Multiwavelets: Theory and Applications

by

Vasily Strela

M.Sc., Moscow Institute of Physics and Technology

Submitted to the Department of Mathematics
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Mathematics

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 1996

Author . .....
Department of Mathematics
May 3, 1996

Certified by .....
Gilbert Strang
Professor of Mathematics
Thesis Supervisor

Accepted by .....
David A. Vogan
Chairman, Departmental Committee on Graduate Students

Multiwavelets: Theory and Applications

by

Vasily Strela

Submitted to the Department of Mathematics
on May 3, 1996, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Mathematics

Abstract

A function $\phi(t)$ is *refinable* if it satisfies a *dilation equation* $\phi(t) = \sum_k C_k \phi(2t-k)$. A refinable function (*scaling* function) generates a *multiresolution analysis* (MRA): Set of nested subspaces $\dots V_{-1} \subset V_0 \subset V_1 \dots$ such that $\overline{\bigcup_{j=-\infty}^{\infty} V_j} = L^2(\mathbf{R})$, $\bigcap_{j=-\infty}^{\infty} V_j = \{0\}$, and translates $\phi(t-k)$ constitute a basis of V_0. Then a basis $\{w_{jk} : w_{jk} = w(2^j t - k)\ j,k \in \mathbf{Z}\}$ of $L^2(\mathbf{R})$ is generated by a wavelet $w(t)$, whose translates $w(t-k)$ form a basis of W_0, $V_1 = V_0 \oplus W_0$.

A standard (*scalar*) MRA assumes that there is only one scaling function. We make a step forward and allow several scaling functions $\phi_0(t), \dots, \phi_{r-1}(t)$ to generate a basis of V_0. The vector $\boldsymbol{\phi}(t) = [\phi_0(t) \dots \phi_{r-1}(t)]^T$ satisfies a dilation equation with *matrix* coefficients $\boldsymbol{C}_k$. Associated with $\boldsymbol{\phi}(t)$ is a *multiwavelet* $\boldsymbol{w}(t) = [w_0(t) \dots w_{r-1}(t)]^T$. Unlike the scalar case, construction of a multiwavelet is a nontrivial design problem. We discuss the ways to solve it.

This work extends wavelet theory to the *vector* case. Multiwavelets are a natural generalization of scalar wavelets. They have new features arising from the matrix structure of the dilation equation. For example, a multi-scaling function can combine orthogonality, symmetry and high approximation order while a scalar function cannot.

The properties of an MRA can be formulated either in *time* or in *frequency* (*Fourier*) domain. We start with establishing conditions of orthogonality and approximation in the time domain. Also we find a way to construct multi-scaling functions and present the finite element example. This method requires solution of a large system of nonlinear equations. We therefore switch to the frequency domain.

Working in the Fourier domain one faces the necessity to deal with the *refinement mask* (*two-scale symbol*) $\boldsymbol{P}(\omega) = \sum_k e^{-ik\omega} \boldsymbol{C}_k$. In the scalar case $P(\omega)$ is a *trigonometric polynomial*. Its factorization $P(\omega) = (1+e^{-i\omega})^m P_0(\omega)$ defines regularity and approximation properties of the scaling function. In the vector case $\boldsymbol{P}(\omega)$ becomes a *matrix trigonometric polynomial*. Each factor $(1+e^{-i\omega})$ changes to a *two-scale similarity transform* (TST) which connects matrix polynomials $\boldsymbol{P}(\omega)$ and $\boldsymbol{Q}(\omega)$ by the relation $\boldsymbol{Q}(\omega) = \boldsymbol{M}(2\omega)\boldsymbol{P}(\omega)\boldsymbol{M}^{-1}(\omega)$. We study the properties of TST and show how it is connected with the approximation order, regularity, symmetry and orthogonality of the multi-scaling functions. TST appears to be a very useful tool in the whole multiwavelet theory. Moreover, it gives an efficient way of construction of multi-scaling functions with given properties. We present an algorithm for this procedure and examples obtained with its help.

Given a multi-scaling function we develop the *cofactor method* for the construction of dual scaling functions and biorthogonal multiwavelets. It turns out that this method extends approximation and symmetry from the analysis to the synthesis basis.

Finally, we apply multiwavelet filter banks in image compression and signal denoising. We develop preprocessing and symmetric extension techniques and construct some special multi-scaling functions. In many experiments multiwavelets perform better than the scalar ones. A further development of the theory and applications is needed.

Thesis Supervisor: Gilbert Strang
Title: Professor of Mathematics

Acknowledgments

There is no way I can thank everybody who helped me in my work, but this thesis would be just impossible without the support of my advisor, Professor Gilbert Strang. He carefully guided me through three and a half years at MIT, discussed all ideas in this thesis, read all its words and taught me a lot. Thank you very much!

I am also very grateful to my coauthors and friends Chris Heil, Peter Heller, Gerlind Plonka and Pankaj Topiwala. It was a great pleasure to work with them and I hope that our collaboration will continue. Communication with Professor Stephane Mallat was also very rewarding.

I am very thankful to the members of my thesis committee Professor Alan Edelman and Professor John Williams for their readiness to read this text and evaluate my work.

Thanks to Phyllis Block and Maureen Lynch who always were very supportive and helpful.

Many thanks to Professors Ingrid Daubechies, Yves Meyer and Truong Nguyen for their willingness to support my future.

Thanks to Professor Manfred Tasche and all friends in Rostock University for organizing my visit there.

Thanks to *all* my friends and especially to my office-mates Dr. Olga Kravchenko, M.D. Alia Zigangirova, Dr. Dmitri Kaledin and Dr. Marat Rovinsky who encouraged me a lot.

I greatly appreciate the financial support of MIT and AWARE Inc. which made my work and life possible.

Finally in this list, but on the first place in my heart, I would like to thank my parents and grandparents for their tremendous (although remote) emotional support, compassion, and useful advices.

It is finally finished, but I am sorry it is over!

To my parents and grandparents

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Brief History of the Subject

Wavelets are an interesting mathematical object and a useful tool in many applications such as image compression, signal denoising, problems of numerical analysis and physics.

The initial idea of wavelet theory is very beautiful and simple: Try to compose a basis of $L^2(\mathbf{R})$ from integer shifts of dyadic dilates of *one* function, called a wavelet. The first wavelet construction is due to Haar. His 1910 paper [30] presented an orthonormal basis of L^2 generated by the box function (See Figure 1.1).

