Preface

This book is about filter banks and wavelets. Those are new ways to see and represent a signal. They are alternatives to the Fourier transform, using short wavelets instead of long waves. We will explain the advantages (and disadvantages!) of the new methods. The final decision will depend on the application itself, the actual signal, and its bandwidth.

To design and understand wavelets we still use Fourier techniques — the connections between time and frequency. This idea remains at the center of signal processing. Wavelets are "alternatives" rather than replacements. The classical transforms will survive very well. But other ideas have come quickly forward, to be understood and applied.

In a word, the new transforms are much more *local*. An event stays connected to the time when it occurs. Instead of transforming a pure "time description" into a pure "frequency description", the new methods find a good compromise — a *time-frequency description*. This is like a musical score, with specified frequencies at specified times. Remember that the Heisenberg uncertainty principle stands in the way of perfection! We lose accuracy in time when we gain accuracy in frequency. The musician cannot and would not change frequencies instantly. But our eyes and ears succeed to give location as well as frequency, and the new wavelet transforms have the same purpose.

The extreme case is an instantaneous impulse, with all frequencies in equal amounts. Its Fourier transform has constant magnitude over the whole spectrum. By contrast, a wavelet transform will involve only a small fraction of the wavelets — those that overlap the impulse. Figure 0.1 shows a sum of two extreme cases — an impulse and a pure wave $2\cos\omega n$. The Fourier transform spreads the impulse while it concentrates the wave (at frequencies ω and $-\omega$ of $e^{i\omega n} + e^{-i\omega n}$). The wavelet transform in the third figure is large at the *time* of the impulse and also large at the *frequency* of the wave.

Purpose of the Book

There are already good books on this subject. The ideas and applications are beautiful, and the word has spread. The bibliography lists many of those books, which have special strengths. Our purpose is different. We believe that a *textbook* is needed. Our text explains filter banks and wavelets from the beginning — in several ways and at least two languages. The examples and exercises come from our courses at M.I.T. and Wisconsin, which brought students from all over engineering and science.

Whether you are working individually or in class, we hope this book clarifies the central ideas. Implicit in that goal is the recognition that we cannot describe every filter and prove every theorem. The book concentrates on the underpinning of the subject, which is now stable. There is a special "*glossary*" to organize and define the terms that are constantly used, some from signal processing and others from mathematics. The central idea is a *perfect reconstruction filter bank*, with properties and purposes selected from this list:

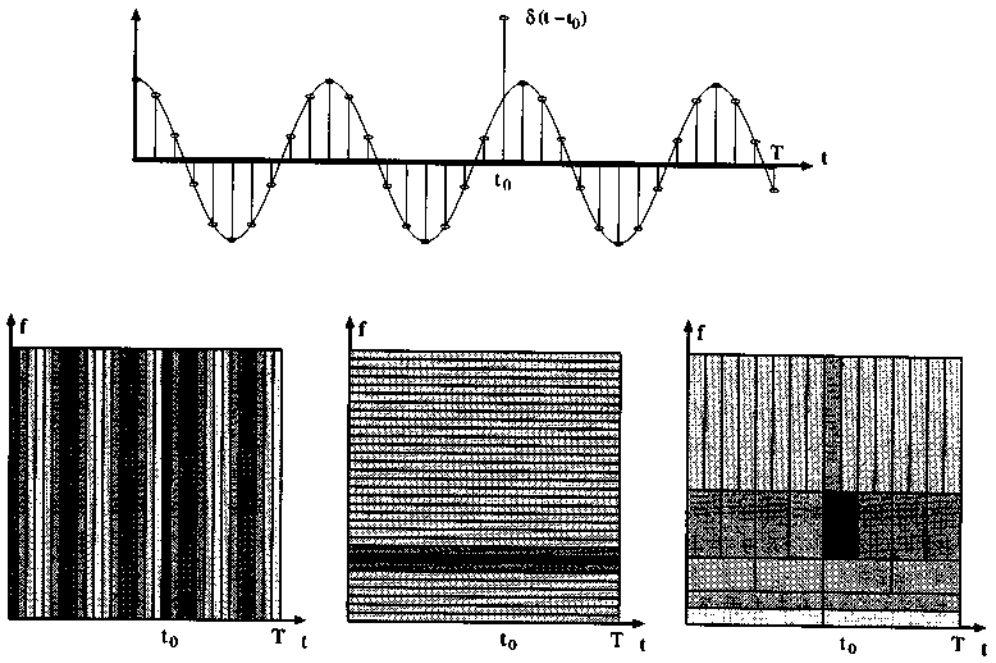


Figure 0.1: Impulse plus sinusoid, in time and frequency and time-frequency(time-scale).

Properties: Orthogonality, symmetry, short length, good attenuation.

Purposes: Audio/video compression, echo cancellation, radar, image analysis, communications, medical imaging, ...

Design and analysis techniques: Time domain, frequency domain, z-transform.

If the reader is willing, we would like to develop specific examples in this preface. Traditionally, the opening pages thank those who helped to create the book. Our debt to friends will soon be very gratefully acknowledged. But first, we go directly to transforms.

Transforms

Start with the basic idea and its purpose. The transform of a signal (a vector) is a *new representation* of that signal. The components $x(0), x(1), x(2), x(3)$ of a four-dimensional signal are replaced by four other numbers. Those numbers $y(0), y(1), y(2), y(3)$ are combinations of the x's. Our transforms are linear, so these are *linear combinations* — for example sums and differences:

$$\begin{array}{lllllll} y(0) & = & x(0)+x(1) & \qquad & y(2) & = & x(2)+x(3) \\ y(1) & = & x(0)-x(1) & \qquad & y(3) & = & x(2)-x(3). \end{array}$$

What is the purpose of the y's? And can we get back to the x's? The second question is easier and we answer it first.

