer have been able to use the wavelet approach (Littlewood-Paley theory) to find new self-similar solutions of the Navier-Stokes equations [?], among other things. There is an encyclopedic book by Lemarié that treats recent developments in the Navier-Stokes problem using harmonic analysis techniques, in particular wavelets [?] P.G. Lemarié-Rieusset, *Recent developments in the Navier-Stokes problem.* Chapman & Hall/CRC, 2002. We will concentrate, however, on the computational side.

After their resounding successes in signal processing, wavelets were touted as the solution to all the world's problems. As yet, they have not been very successful in PDEs, and the bold claims about them have caused resentment from those using well-established techniques. In many cases, the beneficial properties of wavelets, such as adaptivity and multiscale behavior, are already present in these techniques. Before anyone will abandon the huge existing codes and write wavelet versions, the advantage of wavelets must be absolutely clear.

5.6.1 Galerkin Methods using Wavelets

To illustrate the basic issues around using wavelets for differential equations, we first consider a standard one-dimensional boundary-value problem

$$\begin{aligned} u''(t) &= f(t, u, u'), \\ u(0) &= u(1) = 0\,. \end{aligned}$$

A Galerkin method chooses a set of functions $\{v_i(t)\}_{i=1}^n$ and assumes that

$$u(t) \approx v(t) = \sum_{i=1}^{n} c_i v_i(t)$$

for some coefficients c_i. The goal is then to find the coefficients so that $v(t)$ approximately solves the given equation. Let P denote the projection onto the span of $\{v_i(t)\}$. We try to solve

$$\begin{aligned} P(v''(t) - f(t, v, v')) &= 0, \\ v(0) = v(1) &= 0\,. \end{aligned}$$

Typically, the resulting system of equations for $\{c_i\}$ will be inconsistent, so they must be solved in some approximate sense (often least-squares).

The first property that one wants for $\{v_i\}$ is that they allow you to approximate $u(t)$ well with few terms. The localization and vanishing moments properties of wavelets allows one to argue that, under certain conditions, they will have this property.

The second property that one wants is to satisfy the boundary conditions in a straightforward way. For this we need a wavelet basis adapted to the interval, as in Section 4.5. These constructions are not completely satisfactory in dimension one, and their generalization to domains in higher dimensions is problematic. Treatment of the boundary is the greatest weakness of wavelet approaches.

There is a whole wealth of work done by W. Dahmen and collaborators A. Kunoth, K. Urban, etc. They have systematically explored Galerkin-wavelet methods for linear and non-linear equations. You can start a search from Wolfang Dahmen's webpage [web:7]. See also [?, Ch. 6].

5.6.2 Operator Calculus Approaches for PDEs

Many partial differential equations, such as the incompressible Navier-Stokes equations can be written in the form $u_t = \mathcal{L}u + \mathcal{N}(u)$, where $\mathcal{L}$ is the linear part and $\mathcal{N}(u)$ is the nonlinear part. One can then apply the semi-group approach to obtain a solution of the form

$$u(x,t) = e^{(t-t_0)\mathcal{L}} u_0(x) + \int_{t_0}^{t} e^{(t-\tau)\mathcal{L}} \mathcal{N}(u(x,\tau)) d\tau.$$

It turns out that if $\mathcal{L}$ is self-adjoint, strictly elliptic operator, then the operator $e^{t\mathcal{L}}$ is sparse in wavelet basis as well (for a finite but arbitrary precision) for all $t \geq 0$. Therefore using exponentials of such operators for numerical purposes is reasonable.

This is an example of a "numerical operator calculus" using wavelets. This approach originated in [**?**] and has been developed by Beylkin and his collaborators [**?**, **?**, **?**, **?**, **?**, web:2]. For PDEs, the key to handling the boundary conditions is to use the Alpert multiwavelets from Section 4.3.2. To extend the techniques efficiently to higher dimensions, the methods in [**?**] were developed.

6 References and Further Reading

Lemarie and Lemarie-Rieusset are the same person, we have two references [Lem91] and [LR92], and I introduced a third reference... perhaps we should have all of them [Lemxxxx]?

There are now many books, tutorials, articles, software libraries, and websites related to wavelets. In order to help you choose what to look at next, we list and comment on some of these.

There are books available at several levels. [**?**] is a delightful non-technical account of wavelets and their history. [**?**] is an introduction specialize to science and engineering. [**?**] is a mathematical undergraduate text. [**?**] is an undergraduate text aimed at engineers. [**?**] is a graduate level text, with emphasis on the mathematical theory. [**?**] is also graduate level, and has lots of applications. [**?**] and [**?**] are advanced books, oriented towards engineering and computer science. [**?**] is the classic, showing the construction for the Daubechies family of wavelets. [**?**] is also a classic, with research level mathematics.

There are both commercial and free software packages available. The main commercial one is the *Matlab Wavelet Toolbox* [web:9], which is an addition to Matlab, and has extensive documentation and a manual available. If you have the base Matlab, but do not want to pay for the toolbox, you can use *Wavelab* [web:10], a Stanford based free software package. *Lastwave* [web:11] is a free toolbox with subroutines written in C, created at Ecole Polytechnique.

Internet References

[web:1] `http://www.mcs.drexel.edu/~vstrela/MWMP/`

[web:2] `http://amath.colorado.edu/faculty/beylkin/`

[web:3] `http://www.siam.org/siamnews/mtc/mtc593.htm`

[web:4] `http://www.wavelet.org/add.html`

[web:4] `http://www.jpeg.org/FCD15444-1.htm`

[web:5] `http://www.math.nmsu.edu/~jlakey/nmas.html`

[web:6] `http://cm.bell-labs.com/who/wim/`

[web:7] `http://elc2.igpm.rwth-aachen.de/~dahmen/`

[web:8] `http://www.fftw.org/`

[web:9] `http://www.mathworks.com/products/wavelet/`

[web:10] `http://www-stat.stanford.edu/ wavelab/`

[web:11] `http://www.cmap.polytechnique.fr/ bacry/LastWave/index.html`

[web:12] `http://herodot.informatik.uni-mannheim.de/appverw/appletlist2.php`