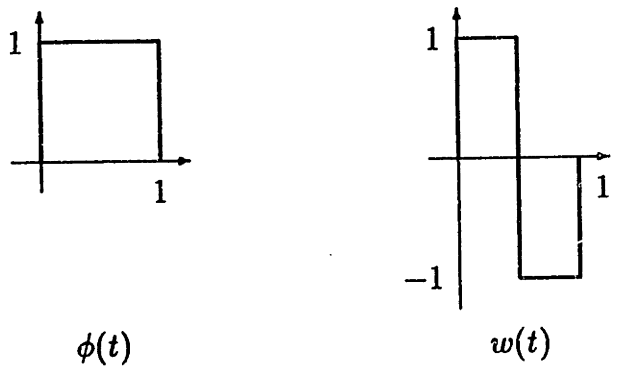


Figure 1.1: Haar scaling function and wavelet

Since Haar many examples of wavelet bases were constructed. In 1982 Strömberg [75] presented infinitely supported orthogonal piecewise polynomial wavelets. Similar bases were independently obtained by Battle [4] and Lemarié [43]. In 1985 Meyer gave an example of C^∞ wavelets with exponential decay [49], but Haar wavelets were the only ones combining orthogonality and finite support. A breakthrough was made

by Daubechies in 1988, when she constructed a finitely supported orthogonal scaling function and wavelet [18] (see Figure 1.2).

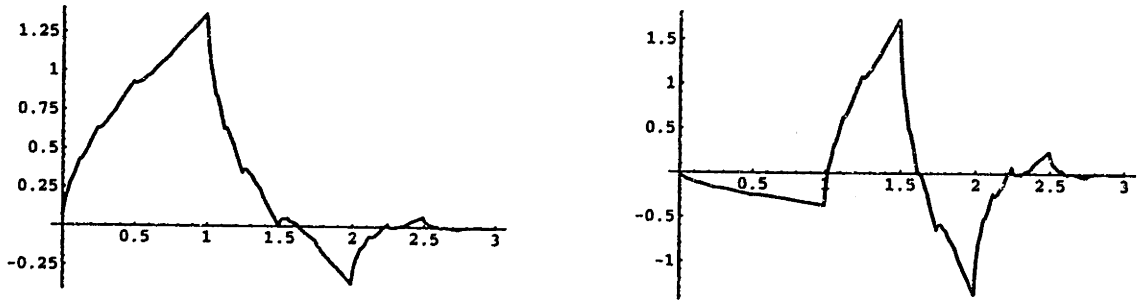


Figure 1.2: Daubechies orthogonal scaling function and wavelet

A very important notion of *multiresolution analysis* was invented by Meyer and Mallat in [50, 45]. Using it, many properties of wavelets such as completeness, linear independence, approximation, regularity, symmetry, orthogonality are well studied by now (see for example [13, 14, 20, 24, 37, 41, 83]). The theory of biorthogonal wavelet bases was also developed [12, 13, 77].

Based on the theoretical results, applied algorithms started to emerge. They use the discrete wavelet decomposition suggested by Mallat [46]. The amazing fact is that it can be performed in $O(n)$ arithmetic operations — even faster than the discrete Fourier transform. The main field of wavelet applications is signal processing and image compression (see [3, 7, 8, 47, 60, 68, 80, 81]), but wavelets also turn out to be useful in numerical analysis [5, 86, 87] and other branches of science and engineering.

The ideas of multiresolution analysis and wavelets have been generalized in many ways. Wavelet packets [16], m-band filter banks [80, 63], second generation wavelets [76] are just a few names. The generalization to *multiwavelets* is the topic of this thesis.

1.1.1 Multiwavelets—several wavelets with several scaling functions

We say that a *scaling function* $\phi(t)$ generates a multiresolution analysis (MRA) if

- Translates $\phi(t-k)$ are linearly independent and produce a basis of the subspace V_0;

- dilates $\phi(2^j t - k)$ generate subspaces V_j, $j \in \mathbf{Z}$, such that

$$\cdots \subset V_{-1} \subset V_0 \subset V_1 \subset \cdots \subset V_j \subset \cdots \tag{1.1}$$

$$\overline{\bigcup_{j=-\infty}^{\infty} V_j} = L^2(\mathbf{R}), \qquad \bigcap_{j=-\infty}^{\infty} V_j = 0. \tag{1.2}$$

- There is a wavelet $w(t)$, such that its translates $w(t-k)$ are linearly independent and produce a basis of the subspace W_0:

$$V_1 = V_0 \oplus W_0. \tag{1.3}$$

From (1.1)-(1.3) follows that

$$L^2(\mathbf{R}) = \bigoplus_{j=-\infty}^{\infty} W_j.$$

and $\{w_{kj} : w_{kj} = w(2^j - k),\ k, j \in \mathbf{Z}\}$ is a basis of L^2. In the Haar case, V_j is a subspace of piecewise constants on intervals $[\frac{k}{2^j}, \frac{k+1}{2^j}]$, $k \in \mathbf{Z}$. Moreover, $\phi(t-k)$ and $w(t-k)$, $k \in \mathbf{Z}$ produce orthogonal bases of V_0 and W_0 respectively and $\phi(t)$ is orthogonal to $w(t)$, thus Haar wavelets constitute an orthogonal basis of $L^2(\mathbf{R})$.

The definition of MRA implies two-scale relations for the scaling function and wavelet. Really, $\phi(2t-k)$, $k \in \mathbf{Z}$ is a basis of V_1. On the other hand, since $\phi(t) \in V_0$ and $V_0 \subset V_1$, $\phi(t)$ must be a linear combination of dilated translates of itself

$$\phi(t) = \sum_k C_k \phi(2t-k). \tag{1.4}$$

This equation is called a *dilation* (*refinement*) equation. Similarly, $w(t)$ must satisfy a *wavelet* equation

$$w(t) = \sum_k D_k \phi(2t-k). \tag{1.5}$$

The scaling function and wavelet have finite support, if and only if the number of coefficients C_k and D_k is finite. Actually, $\phi(t)$ is supported on $[0, N]$ if the summation in (1.4) is from 0 to N [19]. The Haar scaling function and wavelet are supported on $[0, 1]$ and satisfy two-scale relations (3.4), (3.5) with two coefficients:

$$\phi(t) = \phi(2t) + \phi(2t-1), \qquad w(t) = \phi(2t) - \phi(2t-1).$$

However, one can imagine a more general MRA setting when *several* scaling functions are allowed. This idea leads to the notion of *multiwavelets.* Multiwavelets have some advantages in comparison to scalar ones. For example, such features as short support, orthogonality, symmetry and vanishing moments are known to be important in signal processing. A scalar wavelet *cannot* possess all these properties at the same time [19]. On the other hand, a multiwavelet system *can* have all of them simultaneously. This suggests that multiwavelets could perform better in various applications (see Chapter 4).

In the "multi" case the notion of MRA stays the same except that now a basis for V_0 is generated by linearly independent translates of r scaling functions $\phi_0(t-k),\ldots,\phi_{r-1}(t-k)$, $k \in \mathbf{Z}$. Similarly to the scalar case ($r = 1$), vector $\boldsymbol{\phi}(t) = [\phi_0(t) \ \ldots \ \phi_{r-1}(t)]^T$, satisfies a *matrix* dilation equation

$$\boldsymbol{\phi}(t) = \sum_k \boldsymbol{C}_k \boldsymbol{\phi}(2t-k), \tag{1.6}$$

where, in agreement with $\boldsymbol{\phi}(t)$, coefficients $\boldsymbol{C}_k$ are r by r matrices. Associated with these scaling functions there are r wavelets $w_0(t),\ldots,w_{r-1}(t)$, satisfying the *matrix* wavelet equation

$$\boldsymbol{w}(t) = \sum_k \boldsymbol{D}_k \boldsymbol{\phi}(2t-k), \tag{1.7}$$

here $\boldsymbol{w}(t) = [w_0(t) \ \ldots \ w_{r-1}(t)]^T$ is a vector and $\boldsymbol{D}_k$ are r by r matrices. We call $\boldsymbol{\phi}(t)$ a *multi-scaling function* and $\boldsymbol{w}(t)$ a *multiwavelet.*

Again, finite support of $\boldsymbol{\phi}(t)$ and $\boldsymbol{w}(t)$ is ensured in the case of finite nuber of $\boldsymbol{C}$'s and $\boldsymbol{D}$'s, but there is no straightforward relation between length of support and the nuber of dilation coefficients. For some results on this subject see [62, 48].