This transform can be inverted. If you add the equations for $y(0)$ and $y(1)$, you get $2x(0)$. Subtracting those equations yields $2x(1)$. The ***inverse transform*** uses addition and subtraction (like the original!), and then division by 2:

$$\begin{array}{lllllll} x(0) & = & 0.5\,(y(0)+y(1)) & \qquad & x(2) & = & 0.5\,(y(2)+y(3)) \\ x(1) & = & 0.5\,(y(0)-y(1)) & \qquad & x(3) & = & 0.5\,(y(2)-y(3))\,. \end{array}$$

The y's allow perfect reconstruction of the x's, by the inverse transform. Looking ahead for a brief moment, we divide transforms into three groups:

(a)	**Lossless (orthogonal) transforms**	(orthogonal and unitary matrices)
(b)	**Invertible (biorthogonal) transforms**	(invertible matrices)
(c)	**Lossy transforms**	(not invertible)

A lossless unitary transform is like a rotation. The transformed signal has the same length as the original. This is true of the Fourier transform and all its real versions (DCT = discrete cosine transform, DST = discrete sine transform, HT = Hartley transform). The same signal is measured along new perpendicular axes.

For biorthogonal transforms, lengths and angles may change. The new axes are not necessarily perpendicular, but no information is lost. Perfect reconstruction is still available. We just invert. *Orthogonal wavelets give orthogonal matrices and unitary transforms, biorthogonal wavelets give invertible matrices and perfect reconstruction.* These transforms don't remove any information (or any noise), they just move it around — aiming to separate out the noise and decorrelate the signal. The irreversible step is to destroy small components, as we do below in "compression." Then invertibility is lost.

Matrices will appear very early, and we make no apology. A matrix displays the details of the transform. (The key texts in signal processing are amazingly empty of matrices. With the importance of systems like MATLAB, this must change.) Our book maintains a time-domain description by matrices, in parallel with the frequency-domain description by functions of ω. When the inputs $x(n)$ and outputs $y(n)$ are seen as vectors, the transform from x to y is executed by a matrix:

$$\begin{bmatrix} y(0) \\ y(1) \\ y(2) \\ y(3) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \end{bmatrix}.$$

The sum $x(0)+x(1)$ comes from the first row. The difference $x(0)-x(1)$ comes from the second row. Readers may recognize this as the 2-point DFT, and now comes its inverse. The matrix that recovers the x's from the y's is changed only by the factor $\frac{1}{2}$:

$$\begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} y(0) \\ y(1) \\ y(2) \\ y(3) \end{bmatrix}.$$

If you multiply these matrices, the result is the 4×4 identity matrix.

Small note When the first matrix is divided by $\sqrt{2}$, *it becomes an orthogonal matrix.* The rows of length $\sqrt{1+1} = \sqrt{2}$ will become unit vectors. The inner products between rows remain at zero. The transform becomes unitary (lossless) when the perpendicular axes — the four rows — are correctly normalized. ***The inverse is the transpose.***

The length squared of the transform is now

$$\left(\frac{x(0)+x(1)}{\sqrt{2}}\right)^2 + \left(\frac{x(0)-x(1)}{\sqrt{2}}\right)^2 + \left(\frac{x(2)+x(3)}{\sqrt{2}}\right)^2 + \left(\frac{x(2)-x(3)}{\sqrt{2}}\right)^2.$$

This simplifies to $x(0)^2+x(1)^2+x(2)^2+x(3)^2$. Thus $\|y\|^2 = \|x\|^2$ and the transform is unitary. For real matrices we also use the word orthogonal.

Purpose of the Transform

One important purpose is to see patterns in the y's that are not so clear in the x's. This means that the computer should see the pattern. What it sees best is *large versus small*. The computer will notice nothing special about the four numbers

$$x(0) = 1.2 \quad x(1) = 1.0 \quad x(2) = -1.0 \quad x(3) = -1.2.$$

Your eye notices various things in Figure 0.2 (I hope). Maybe the small movement and then the jump. Maybe the antisymmetry. To see this signal in the y representation, compute sums and differences:

$$y(0) = 2.2 \quad y(1) = 0.2 \quad y(2) = -2.2 \quad y(3) = 0.2.$$

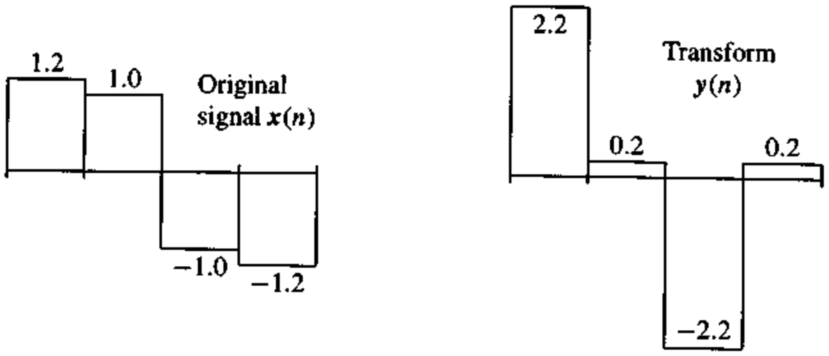


Figure 0.2: The x's transform to the y's by sums and differences.

First point: $y(1)$ and $y(3)$ are much smaller than $y(0)$ and $y(2)$. The differences are an order of magnitude smaller than the sums. Our transform has *almost lined up the signal* in the $y(0)$–$y(2)$ plane. Those two components of y do most of the work of four. It is true that we need all four y's to reconstruct perfectly all four x's. But if we change the small numbers $y(1)$ and $y(3)$ — just cancel them! — the compressed signal y_c is

$$y_c(0) = 2.2 \quad y_c(1) = 0 \quad y_c(2) = -2.2 \quad y_c(3) = 0.$$

Those numbers 0.2 were below our threshold. In the compressed y_c they are gone. Figure 0.3 shows the signal x_c reconstructed from y_c — the small difference between $x(0)$ and $x(1)$ is lost.

For comparison we compute the 4-point DFT, expecting and seeing the imaginary number $i = j = \sqrt{-1}$:

$$\begin{bmatrix} \widehat{x}(0) \\ \widehat{x}(1) \\ \widehat{x}(2) \\ \widehat{x}(3) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & i & i^2 & i^3 \\ 1 & i^2 & i^4 & i^6 \\ 1 & i^3 & i^6 & i^9 \end{bmatrix} \begin{bmatrix} 1.2 \\ 1.0 \\ -1.0 \\ -1.2 \end{bmatrix} = \begin{bmatrix} 0 \\ 2.2(1+i) \\ 0.4 \\ 2.2(1-i) \end{bmatrix}.$$

That zero in $\widehat{x}$ reflects the antisymmetry. To see it in the sum-difference transform, we must go to the next level.

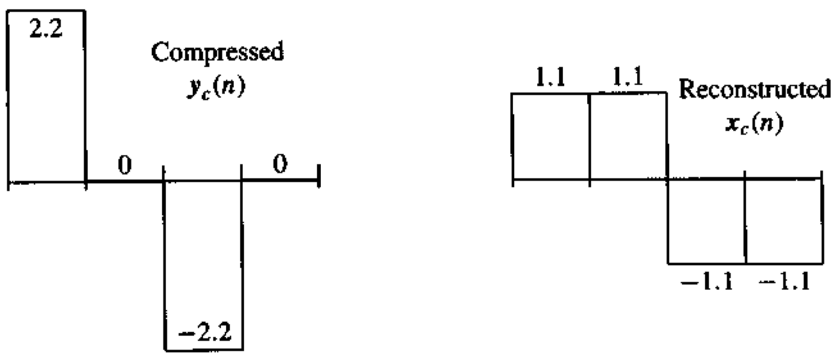


Figure 0.3: y_c and x_c are close to y and x, when small becomes zero.

Multilevel Transforms by Recursion

A key idea for wavelets is the concept of "*scale.*" Sums and differences of neighbors are at the finest scale. We move to a larger picture by taking sums and differences again. This is recursion—the same transform at a new scale. It leads to a ***multiresolution*** of the original signal $x(0), x(1), x(2), x(3)$. Averages and details will appear at different scales.

The wavelet formulation keeps the differences $y(1)$ and $y(3)$ at the finest level, and *iterates only on* $y(0)$ *and* $y(2)$. Iteration means sum and difference of the transform:

$$z(0) = y(0) + y(2) = 0 \quad \text{and} \quad z(2) = y(0) - y(2) = 4.4.$$

At this level there is extra compression. *One component* $z(2)$ now does most of the work of the original four. The wavelet transform $z(0), y(1), z(2), y(3)$ is in Figure 0.4.

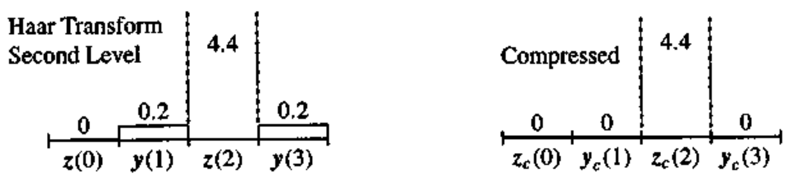


Figure 0.4: $z(0)$ and $z(2)$ are the sum and difference of $y(0)$ and $y(2)$. Compress to $z(2)$.

The key point is the multilevel construction, which is clear in a flow-graph:

$$\begin{tikzcd}
 & & \text{sum } z(0) \\
 & \text{sums } y(0), y(2) \arrow[ur] \arrow[dr] & \\
x \arrow[ur] \arrow[dr] & & \text{difference } z(2) \\
 & \text{differences } y(1), y(3) &
\end{tikzcd}$$

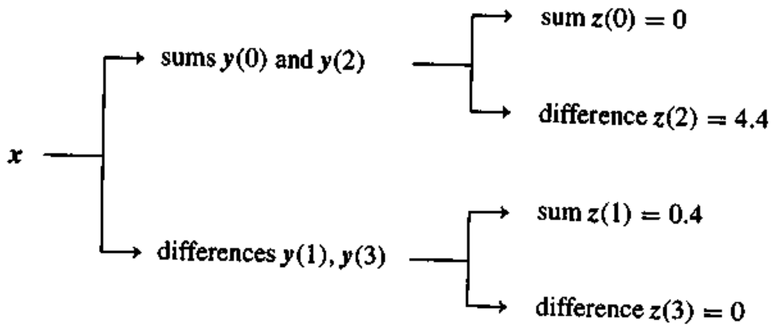
This two-step wavelet transform is executed by a product of two matrices:

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

That is invertible! To draw the flow-graph of the inverse, just reverse the arrows. To invert the matrix on the right, multiply the inverses of the matrices on the left (in reverse order of course).

The wavelet transform becomes unitary as before, when we divide sums and differences by $\sqrt{2}$. You will see this number appearing throughout the book, to compensate for scale changes.

Perhaps one more transform could be mentioned. It is the Walsh-Hadamard transform, which iterates *also on the differences* $y(1)$ *and* $y(3)$. Instead of the "logarithmic tree" for wavelets, we have a complete binary tree for Walsh-Hadamard:



This also counts as a *wavelet packet* (those include every binary tree). You would not want to miss the "Hadamard matrix" for this transform, which is exceptional. All entries are 1 and -1 and all rows are orthogonal:

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

When divided twice by $\sqrt{2}$ this matrix is orthogonal and also symmetric. The original x's can be reconstructed from the y's or the w's or the wavelet outputs $y(1), y(3), z(0), z(2)$. From compressed outputs we get approximate (but good) reconstruction.

Don't forget the disadvantage compared to wavelets! All sixteen entries are nonzero. The complete tree takes more computation than the wavelet tree. For signals of length n (a power of two), the Hadamard matrix has n^2 nonzeros and the wavelet matrix has only $n + n \log_2 n$. Those are the costs *without recursion*, jumping from the original x's directly to the final z's.

When the computations are recursive, as they always should be, we have a product of very sparse matrices. The flow-graph gives $2n \log_2 n$ for the complete Walsh-Hadamard transform (*exactly matching the Fast Fourier Transform*). This is good but wavelets are slightly better. The transform to $z(0), y(1), z(2), y(3)$ computes the sum and difference of $n - 1$ pairs. This needs only $2n - 2$ calls to memory.

The wavelet transform achieves the Holy Grail of complexity theory (or simplicity theory). *The transform is an* $O(n)$ *computation.* But does it separate the true signal from noise? Does it allow compression of 4:1 or 8:1 or 20:1? This sum-difference transform is analyzed further in Chapter 1 — it is such a good example — but we admit here that it is too simple. Better filters are needed, and they lead to better wavelets (still with $O(n)$ operations). That is the subject of our book.

Applications

The choice of waves or wavelets, Fourier analysis or filter analysis, depends on the signal. It must. Signals coming from different applications have different characteristics. It is helpful to see a broad picture:

Audio: Use many subband filters or a windowed (short time) Fourier transform.

Speech: The time variation is irregular and requires nonuniform intervals.

Images: Finite length signals need special treatment to reduce blocking. Symmetric extension goes with symmetric filters.