Multiwavelets naturally generalize the scalar wavelets, however, they have several completely new features arising from the matrix nature of the equation (1.6). This makes them interesting both for theory and practice.

First multiwavelet constructions are due to Alpert [2] and Hervé [33]. In [2] was considered a multi-scaling function whose components are polynomials of degree $r-1$ supported on $[0,1]$. The simplest Alpert pair is presented in Figure 1.3. One can easily see why it generates an MRA. Really, one scaling function is the box function, so as we saw above, it is sum of two half boxes

$$\phi_0(t) = \phi_0(2t) + \phi_0(2t-1) \tag{1.8}$$

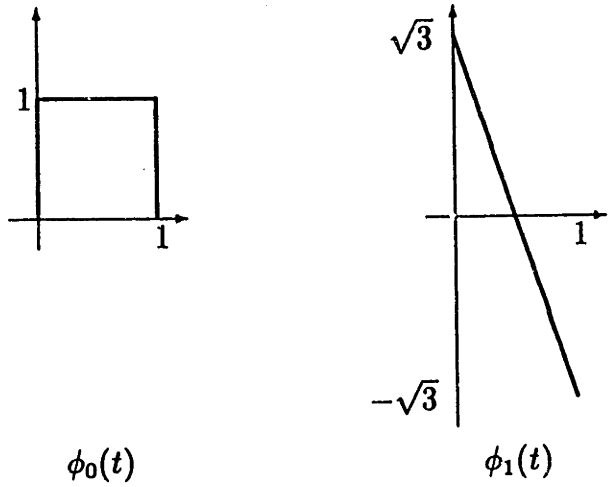


Figure 1.3: Simplest Alpert multiscaling function

The second scaling function – a piece of a line – turns out to be a linear combination of the dilated shifts of itself and the box function (see Figure 1.4):

$$\phi_1(t) = \frac{\sqrt{3}}{2}\phi_0(2t) + \frac{1}{2}\phi_1(2t) + \frac{\sqrt{3}}{2}\phi_0(2t-1) + \frac{1}{2}\phi_1(2t-1) \tag{1.9}$$

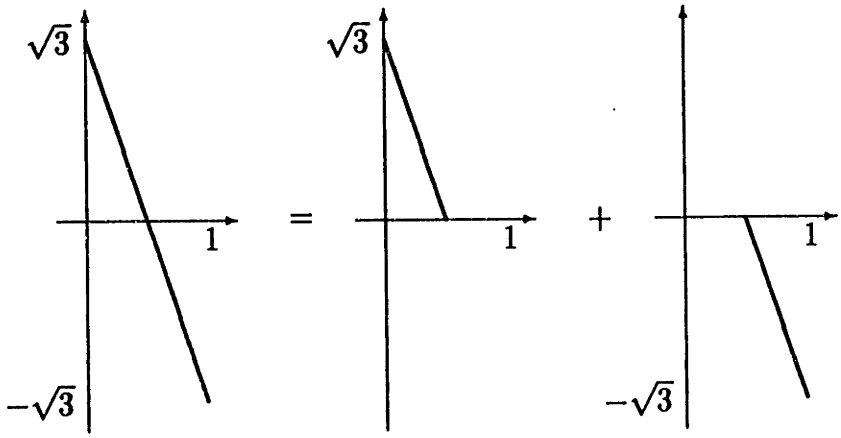


Figure 1.4: ϕ_1 is a linear combination of dilates of ϕ_0 and ϕ_1

Combining (1.8) and (1.9) we get the matrix dilation equation for the multi-scaling vector $\phi(t) = [\phi_0(t) \;\; \phi_1(t)]^T$:

$$\begin{bmatrix} \phi_0(t) \\ \phi_1(t) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \sqrt{3}/2 & 1/2 \end{bmatrix} \begin{bmatrix} \phi_0(2t) \\ \phi_1(2t) \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ -\sqrt{3}/2 & 1/2 \end{bmatrix} \begin{bmatrix} \phi_0(2t-1) \\ \phi_1(2t-1) \end{bmatrix}.$$

Other early multiwavelet constructions based on piecewise polynomials can be found in [28, 29, 56]

Using fractal interpolation, Geronimo, Hardin, and Massopust succeeded to construct continuous multi-scaling function $\phi(t) = [\phi_0(t) \;\; \phi_1(t)]^T$ with short support, symmetry and second approximation order [26]. The plot of this pair is presented in Figure 1.5. The results of [26] triggered many attempts to construct more examples ([9, 23, 40, 52, 53, 70]) as well as the systematic study of multi-scaling functions.

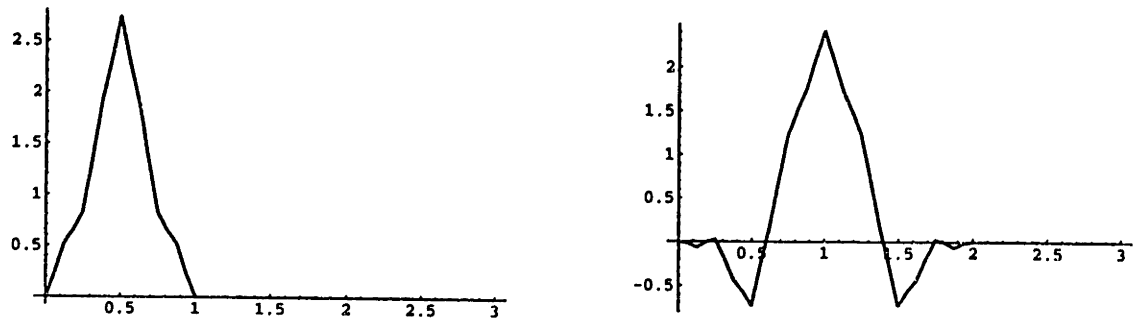


Figure 1.5: GHM symmetric orthogonal multi-scaling function with approximation order 2

Our goal was to build the whole theory starting from the dilation equation and describe the properties of multiwavelets in terms of the matrix coefficients C_k. We started from establishing conditions of orthogonality which allowed us to construct the multiwavelets corresponding to GHM multi-scaling functions (Figure 1.6) [69] (in parallel but independently with [26]). Then we found conditions of approximation and constructed several new examples [70]. In the time domain, such a construction requires solution of nonlinear system of equations, which is not convenient [32]. Therefore, we decided to switch to the frequency domain.