Video: Use motion prediction (optical flow of images) or space-time filtering.

The best choice for medical imaging is not clear. The legal questions arising from lossy compression are not clear either. Identification from synthetic aperture radar (SAR images) is an enormous problem. So is de-noising, which is at the heart of signal representation. Ultimately we are trying to choose a good basis.

The problem is to represent typical signals *with a small number of conveniently computable functions*. The traditional bases (Fourier, Bessel, Legendre, ...) come from differential equations. Wavelets do not come from differential equations! One reason is, those equations don't include dilation. *A dilation equation* $\phi(t) = \sum 2h(k)\phi(2t-k)$ *involves two time scales*. Its solution $\phi(t)$ is nonzero only on a finite interval. Then $\phi(2t)$ is nonzero on half of that interval. The basis is localized. It is quickly (and recursively) computed from the numbers $h(k)$. Those are the coefficients in the corresponding lowpass filter.

This relation of filters to functions is the heart of the book.

Acknowledgments

Many friends have created this theory. A lot of them also helped us to write about it. The bibliography points to their original work (we wish it were possible to be complete — that is better attempted electronically). Our personal thanks must begin with Vasily Strela and Chris Heil. They read most of the words and discussed all the ideas. Their unselfish help is deeply appreciated.

Other friends will see, at specific places in the book, exactly where their ideas and suggestions made a difference. We are enormously grateful to Kevin Amaratunga, Ross Barmish, Christopher Brislawn, Charles Chui, Albert Cohen, Adriannus Djohan, Dave Donoho, Jeff Geronimo, Doug Hardin, Peter Heller, Tom Hopper, Angela Kunoth, Seng Luan Lee, Michael Lightstone, Eric Majani, Stéphane Mallat, Henrique Malvar, Ricardo de Queiroz, Jianhong Shen, Wim Sweldens, Pankaj Topiwala, Steffen Trautmann, Chi Wah Kok, Victor Wickerhauser, and Zhifeng Zhang. There is no way to thank you enough for such generous help.

We want to recognize separately the outstanding leadership of four authors, P. P. Vaidyanathan and Martin Vetterli in signal processing and Ingrid Daubechies and Yves Meyer in mathematics. The two subjects are thoroughly mixed, thanks to these four! Their work has been a model in every way, personal (to us) as well as scientific (for all).

Our writing of the book was shared in a natural way — words mostly from the first author, filter designs and applications from the second author, ideas from both. Vasily Strela and Robert Becker helped to get Versions 1.0 to 9.9 into TEX, and Martin Stock made it more beautiful. The whole project grew out of our courses and workshops (which will continue). Those were tremendous sources of inspiration, thanks to the participants.

This subject is not just theory. The ideas have to be implemented, and MATLAB is the outstanding way to do that. It is a pleasure to have this book closely linked to the *Wavelet Toolbox* offered by MathWorks. Exercises that use this Toolbox are in the book; the reader can *see* the wavelets and the multiresolution they produce. A more extensive Wavelet Manual will come from the same sources: Wellesley-Cambridge Press and MathWorks.

For correspondence about this whole subject, and for comments and corrections of any kind, the authors thank the readers. Especially we thank Jill and Thuy Duong for their patience and support. It is finished at last! We hope you enjoy this book.

Gilbert Strang and Truong Nguyen

M.I.T. and Wisconsin, November 1995

`gs@math.mit.edu` and `nguyen@ece.wisc.edu`

`http://saigon.ece.wisc.edu/~waveweb/Tutorials/book.html`

Guide to the Book

This book has a two-part subject. One part is discrete, the other is continuous. In discrete time we develop the idea and applications of ***filter banks***. In continuous time we have ***scaling functions*** $\phi(t)$ and ***wavelets*** $w(t)$. By a natural limiting process, iteration of the lowpass filter leads to the scaling function. One highpass filter then produces a wavelet. Our goal is to make this connection clear. We find the conditions on the discrete coefficients that lead to good filter banks and good wavelets.

Historically and mathematically, the filters come first. Perfect reconstruction filter banks were developed in the early 1980's. The excitement around wavelets started later (and grew quickly). This excitement was not universal — designers of filter banks naturally asked what was new. Part of the answer is precisely in that process of ***iteration***. For a filter to behave well in practice, when it is combined with subsampling and repeated five times, it must have an extra property — not built into earlier designs. This property expresses itself in the frequency domain by a sufficient number of "zeros at π". Then the frequency band can be successfully separated into five octaves.

The underlying problem is to choose a good basis. We want to represent a signal well, by a small number of basic signals. These can be sinusoids and they can be wavelets. On a discrete grid, $\omega = \pi$ is the highest frequency at which a signal can oscillate. Those oscillations $x(n) = e^{i\pi n} = (-1)^n$ are stopped by the lowpass filter with a "zero at π". The highpass filter lets fast oscillations through, and the synthesis filters can reconstruct the exact input. But ***compression*** may come between analysis and synthesis. Frequencies that are barely represented will be intentionally lost. That mostly means high frequencies but the filter bank is impartial — it keeps the basis functions that are important to the specific signal. We want to show when, and why, filter banks and wavelets are effective in reconstruction and signal representation and compression.

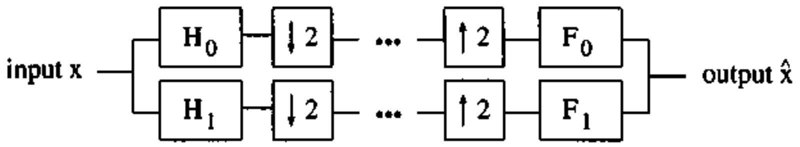
Filter Banks

Some readers will begin this book with Chapter 1. Others will jump forward to a topic of particular interest. This brief guide is for both, especially to tell the first readers what is coming and the second group where to look. We are pointing to places where preparation and explanation come together, to design and study new structures.

For filter banks, that place is Section 4.1. There we identify the two conditions for perfect reconstruction (in the absence of lossy compression). One condition removes distortion, the other condition removes aliasing. The anti-distortion condition applies to the products F_0H_0 and F_1H_1 along the channels of the filter bank. Then the anti-aliasing condition controls how those products can be separated into the four filters.