Working in the frequency domain, one faces the necessity to deal with the Fourier transformation of (1.6):

$$\widehat{\phi}(\omega) = \boldsymbol{P}\left(\frac{\omega}{2}\right) \widehat{\phi}\left(\frac{\omega}{2}\right),$$

where $\widehat{\phi} = [\widehat{\phi}_0, \ldots, \widehat{\phi}_{r-1}]^T$, $\widehat{\phi}_k = \int_{-\infty}^{\infty} \phi_k(t)\, e^{-it\omega}\, dt$, and $\boldsymbol{P}(\omega)$ is the *refinement mask* corresponding to $\phi(t)$, $\boldsymbol{P}(\omega) = \frac{1}{2} \sum_{k=0}^{N} \boldsymbol{C}_k\, e^{-i\omega k}$. In the scalar case, $P(\omega)$ is a *trigonometric polynomial.* In the vector case, $\boldsymbol{P}(\omega)$ becomes a *matrix trigonometric polynomial.* Conditions of existence of $\widehat{\phi}(\omega)$ are studied in [31, 15].

To ensure certain approximation order, $\boldsymbol{P}(\omega)$ must satisfy necessary and sufficient conditions in the frequency domain. Those conditions were formulated and proved

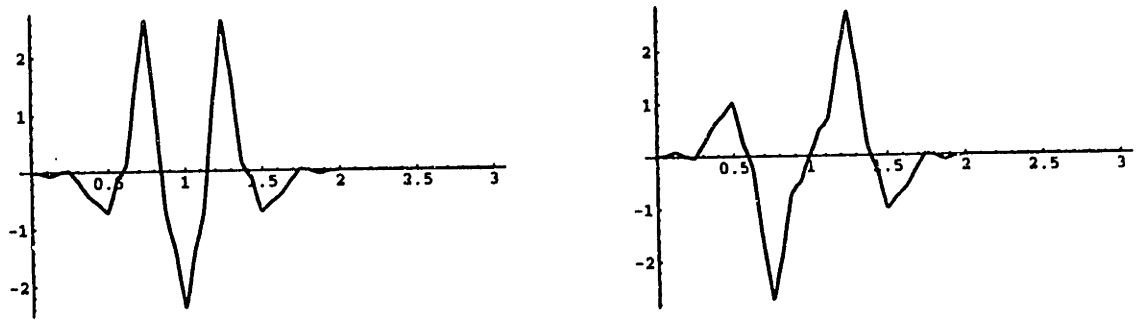


Figure 1.6: Symmetric GHM multiwavelet

in [32, 57]. But it was not clear what is the analog of a zero at π. In other words a factorization of the frequency response $\boldsymbol{P}(\omega)$ was needed. First such factorization was obtained by Plonka [57]:

$$\boldsymbol{P}(\omega) = \frac{1}{2^m}\, \boldsymbol{A}_{m-1}(2\omega) \dots \boldsymbol{A}_0(2\omega)\, \boldsymbol{P}^{(0)}(\omega)\, \boldsymbol{A}_0(\omega)^{-1} \dots \boldsymbol{A}_{m-1}(\omega)^{-1},$$

where $\boldsymbol{P}^{(0)}(\omega)$ is well-defined and $\boldsymbol{A}_0(\omega), \dots, \boldsymbol{A}_{m-1}(\omega)$ are matrices of a special form. This factorization is not unique, and I was able to find the whole set of possible factorizations [71]. It turned out that they are based on the *two-scale similarity transform* (TST). Moreover, TST appeared to be a very useful tool for the development of multi-wavelet theory. I studied properties of the TST and showed how it is connected with the approximation, orthogonality, symmetry and regularity of multi-scaling functions [71]. Similar results on regularity were obtained independently in [15]. L^2-smoothness of the refinable vectors was approached in [61]. Using TST and the general form of the factorization of $\boldsymbol{P}(\omega)$ we created a handy algorithm for the construction of multi-scaling functions with given approximation order [58].

Biorthogonal multiwavelet bases wre discussed in several recent papers [1, 17, 39, 51], but no explicit algorithms for the construction of the corresponding scaling functions and wavelets were given. We filled this gap and suggested the *cofactor method* for the construction of biorthogonal multiwavelets [74].

Some results on stability and linear independece of multiwavelets and multiwavelet frames can be found in [35, 59, 85].

Finally, first works on *multi-filter banks* [10, 11, 82, 89] and implementation of multiwavelets in signal processing [72, 79, 88] have appeared, showing that new constructions have certain advantages in comparison with scalar wavelets. Application

of multiwavelets in other fields such as PDE's, integral equations, fluid dynamics, are in the future.

1.2 Outline and Notation

This thesis presents my contribution to the theory and applications of multiwavelets. It is organized as follows.

Chapter 2 develops time domain methods. It states and proves conditions of approximation in time domain. It is shown how these conditions are connected to eigenvalues and eigenvectors of a doubly infinite matrix $\boldsymbol{L}$, consisting of the dilation coefficients $\boldsymbol{C_k}$. Then time domain techniques are applied to the construction of finite element multiwavelets. In the end of Chapter 2 we present time domain form of the conditions of orthogonality.

Chapter 3 is devoted to the Fourier methods. First, it states conditions of approximation in frequency domain. Then two-scale similarity transform (TST) is introduced and studied. After it, TST is used to obtain a general class of factorizations of the refinement mask, and to study pointwise regularity, symmetry and orthogonality of the refinable vectors. The factorization of the symbol $\boldsymbol{P}(\omega)$ leads to an algorithm for the construction of scaling functions with given approximation order and symmetry. Finally, the cofactor method is presented. With its help biorthogonal multi-filter banks can be easily built up, starting from the low-pass analysis part. It turns out that synthesis filters, constructed using cofactor method have the same approximation and symmetry as the initial analysis filters. We complete the finite element multi-filter bank as an example. Chapter 3 ends with the description of how orthogonal multiwavelets can be obtained from a given orthogonal multi-scaling function.

Chapter 4 points out the ways to apply multiwavelets in signal processing. It is not straightforward, because a preprocessing of the input data is needed. Several types of such preprocessing are presented. A special family of multiwavelets (constrained pairs) is designed especially for the image processing. Also a method for symmetric data extension at the boundaries is suggested. Then all these techniques are applied to signal denoising and image compression. The results of the numerical experiments show that multiwavelets can behave better that comparable scalar filters.

Everywhere in the text $i^2 = -1$ and $\widehat{(\cdot)}$ denotes a Fourier image $\int e^{-i\omega t}(\cdot)dt$. A hat over a vector or a matrix means that we take Fourier transform of each component. The scalar product of two vector functions is understood in the L^2 sense: $< \boldsymbol{\phi}(\cdot), \boldsymbol{\psi}(\cdot) >= \int \boldsymbol{\phi}(\cdot)\boldsymbol{\psi}^*(\cdot)d\cdot$. $(D^k\cdot)$ denotes the derivative of order k of $(\cdot)$. All

vectors and matrices are given in bold face, while scalars are light. Vectors usually have length r and matrices have size r by r. Generally, $\boldsymbol{C}_k$, $\boldsymbol{D}_k$ are constant matrices and $\boldsymbol{M}(\omega), \boldsymbol{P}(\omega), \boldsymbol{Q}(\omega), \ldots$ are $r \times r$ matrices with components depending on the real variable ω. We assume that those components have a sufficient number of derivatives and are 2π periodic.

Chapter 2

Time Domain Methods

In this chapter we discuss properties of multi-scaling functions and multiwavelets in the time domain. This means that we are going to work with the coefficients C_k of the matrix dilation equation

$$\phi(t) = \sum_{k=0}^{N} C_k \phi(2t-k). \tag{2.1}$$

Here $\phi(t) = [\phi_0(t) \quad \dots \quad \phi_{r-1}(t)]^T$ is a multi-scaling function and $C_0, \dots, C_N$ are r by r matrices with constant elements.

As we will see below, an important role in the time domain is played by a doubly infinite block matrix L

$$L = \begin{bmatrix} \dots & & & & & & & & \\ \dots & C_3 & C_2 & C_1 & C_0 & & & & \\ & & \dots & C_3 & C_2 & C_1 & C_0 & & \\ & & & & \dots & C_3 & C_2 & C_1 & C_0 \\ & & & & & & & & \dots \end{bmatrix}. \tag{2.2}$$

With its help equation (2.1) can be rewritten in the doubly infinite form:

$$\Theta(t) = L\Theta(2t), \tag{2.3}$$

where $\Theta(t) = [\dots \phi(t-1) \quad \phi(t) \quad \phi(t+1) \dots]^T$.

2.1 Conditions of Approximation in Time Domain

We say that a multi-scaling function $\phi(t)$ has approximation order m if each polynomial t^j, $j = 0, \ldots, m-1$ is a linear combination of integer translates $\phi(t-k)$:

$$t^j = \sum_{-\infty}^{\infty} y_k^{(j)} \phi(t-k) \text{ a.e.} \quad j = 0, \ldots, m-1. \tag{2.4}$$

In accordance with ϕ's, $y_k^{(j)}$ are constant row-vectors of length r.

Theorem 2.1.1 *Suppose that $\phi(t) \in \bar{L}^1$ and integer translates $\phi_j(t-k)$, $j = 0, \ldots, r-1$, $k \in \mathbf{Z}$ are linearly independent. Then $\phi(t)$ provides approximation order m if and only if $\boldsymbol{L}$ has eigenvalues 2^{-j} corresponding to the left eigenvectors $\boldsymbol{y}^{(j)} = [\ldots \boldsymbol{y}_0^{(j)} \ \boldsymbol{y}_1^{(j)} \ \boldsymbol{y}_2^{(j)} \ldots]$,*

$$\boldsymbol{y}_k^{(j)} = \sum_{l=0}^{j} \binom{j}{l} (-k)^{j-l} \boldsymbol{u}_l \qquad j = 0, \ldots, m-1, \tag{2.5}$$

where $\boldsymbol{u}_j$ are constant row-vectors.

Proof.

1. *Necessity.* Approximation order m, implies that polynomials $1, t, \ldots, t^{m-1}$ are in the scaling subspace spanned by $\phi(t-k)$:

$$t^j = G_j(t) = \sum_{k=-\infty}^{\infty} \boldsymbol{y}_k^{(j)} \phi(t-k) = \boldsymbol{y}^{(j)} \boldsymbol{\Theta}(t),$$

From (2.3) follows that $G_j(t) = \boldsymbol{y}^{(j)} \boldsymbol{\Theta}(t) = \boldsymbol{y}^{(j)} \boldsymbol{L} \boldsymbol{\Theta}(2t)$. On the other hand, $G_j(t) = t^j = 2^{-j}(2t)^j = 2^{-j} G_j(2t) = 2^{-j} \boldsymbol{y}^{(j)} \boldsymbol{\Theta}(2t)$. Thus,

$$\boldsymbol{y}^{(j)} \boldsymbol{L} \boldsymbol{\Theta}(2t) = 2^{-j} \boldsymbol{y}^{(j)} \boldsymbol{\Theta}(2t).$$

Linear independence of translates $\phi(2t-k)$ gives

$$\boldsymbol{y}^{(j)} \boldsymbol{L} = 2^{-j} \boldsymbol{y}^{(j)}$$

which means that $\boldsymbol{L}$ has eigenvalue 2^{-j} with left eigenvector $\boldsymbol{y}^{(j)}$.

This argument is almost reversible: If $\boldsymbol{L}$ has eigenvalues $1, \frac{1}{2}, \ldots, (\frac{1}{2})^{m-1}$ with eigenvectors $\boldsymbol{y}^{(j)}$, then $G_j(t) = \sum_{-\infty}^{\infty} \boldsymbol{y}_k^{(j)} \phi(t-k) = 2^{-j} G_j(2t)$.

Further, the presence of $1, t, \ldots, t^{m-1}$ in the first scaling subspace defines the structure of $\boldsymbol{y}^{(j)}$. Really, from (2.4)

$$G_j(t+1) = (t+1)^j = \sum_{-\infty}^{\infty} \boldsymbol{y}_k^{(j)} \phi(t+1-k) = \sum_{-\infty}^{\infty} \boldsymbol{y}_{k-1}^{(j)} \phi(t-k)$$

and

$$G_j(t) - G_j(t+1) = \textstyle\sum_{k=-\infty}^{\infty} (\boldsymbol{y}_k^{(j)} - \boldsymbol{y}_{k-1}^{(j)}) \phi(t-k) =$$
$$t^j - (t+1)^j = -\textstyle\sum_{l=0}^{j-1} \begin{pmatrix} j \\ l \end{pmatrix} t^l = -\sum_{k=-\infty}^{\infty} \sum_{l=0}^{j-1} \begin{pmatrix} j \\ l \end{pmatrix} \boldsymbol{y}_k^{(l)} \phi(t-k).$$

By linear independence of the translates $\phi(t-k)$

$$(\boldsymbol{y}_k^{(j)} - \boldsymbol{y}_{k-1}^{(j)}) = -\sum_{l=0}^{j-1} \begin{pmatrix} j \\ l \end{pmatrix} \boldsymbol{y}_k^{(l)}$$

or

$$\sum_{l=0}^{j} \begin{pmatrix} j \\ l \end{pmatrix} \boldsymbol{y}_k^{(l)} = \boldsymbol{y}_{k-1}^{(j)}.$$

It is easy to check by direct substitution that

$$\boldsymbol{y}_k^{(j)} = \sum_{l=0}^{j} (-1)^l \begin{pmatrix} j \\ l \end{pmatrix} k^l \boldsymbol{u}_{j-l} = \boldsymbol{y}_k^{(j)} = \sum_{l=0}^{j} (-k)^{j-l} \begin{pmatrix} j \\ l \end{pmatrix} \boldsymbol{u}_l$$

is the solution of this finite difference equation. Here $\boldsymbol{u}_j$ are some constant vectors.