The design of a perfect reconstruction filter bank is a choice of F_0H_0 and then a factorization. To understand the conditions on distortion and aliasing, we apply the techniques of multirate filtering. Those techniques are explained in Chapters 1–3, with examples throughout. Of course filters are to be defined! But we can go forward even now, to illustrate a filter bank that gives perfect reconstruction.

The analysis bank is on the left. It has a lowpass filter $\boldsymbol{H}_0$, and a highpass filter $\boldsymbol{H}_1$, and decimation by $(\downarrow 2)$ — which removes the odd-numbered components after filtering. The analysis bank yields two "half-length" outputs. Then the synthesis bank on the right begins with the upsampling operation $(\uparrow 2)$ — which inserts zeros in those odd components:

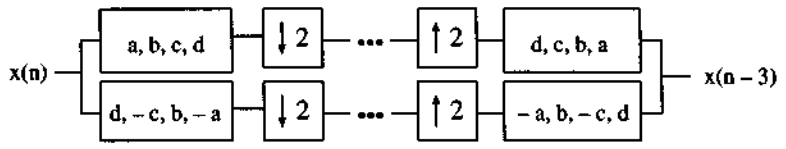


Two-channel filter bank: Separate the input into frequency bands (filter and downsample). Then reassemble (upsample and filter).

The gap in the center indicates where the subband signals are compressed or enhanced. The applications of this structure are extremely widespread. We believe that any reader interested in signal processing (and image processing) will find that filter bank analysis is extremely useful.

The filters $\boldsymbol{H}_0, \boldsymbol{H}_1, \boldsymbol{F}_0$, and $\boldsymbol{F}_1$ are linear and time-invariant. The operators $(\downarrow 2)$ and $(\uparrow 2)$ are ***not*** time-invariant. These multirate operations are responsible for *aliasing* and for *imaging* — they create undesirable and extraneous signals that the filters must cancel. To understand how that happens, and to design good filters, we use the tools developed in Chapters 1–3: especially transformation to the frequency domain and z-domain. We try to explain the analysis of multirate filtering, with $(\downarrow 2)$ and $(\uparrow 2)$, clearly and memorably.

The theory and the design of filter banks and wavelets will dominate Chapters 4–6. This is the heart of the book. The structure of an ***orthogonal*** bank is very special, and the next figure shows how the filters are related. For length 4 all filters use the four coefficients a, b, c, d that Daubechies derived:



The form of an orthogonal filter bank with four coefficients.

How did she choose a, b, c, d? Part of the answer will have to wait, but here is the essential idea. The product along the top channel gives a particular "halfband filter":

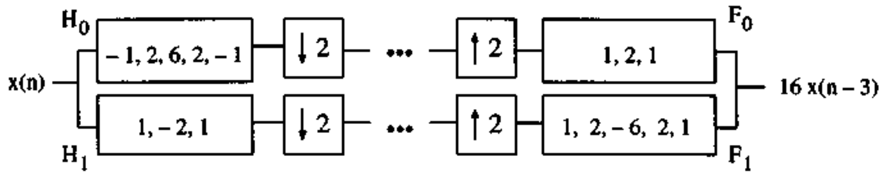
$$(a,\ b,\ c,\ d) * (d,\ c,\ b,\ a) = (-1,\ 0,\ 9,\ 16,\ 9,\ 0,\ -1)/16.$$

This convolution is a multiplication of two polynomials, when $a,\ b,\ c,\ d$ are the coefficients:

$$(a + bz^{-1} + cz^{-2} + dz^{-3})\,(d + cz^{-1} + bz^{-2} + az^{-3}) =$$
$$(-1 + 9z^{-2} + 16z^{-3} + 9z^{-4} - z^{-6})/16.$$

The four coefficients are pleasant to calculate. The serious job in Chapter 4 is to explain what is special about that 6th-degree polynomial in which z^{-1} and z^{-5} are missing.

A filter bank also gives perfect reconstruction if it is ***biorthogonal***. This design is less restricted. The product F_0H_0 must skip the same odd powers of z^{-1}, but F_0 does not have to be the transpose (the flip) of H_0. Here are specific numbers for the filter coefficients — not the only choice and maybe not the best. They show how the filters F_0 and F_1 on the synthesis side are related to the analysis filters H_1 and H_0 (by alternating signs):



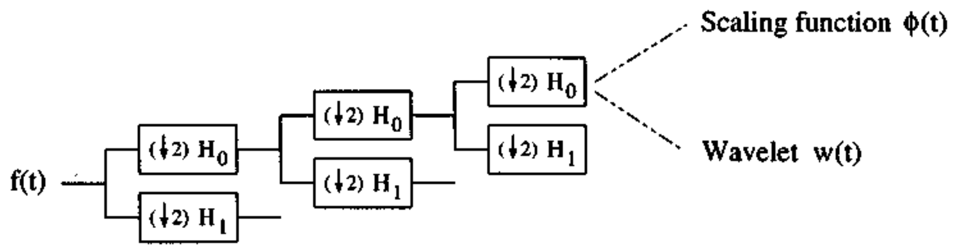
A biorthogonal filter bank: Perfect reconstruction with 3 delays.

For filters, we stop here. This time $F_0(z) = 1 + 2z^{-1} + z^{-2}$. Multiplied by $H_0(z)$ it gives the same important 6th-degree polynomial as before. To understand why the zero coefficients are necessary in that polynomial, and why $-\frac{1}{16}$ and $\frac{9}{16}$ are desirable, I am afraid that you have to read the book!

Our discussion went this far (further than we intended) so as to make a basic and encouraging point: *The construction of new filter banks need not be complicated.* This subject is accessible to new ideas and experiments.

Wavelets

Wavelets are localized waves. Instead of oscillating forever, they drop to zero. They come from the ***iteration*** of filters (with rescaling). The link between discrete-time filters and continuous-time wavelets is in the limit of a logarithmic filter tree:



Scaling function and wavelets from iteration of the lowpass filter.

Scaling functions and wavelets have remarkable properties. They inherit orthogonality, or biorthogonality, from the filter bank. Because of the repeated rescaling that produces them, wavelets decompose a signal into details at all scales. The wavelet $w(t)$ and its shifts $w(t-k)$ are at unit scale. The wavelets $w(2^jt)$ and $w(2^jt-k)$ are at scale 2^{-j}. The biorthogonal functions $\tilde{\phi}(t)$ and $\tilde{w}(t)$ come from iterating the synthesis bank.