2. *Sufficiency.* Suppose $\boldsymbol{L}$ has eigenvalues 2^{-j} corresponding to the eigenvectors $\boldsymbol{y}^{(j)}$ defined by (2.5). As was mentioned in **1**, this means that

$$G_j(t) = \sum_{k=-\infty}^{\infty} \boldsymbol{y}_k^{(j)} \phi(t-k) = 2^{-j} G_j(2t). \tag{2.6}$$

By (2.5) $\boldsymbol{y}_k^{(0)} - \boldsymbol{y}_{k-1}^{(0)} = \boldsymbol{u}_0 - \boldsymbol{u}_0 = \boldsymbol{0}$, so $G_0(t) - G_0(t+1) = 0$. In other words, $G_0(t)$ is a periodic function and by (2.6) $G_0(t) = G_0(2t)$. Because of assumption $\phi(t) \in L^1$, $G_0(t)$ is also in L^1, and we can apply the Birkhoff Ergodic Theorem [84] and get

$$G_0(t) = \sum_{k=-\infty}^{\infty} \boldsymbol{u}_0 \phi(t+k) = \text{constant} \quad \text{a.e.}$$

We normalize $\boldsymbol{u}_0$ such that $G_0(t) = 1$.

Substitution of $j = 1$ into (2.5) gives $\boldsymbol{y}_k^{(1)} - \boldsymbol{y}_{k-1}^{(1)} = -\boldsymbol{u}_{(0)}$, $G_1(t) - G_1(t+1) = \sum_{k=-\infty}^{\infty} (\boldsymbol{y}_k^{(1)} - \boldsymbol{y}_{k-1}^{(1)}) \phi(t+k) = -\sum_{k=-\infty}^{\infty} \boldsymbol{u}_{(0)} \phi(t+k) = -1$. This means that $F_1(t) =$

$G_1(t)-t$ is a periodic function. By (2.6) $G_1(t) = 2^{-1}G_1(2t)$, and the Birkhoff Ergodic Theorem gives $F_1(t) = 0$ or

$$G_1(t) = \sum_{k=-\infty}^{\infty} y_k^{(1)}\phi(t+k) = t \quad \text{a.e.}$$

In the general case,

$$\begin{aligned} &G_j(t) - G_j(t+1) = \textstyle\sum_{k=-\infty}^{\infty}(y_k^{(j)} - y_{k-1}^{(j)})\phi(t-k) \\ &= -\textstyle\sum_{k=-\infty}^{\infty}\sum_{l=0}^{j-1}\begin{pmatrix} j \\ l \end{pmatrix} y_k^{(l)}\phi(t-k) = -\sum_{l=0}^{j-1}\begin{pmatrix} j \\ l \end{pmatrix} t^l \cdot \end{aligned}$$

$F_j(t) = G_j(t) - t^j$ is a periodic function because

$$\begin{aligned} &F_j(t+1) = G_j(t+1) - (t+1)^j \\ &= G_j(t) + \textstyle\sum_{l=0}^{j-1}\begin{pmatrix} j \\ l \end{pmatrix} t^l - \sum_{l=0}^{j}\begin{pmatrix} j \\ l \end{pmatrix} t^l = G_j(t) - t^j = F_j(t). \end{aligned}$$

By (2.6) $G_j(t) = 2^{-j}G_j(2t)$, and as in previous cases we get

$$G_j(t) = \sum_{k=-\infty}^{\infty} y_k^{(j)}\phi(t-k) = t^j \quad a.e.$$

So, we have proved that all t^j, $j < m$, are in the scaling subspace. ∎

Remark 2.1.2 The Birkhoff Ergodic Theorem can be avoided by assuming that $\phi(t) < \infty$ and $\sum_{k=-\infty}^{\infty}\phi(t-k)$ is continuous at a point $t = t_0$.

Now the structure of vectors $y^{(j)}$ is clear. The constant vectors u_j in (2.5) can be found from the fact that $y^{(j)}$ is an eigenvector of L corresponding to the eigenvalue 2^{-j}.

Theorem 2.1.3 *Suppose $y^{(j)}$ is defined by (2.5) and L corresponds to a multi-scaling function with approximation order m. Then $y^{(j)}L = 2^{-j}y^{(j)}$, $j = 0, \ldots, m-1$ if and only if the following finite equations are satisfied:*

$$\begin{aligned} &\textstyle\sum_k y_k^{(j)}C_{2k+1} = 2^{-j}u_j \\ &\textstyle\sum_k y_k^{(j)}C_{2k} = 2^{-j}y_1^{(j)} = 2^{-j}\sum_{l=0}^{j}(-1)^{j-l}\begin{pmatrix} j \\ l \end{pmatrix} u_l \end{aligned} \tag{2.7}$$

Proof. Using the block structure of L we rewrite $y^{(j)}L = 2^{-j}y^{(j)}$ as the infinite set of equations

$$\begin{aligned} &\textstyle\sum_k y_{k+l}^{(j)}C_{2k+1} = 2^{-j}y_{2l}^{(j)} \\ &\textstyle\sum_k y_{k+l}^{(j)}C_{2k} = 2^{-j}y_{2l+1}^{(j)} \end{aligned} \qquad l \in \mathbf{Z} \tag{2.8}$$

Conditions (2.7) are the particular case of (2.8) with $l = 0$. So we need only to prove that if (2.8) is true for $l = 0$ then it is true for all l.

We do it by induction on j. For $j = 0$ (2.8) simplifies since $y_k^{(0)} = u_0$ for all k. In fact, (2.8) becomes

$$\sum_k y_0^{(0)} C_{2k+1} = u_0$$
$$\sum_k y_0^{(0)} C_{2k} = u_0,$$

which is independent of l and equals (2.7) with $j = 0$.

Now suppose that (2.7) is valid for $j = 0, \ldots, n-1$. We show that the first equation in (2.8) is valid for $j = n$ and all $l \in \mathbf{Z}$ as follows. First we recall that

$$\sum_k y_{k+l}^{(n)} \phi(t-k) = \sum_k y_k^{(n)} \phi(x-l-k) = (x-l)^n =$$
$$\sum_{j=0}^n \binom{n}{j} t^j (-l)^{n-j} = \sum_{j=0}^n \binom{n}{j} (-l)^{n-j} \sum_k y_k^{(j)} \phi(t-k)$$

and independence implies

$$y_{k+l}^{(n)} = \sum_{j=0}^n \binom{n}{j} (-l)^{n-j} y_k^{(j)}$$

Then, using the induction hypothesis and conditions (2.7) we compute

$$\begin{aligned}
\sum_k y_{k+l}^{(j)} C_{2k+1} &= \sum_k \sum_{j=0}^n \binom{n}{j} (-l)^{n-j} y_k^{(j)} C_{2k+1} \\
&= \sum_{j=0}^n \binom{n}{j} (-l)^{n-j} \sum_k y_k^{(j)} C_{2k+1} \\
&= \sum_{j=0}^n \binom{n}{j} (-l)^{n-j} 2^{-j} u_j \\
&= 2^{-n} \sum_{j=0}^n \binom{n}{j} (-2l)^{n-j} y_0^{(j)} \\
&= 2^{-n} y_{2l}^{(n)}
\end{aligned}$$

An analogous computation shows that the second equation in (2.8) holds for $j = n$ and all $l \in \mathbf{Z}$, and finishes the proof. ∎

Results of Theorems 2.1.1, 2.1.3 can be summarized in the following form.