Wavelets produce a natural ***multiresolution*** of every image, including the all-important edges. Where the low frequency part of the Fourier transform is often a blur, the output from the lowpass channel is a useful compression.

Sections 5.5 and 6.2 study the particular wavelets created by Ingrid Daubechies. They are orthogonal, with the advantages and limitations that this property brings. Sections 4.1 and 6.5 study biorthogonal alternatives, which come from different factorizations of the same polynomial (as above). This polynomial corresponds to a "maxflat halfband filter", and we hope you will like the connections.

More than that, we hope you enjoy the whole book. This subject is a beautiful combination of mathematical analysis and signal processing applications. The analysis and the applications are based on designs that give perfect reconstruction. To explain both sides of this subject, we need words from mathematics and words from digital signal processing. The Glossary at the end is a dictionary of their meanings. Above all we need ***ideas*** from both sides, and from a tremendous range of application areas. It is to the understanding of filters and wavelets, and the growth of successful applications, that this book is dedicated.

Summary of the Theory

There are four conditions that play a central part in this book. Because of their importance we highlight them here. They apply directly to the coefficients in the filter banks — and the consequences are felt (after iteration!) in the scaling functions and wavelets. Here are the four conditions — some might say in decreasing order of importance:

PR Condition	*Perfect reconstruction.* The synthesis bank inverts the analysis bank, with ℓ delays. Biorthogonal banks with no aliasing and no distortion.
Condition O	*Orthogonality.* The analysis bank is inverted by its transpose. The wavelets are orthogonal to all their dilates and translates.
Condition A_p	*Accuracy of order p for approximation by scaling functions $\phi(t-k)$.* p vanishing moments in the wavelets. pth order decay of wavelet coefficients for smooth $f(t)$.
Condition E	*Eigenvalue condition on the cascade algorithm.* Determines convergence to $\phi(t)$ and smoothness of wavelets. Equivalent to stability of the wavelet basis.

One step further and this Guide is ended. The four fundamental conditions will be stated explicitly for a two-channel filter bank, with the sections in which they appear. We continue to use the polynomials $H_0(z)$, $H_1(z)$, $F_0(z)$, and $F_1(z)$, whose coefficients come directly from the filters. By convention, these are polynomials in z^{-1}, and the lowpass analysis filter is represented by $H_0(z) = \boldsymbol{h}(0) + \boldsymbol{h}(1)z^{-1} + \cdots + \boldsymbol{h}(N)z^{-N}$. Here are the conditions that give filters and wavelets with good properties:

1. Perfect Reconstruction (***PR condition in Section 4.1***)

$$F_0(z)H_0(z) + F_1(z)H_1(z) = 2z^{-\ell} \quad \text{and} \quad F_0(z)H_0(-z) + F_1(z)H_1(-z) = 0.$$

The second equation gives the anti-aliasing choices $F_0(z) = H_1(-z)$ and $F_1(z) = -H_0(-z)$.

2. Orthogonality (***Condition O in Section 5.3***)
The filter coefficients are reversed by $F_0(z) = z^{-N} H_0(z^{-1})$ and $F_1(z) = z^{-N} H_1(z^{-1})$. Then perfect reconstruction depends on the "double-shift orthogonality" of the lowpass coefficients $\boldsymbol{h}(k)$:

$$\sum \boldsymbol{h}(k)\,\boldsymbol{h}(k+2n) = \delta(n).$$

In terms of the polynomials this is $H_0(z)H_0(z^{-1}) + H_0(-z)H_0(-z^{-1}) = 2$.

3. Accuracy of order p (***Condition*** $\boldsymbol{A_p}$ ***in Sections 5.5 and 7.1***)
The lowpass filter has a zero of order p at $z = -1$:

$$H_0(z) = \left(\frac{1+z^{-1}}{2}\right)^p Q(z).$$

4. Convergence and Stability (***Condition E in Section 7.2***)
The transition matrix $\boldsymbol{T}$ has $\lambda = 1$ as simple eigenvalue and all other $|\lambda(\boldsymbol{T})| < 1$.

Final note: The sixth degree polynomial in the examples above has *four zeros* at $z = -1$:

$$-1 + 9z^{-2} + 16z^{-3} + 9z^{-4} - z^{-6} = (1 + z^{-1})^4\,(-1 + 4z^{-1} - z^{-2}).$$

These zeros give flat responses near $\omega = \pi$ and also $\omega = 0$. The absence of z^{-1} and z^{-5} is the key to perfect reconstruction. Polynomials of higher degree, also with zeros at $z = -1$ and also with only one odd power, factor into $F_0(z)H_0(z)$ to give the best filters for iteration. In the limit of the iterations, these filters give good wavelets.

Filters and Matrices

- $h = \big(h(0),\ \ldots,\ h(N)\big)$ Causal FIR lowpass filter (impulse response) with $\sum h(k) = 1$
- H Toeplitz matrix with $h(k)$ on the kth diagonal: $H_{ij} = h(i-j)$
- $H(\omega) = \sum h(k)\, e^{-ik\omega}$ (frequency response in reduced notation)
  $H(e^{j\omega}) = \sum h(k)\, e^{-jk\omega}$ (signal processing notation)
- $H(z) = \sum h(k) z^{-k}$ Transfer function in the z-domain with $z = e^{j\omega}$
- $y = Hx = h * x$ corresponds to $Y(z) = H(z)X(z)$ ***Convolution Rule***
- S Delay gives $(Sh)(k) = h(k-1)$ ***Time Invariance is $SH = HS$***
- S^{-1} Advance gives $(S^{-1}h)(k) = h(k+1)$ Multiply transform by $z = e^{-j\omega}$
- $(\downarrow 2)$ Downsampling operator $(\downarrow 2)h = h_{\text{even}} = \big(h(0),\ h(2),\ h(4),\ \ldots\big)$ = even phase of h
- $(\downarrow 2)S^{-1}h = h_{\text{odd}} = \big(h(1),\ h(3),\ h(5),\ \ldots\big)$ Odd phase of h
- $H_p(z) = [H_{\text{even}}(z)\ \ H_{\text{odd}}(z)] = \big[\sum h(2k)z^{-k}\ \ \ \sum h(2k+1)z^{-k}\big]$ Polyphase representation
- $(\uparrow 2)$ Upsampling operator $(\uparrow 2)v = [v(0)\ \ 0\ \ v(1)\ \ 0\ \ v(2)\ \ 0\ \ \cdots]$ has z-transform $V(z^2)$
- $(\uparrow 2)(\downarrow 2)x = [x(0)\ \ 0\ \ x(2)\ \ 0\ \ x(4)\ \ 0]$ has z-transform $\frac{1}{2}[X(z) + X(-z)] = X_{\text{even}}(z^2)$
- $H(z^{-1})$ Transpose filter with coefficients $h(-k)$: anticausal matrix H^T
- $z^{-N}H(z^{-1})$ Flip to $h(N-k)$ produces $(h(N),\ \ldots,\ h(2),\ h(1),\ h(0))$
- $H(-z)$ Alternating sign $(-1)^k h(k)$ produces $\big(h(0),\ -h(1),\ h(2),\ -h(3),\ \ldots\big)$
- $z^{-N}H(-z^{-1})$ Alternating flip $(-1)^k h(N-k)$ produces $\big(h(N),\ \ldots\ ,(-1)^N h(0)\big)$
- $h(k) = h(N-k)$ Symmetric filter (**W** symmetry for odd N and **H** symmetry for even N)