Theorem 2.1.4 *A multi-scaling function $\phi(t) \in L^1$ with linearly independent translates $\phi(t-k), k \in \mathbf{Z}$ has approximation order m if and only if there exist vectors u_j, $j = 0, \ldots, m-1$ such that relations (2.7) are satisfied.*

Remark 2.1.5 Some discussion on the assumption of linear independence of translates $\phi(t-k)$ can be found in [32, 35].

Remark 2.1.6 It is easy to check (see [32]) that in the scalar case conditions (2.7) reduce to the sum rules

$$\sum_{k=0}^{N}(-1)^k k^j C_k = 0 \quad j = 0, \ldots, m-1,$$

which ensure approximation order m [19].

Example 2.1.7 The GHM scaling functions displayed in Figure 1.5 satisfy a dilation equation (2.1) with four coefficients:

$$C_0 = \begin{bmatrix} \frac{3}{5} & \frac{4\sqrt{2}}{5} \\ -\frac{1}{10\sqrt{2}} & -\frac{3}{10} \end{bmatrix}, \quad C_1 = \begin{bmatrix} \frac{3}{5} & 0 \\ \frac{9}{10\sqrt{2}} & 1 \end{bmatrix},$$

$$C_2 = \begin{bmatrix} 0 & 0 \\ \frac{9}{10\sqrt{2}} & -\frac{3}{10} \end{bmatrix}, \quad C_3 = \begin{bmatrix} 0 & 0 \\ -\frac{1}{10\sqrt{2}} & 0 \end{bmatrix}$$

These C's satisfy condition (2.7) for $m = 2$ and starting vectors $u_0 = [\sqrt{2} \;\; 1]$ and $u_1 = [1/\sqrt{2} \;\; 1]$. Thus GHM multi-scaling function provides approximation order 2. The fact that both a constant and linear function can be exactly reproduced can be checked directly by plotting corresponding linear combinations of ϕ_0 and ϕ_1:

$$1 = \sum_k u_0 \phi(t-k) = \ldots + \sqrt{2}\phi_0(t) + \phi_1(t) + \sqrt{2}\phi_0(t-1) + \phi_1(t-1) + \ldots$$

$$t = \sum_k (u_1 - k u_0)\phi(t-k) = \ldots + \frac{1}{\sqrt{2}}\phi_0(t) + \phi_1(t) - \frac{1}{\sqrt{2}}\phi_0(t-1) + 0\phi_1(t-1) + \ldots$$

See Figure 2.1. The boundary artifacts appear because the plotted combinations are finite.

Approximation by translates of refinable vectors is also considered in [36, 44, 57].

2.2 Finite Element Multiwavelets: Semiorthogonality

Conditions of approximation (2.7) can be viewed as a system of nonlinear equations for the components of matrices C_k and starting vectors u_j. Hence, (2.7) can be used for the construction of multi-scaling functions with a given approximation order. Unfortunately it is not a very practical way. The nonlinear system (2.7) can be effectively solved only in the case of low approximation orders and small number of

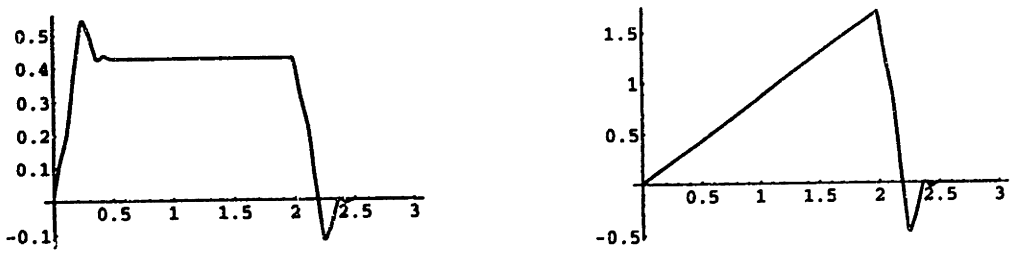


Figure 2.1: Representation of 1 and t by the linear combination of translates of GHM scaling functions.

dilation coefficients (for examples see [70, 72]). Nevertheless conditions of approximation in time domain are useful for some constructions. As an example we present finite element multi-scaling (FEM) functions [73]. Another example — constrained pairs, is considered in Section 4.3

A FEM function of order r is a multi-scaling function $\phi(t) = [\phi_0(t) \ \dots \ \phi_{r-1}]^T$ consisting of finite elements ϕ_j. These finite elements are piecewise polynomials of odd degree $2r-1$, with $r-1$ continuous derivatives. For all r, $\phi_j(t)$ is nonzero only over two intervals. The function value and $r-1$ derivatives are given at each integer node. The functions are alternately symmetric and antisymmetric about $t = 1$, if we place them on the interval $[0,2]$. Their translates span V_0. Rescaled by $t \to 2t$, the space V_1 of fine-mesh elements clearly contains the space V_0 of coarse-mesh elements. Therefore each basis function $\phi_k(t)$ in V_0 is a combination of translates of the r fine-mesh basis functions $\phi_j(2t)$, $j = 0, \dots, r-1$. This is the dilation equation:

$$\phi(t) = C_0\phi(2t) + C_1\phi(2t-1) + C_2\phi(2t-2). \tag{2.9}$$

Since the support is $[0,2]$, the only coefficients are C_0, C_1, C_2. As there are r basis functions at each node, those coefficients C_k are r by r matrices. $\phi(t)$ represents the column vector of basis functions $[\phi_0(t) \dots \phi_{r-1}(t)]^T$. These are polynomials of degree $2r-1$ on the pieces $[0,1]$ and $[1,2]$ determined by

$$(\frac{d}{dt})^{k-1}\phi_j(1) = \delta_{kj} \tag{2.10}$$

$$(\frac{d}{dt})^{k-1}\phi_j(0) = (\frac{d}{dt})^{k-1}\phi_j(2) = 0 \tag{2.11}$$

for $k, j = 1, \dots, r$. There is symmetry for even j and antisymmetry for odd j:

$$\phi_j(2-t) = (-1)^j\phi_j(t). \tag{2.12}$$

Really, since $C_0\phi(2)$ and $C_2\phi(0)$ give no contribution, by (2.11), the derivatives of (2.9) at $t = 1$ are

$$(\frac{d}{dt})^{j-1}\phi(1) = C_1 2^{j-1}(\frac{d}{dt})^{j-1}\phi(1). \tag{2.13}$$

The left side is the jth column of the identity matrix, by (2.10). Dividing by 2^{j-1} therefore gives the jth column of C_1. We now know the diagonal matrix $C_1 = \mathrm{diag}(1, 2^{-1}, \ldots, 2^{-r+1})$.