Filter Banks

- $\boldsymbol{L} = (\downarrow 2)\boldsymbol{C} = (\downarrow 2)\sqrt{2}\boldsymbol{H}_0$ Lowpass analysis channel: Filter and downsample
- $\boldsymbol{B} = (\downarrow 2)\boldsymbol{D} = (\downarrow 2)\sqrt{2}\boldsymbol{H}_1$ Highpass analysis channel: Filter has $\sum \boldsymbol{h}_1(k) = 0$
- $\sqrt{2}\boldsymbol{F}_0(\uparrow 2)$ Lowpass synthesis channel: Upsample and filter with $F_0(z) = H_1(-z)$
- $\sqrt{2}\boldsymbol{F}_1(\uparrow 2)$ Highpass synthesis channel: Upsample and filter with $F_1(z) = -H_0(-z)$
- $F_0(z)H_0(z) - F_0(-z)H_0(-z) = P_0(z) - P_0(-z) = 2z^{-\ell}$ No distortion: ℓ delays *(odd ℓ)*
- $\boldsymbol{H}_p(z) = \begin{bmatrix} H_{0,\text{even}}(z) & H_{0,\text{odd}}(z) \\ H_{1,\text{even}}(z) & H_{1,\text{odd}}(z) \end{bmatrix}$ and $\boldsymbol{F}_p(z) = \begin{bmatrix} F_{0,\text{even}}(z) & F_{1,\text{even}}(z) \\ F_{0,\text{odd}}(z) & F_{1,\text{odd}}(z) \end{bmatrix}$ Polyphase matrices
- $\boldsymbol{H}_m(z) = \begin{bmatrix} H_0(z) & H_0(-z) \\ H_1(z) & H_1(-z) \end{bmatrix}$ and $\boldsymbol{F}_m = \begin{bmatrix} F_0(z) & F_1(z) \\ F_0(-z) & F_1(-z) \end{bmatrix}$ Modulation matrices
- $\boldsymbol{F}_p(z)\boldsymbol{H}_p(z) = \boldsymbol{I}$ Perfect Reconstruction with no delays
- $\boldsymbol{F}_m(z)\boldsymbol{H}_m(z) = \begin{bmatrix} 2z^{-\ell} & 0 \\ 0 & -2z^{-\ell} \end{bmatrix}$ Perfect Reconstruction with ℓ delays *(odd ℓ)*
- $\boldsymbol{H}_p^T(z^{-1})\boldsymbol{H}_p(z) = \boldsymbol{I}$ $\boldsymbol{H}_p(z)$ is paraunitary and the filter bank is orthogonal
- $\sum \boldsymbol{h}_0(n)\boldsymbol{h}_0(n+2k) = \delta(k)$ and $H_0(z)H_0(z^{-1}) + H_0(-z)H_0(-z^{-1}) = 2$ Orthogonal lowpass
- $\sum (-1)^k k^m \boldsymbol{h}_0(k) = 0$ for $0 \le m < p$ Condition A_p gives p zeros of $H_0(e^{j\omega})$ at $\omega = \pi$

List of Scaling Functions and Wavelets

- Haar (box function and up-down square wave)
- Daubechies (maxflat orthogonal with $2p$ filter coefficients)
- Splines and biorthogonal wavelets ($F(z) = (1 + z^{-1})^p$ gives spline of degree $p - 1$)
- Splines and semiorthogonal wavelets (perpendicular to splines, IIR duals)
- Maxflat biorthogonal ($F(z)H(z)$ is Daubechies maxflat product filter)
- Binary biorthogonal (coefficients are integers times 2^{-n}; Section 6.5)
- Shannon (ideal filter, sinc scaling function, box function in ω)
- Meyer (IIR but smooth in ω and t)
- Coifman (zero moments $H^{(k)}(0)$ and $\int t^k \phi(t)dt = 0$ for $0 < k < p$)
- Morlet ($e^{-iat}e^{-t^2/2}$ with $a = \pi\sqrt{2/ln2}$)

Multiresolution and Wavelets

- $\phi(t)$ from the dilation equation (refinement equation) $\phi(t) = \sum 2h(k)\phi(2t-k)$
- $w(t)$ from the wavelet equation $w(t) = \sum 2h_1(k)\phi(2t-k)$
- $w_{jk}(t) = 2^{j/2}w(2^jt-k)$: Normalized wavelet on $[k\Delta t, (k+N)\Delta t]$
- Scale parameter j for stepsize $\Delta t = 2^{-j}$ ($\Delta t = 2^j$ in [D] and MATLAB Toolbox)
- Shift-invariant subspaces $V_j \subset V_{j+1}$ with $V_j \oplus W_j = V_{j+1} : f(t) \in V_0 \leftrightarrow f(2^jt) \in V_j$
- $\{\phi(t-k)\}$ An orthonormal basis (or only a Riesz basis) for V_0
- $\{w(t-k)\}$ An orthonormal basis (or only a Riesz basis) for W_0
- $\left\{2^{j/2}\phi(2^jt-k)\right\}$ and $\left\{2^{j/2}w(2^jt-k)\right\}$ Bases for V_j and W_j. Joint basis for V_{j+1}
- Orthogonal multiresolution Orthonormal bases with V_j perpendicular to W_j
- Semiorthogonal multiresolution Riesz bases with V_j perpendicular to W_j
- Biorthogonal multiresolution Biorthogonal bases with $V_j \perp \widetilde{W}_j$ and $W_j \perp \widetilde{V}_j$
- $\widetilde{\phi}(t)$ and $\widetilde{w}(t)$ from the analysis filters $\boldsymbol{h}_0$ and $\boldsymbol{h}_1$ generate $\widetilde{V}_0 \oplus \widetilde{W}_0 = \widetilde{V}_1$
- $\phi(t)$ and $w(t)$ from the synthesis filters $\boldsymbol{f}_0$ and $\boldsymbol{f}_1$ generate $V_0 \oplus W_0 = V_1$
- $\langle \phi(t-k), \widetilde{\phi}(t-\ell)\rangle = \delta(k-l)$ Biorthogonal (dual) bases for V_0 and $\widetilde{V}_0$
- $\langle w(t-k), \widetilde{w}(t-\ell)\rangle = \delta(k-l)$ Biorthogonal (dual) bases for W_0 and $\widetilde{W}_0$
- $a_{jk} = \langle f(t), \widetilde{\phi}_{jk}(t)\rangle$ Coefficients in $f_j(t) = \sum_k a_{jk}\phi_{jk}(t)$ = projection of $f(t)$ onto V_j
- $b_{jk} = \langle f(t), \widetilde{w}_{jk}(t)\rangle$ Wavelet coefficients in $f(t) = \sum\sum b_{jk}w_{jk}(t)$
- $a_{jk} = \sum \boldsymbol{h}_0(\ell-2k)\, a_{j+1,\ell}$ and $b_{jk} = \sum \boldsymbol{h}_1(\ell-2k)\, a_{j+1,\ell}$ Mallat Fast Wavelet Transform
- $a_{j+1,\ell} = \sum \boldsymbol{f}_0(\ell-2k)\, a_{jk} + \sum \boldsymbol{f}_1(\ell-2k)\, b_{jk}$ Fast Inverse Wavelet Transform

Dilation Equation — Solution and Smoothness

- $\widehat{\phi}(\omega) = H(\frac{\omega}{2})\,\widehat{\phi}(\frac{\omega}{2})$ Fourier transform of the dilation equation
- $\widehat{\phi}(\omega) = \prod_{j=1}^{\infty} H(\frac{\omega}{2^j})$ Fourier transform of the scaling function $\phi(t)$
- $\phi^{(i+1)}(t) = \sum 2\boldsymbol{h}(k)\phi^{(i)}(2t-k)$ Cascade algorithm with $\phi^{(0)}(t) =$ box function
- $\boldsymbol{M} = (\downarrow 2)\; 2\,\boldsymbol{H}$ Cascade matrix with double-shifted rows: $\boldsymbol{M} = \sqrt{2}\boldsymbol{L}$
- $\boldsymbol{m}(0)$ and $\boldsymbol{m}(1)$ N by N submatrices of $\boldsymbol{M}$ Entries $2\boldsymbol{h}(2i-j)$ and $2\boldsymbol{h}(2i-j+1)$
- $\boldsymbol{m}(0)\Phi = \Phi$ Eigenvector with $\lambda = 1$ gives $\Phi = \big(\phi(0),\ \phi(1),\ \ldots\big)$ at the integers
- $\Phi(.\,t_1\,t_2\,\cdots) = \boldsymbol{m}(t_1)\Phi\,(.\,t_2\,t_3\,\cdots)$ Recursion for $\Phi(t) = \big(\phi(t),\ \phi(t+1),\ \ldots\big)$ at dyad
- $\boldsymbol{m}(t_1)\,\boldsymbol{m}(t_2)\,\cdots\,\boldsymbol{m}(t_k)$ All products uniformly bounded $\leftrightarrow$ Bounded recursion for $\phi(t)$
- $\boldsymbol{T} = (\downarrow 2)2\boldsymbol{H}\boldsymbol{H}^T$ Transition matrix from autocorrelation $\boldsymbol{h} * \boldsymbol{h}^T$ corresponding to $|H(\omega)|^2$
- $\boldsymbol{Ta} = \boldsymbol{a}$ Eigenvector of $\boldsymbol{T}$ gives the inner products $\boldsymbol{a}(k) = \langle\phi(t),\ \phi(t+k)\rangle$
- $A(z) = \sum \boldsymbol{a}(k)z^{-k}$ and $A(\omega) = \sum |\widehat{\phi}(\omega + 2\pi k)|^2$ Euler-Frobenius polynomial from $\boldsymbol{h}(k$
- $A(\omega) \equiv 1$ and $A(z) \equiv 1$ and $\boldsymbol{a} = \boldsymbol{\delta}$ Orthonormal basis $\{\phi(t-k)\}$
- $0 < A \le A(\omega) \le B$ and $A\sum a_k^2 \le \|\sum a_k\phi(t-k)\|^2 \le B\sum a_k^2 \leftrightarrow$ Riesz basis $\{\phi(t-k)\}$
- Condition E for Riesz bases $\{\phi(t-k)\}$ and $\{w_{jk}(t)\}$ and L^2 convergence of $\phi^{(i)}(t)$ to $\phi(t)$:

  $\lambda = 1$ is a simple eigenvalue of $\boldsymbol{T}$ and all other $|\lambda(\boldsymbol{T})| < 1$
- Special eigenvalues $\lambda = 1,\ \ldots,\ (\frac{1}{2})^{p-1}$ of $\boldsymbol{M}$ and $\lambda = 1,\ \ldots,\ (\frac{1}{2})^{2p-1}$ of $\boldsymbol{T}$ from p zeros
- $s_{\max} = -\frac{\log\rho}{\log 4}$ Smoothness of $\phi(t)$ with $\rho = |\lambda_{\max}(\boldsymbol{T})|$ excluding $\lambda = 1,\ \frac{1}{2},\ \ldots,\ (\frac{1}{2})^{2p-1}$
- $s_{\max} = p - \frac{1}{2}$ Smoothness in L^2 of splines of degree $p-1$ from $H(z) = \big(\frac{1+z^{-1}}{2}\big)^p$
- $s_{\max} \le p - \frac{1}{2}$ Bound on derivatives of $\phi(t)$ when the filter has p zeros at π