Rewriting (2.12) in the vector form we get

$$\phi(t) = S\phi(2t-1), \tag{2.14}$$

where $S = \mathrm{diag}(1, -1, \ldots, (-1)^{r-1})$. Write down the equation (2.9) for $\phi(2-2t)$, use (2.14) on both sides and compare the result with (2.9):

$$\begin{aligned} &C_0\phi(2t) + C_1(2t-1) + C_2\phi(2t-2) \\ &\quad = SC_2S^{-1}\phi(2t) + SC_1S^{-1}\phi(2t-1) + SC_0S^{-1}\phi(2t-2). \end{aligned}$$

By the linear independence of translates $\phi(t-k)$, $C_0 = SC_2S^{-1}$ and the only coefficient we need to find is C_2. It can be done using the conditions of approximation (2.7).

Integer translates of finite elements of order r represent exactly all polynomials $1, \ldots, t^{2r-1}$. In other words, $\phi(t)$ provides approximation order $2r$. Thus, matrices C_0, C_1, C_2 must satisfy conditions (2.7) with some starting vectors $u_0, \ldots, u_{r-1}$. In our case conditions (2.7) are

$$u_jC_1 = 2^{-j}u_j \qquad j = 0, \ldots, 2r-1 \tag{2.15}$$

$$u_jC_0 + y_1^{(j)}C_2 = 2^{-j}y_1^{(j)} \qquad j = 0, \ldots, 2r-1, \tag{2.16}$$

where $y_1^{(j)} = 2^{-j}\sum_{l=0}^{j}(-1)^{j-l}$ is defined by (2.5). $C_1 = \mathrm{diag}(1, 2^{-1}, \ldots, 2^{-r+1})$ so from (2.15) follows that $u_j = [0 \ \ldots \ a_j \ \ldots \ 0]$. To determine constants a_j $j = 0, \ldots, r-1$ we recall that

$$t^j = \sum_k y_k^{(j)}\phi(t-k) \quad j = 0, \ldots, r-1 \tag{2.17}$$

and $y_0^{(j)} = u_j$, Differentiation of (2.17) j times gives

$$j! = \sum_k y_k^{(j)}\phi^{(j)}(t-k) \quad j = 0, \ldots, r-1 \tag{2.18}$$

By formulas (2.10), (2.11), at $t = 1$ (2.18) becomes

$$j! = y_0^{(j)}\phi^{(j)}(1) = u_j \cdot [0 \ldots 1 \ldots 0]^T$$

which means that $a_j = j!$ and

$$u_j = [0 \ \dots \ j! \ \dots \ 0], \quad j = 0, \dots r-1. \tag{2.19}$$

Further, as C_1 is r by r, it has only r eigenvectors. Then (2.15) implies $u_j = 0$ for $j \leq r$, and (2.16) can be rewritten as

$$y_1^{(j)} C_2 = 2^{-j} y_1^{(j)} \quad j = r, \dots, 2r-1.$$

In other words, the vectors $y_{-1}^{(j)}$ are left eigenvectors of C_2 corresponding to the eigenvalues $\lambda_j = 2^{-j}$, $j = r, \dots, 2r-1$. This means that $C_2 = U^{-1} \Lambda U$, where $\Lambda = \text{diag}(2^{-r}, \dots, 2^{-2r+1})$, U is a r by r matrix with rows $y_1^{(j)}$ $j = r, \dots, 2r-1$. We have proved the following fact:

Theorem 2.2.1 *Finite element multi-scaling function of order r satisfies a dilation equation* (2.9) *with three coefficients. The eigenvalues of C_2 are $2^{-r}, \dots, 2^{-2r+1}$. They enter a diagonal matrix Λ. The left eigenvectors of $C_2 = U^{-1} \Lambda U$ are the rows of U:*

$$[U]_{kj} = (-1)^{r+k-j} \frac{(r+k)!}{(r+k-j)!}.$$

Then $C_0 = S C_2 S^{-1}$ and $C_1 = diag(1, \dots, 2^{-r+1})$ with $S = \text{diag}(1, -1, \dots, (-1)^{-r+1})$.

Example 2.2.2 The piecewise cubic case $r = 2$ (Hermite cubics) leads to

$$\Lambda = \begin{bmatrix} \frac{1}{4} & 0 \\ 0 & \frac{1}{8} \end{bmatrix} \quad \text{and} \quad U = \begin{bmatrix} 1 & 2 \\ 1 & 3 \end{bmatrix}.$$

Then the matrices C_0 and C_1 and C_2 are

$$C_0 = \begin{bmatrix} \frac{1}{2} & -\frac{3}{4} \\ \frac{1}{8} & -\frac{1}{8} \end{bmatrix}, \qquad C_1 = \begin{bmatrix} 1 & 0 \\ 0 & \frac{1}{2} \end{bmatrix}, \qquad C_2 = \begin{bmatrix} \frac{1}{2} & \frac{3}{4} \\ -\frac{1}{8} & -\frac{1}{8} \end{bmatrix}.$$

These are coefficients of the lowpass filter which lies behind the scaling functions. $\phi_0(t)$ and $\phi_1(t)$ are plotted in Figure 2.2.

Now we turn to the construction of highpass filter coefficients D_k which enter the wavelet equation. We choose the wavelets $w_0, \dots w_{r-1}$ to be a basis for the orthogonal complement W_0 of V_0 in V_1: $V_1 = V_0 \oplus W_0$ (semiorthogonal wavelets). At the same time the wavelets are combinations of $\phi_0(2t-k), \dots, \phi_{r-1}(2t-k)$. This gives the wavelet equation with matrix coefficients D_k:

$$w(t) = \sum_{k=0}^{4} D_k \phi(2t-k), \tag{2.20}$$

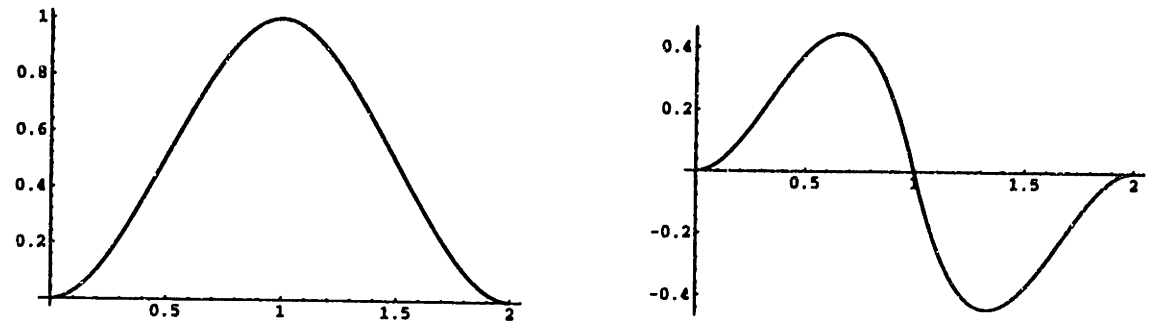


Figure 2.2: Hermite cubic multi-scaling function

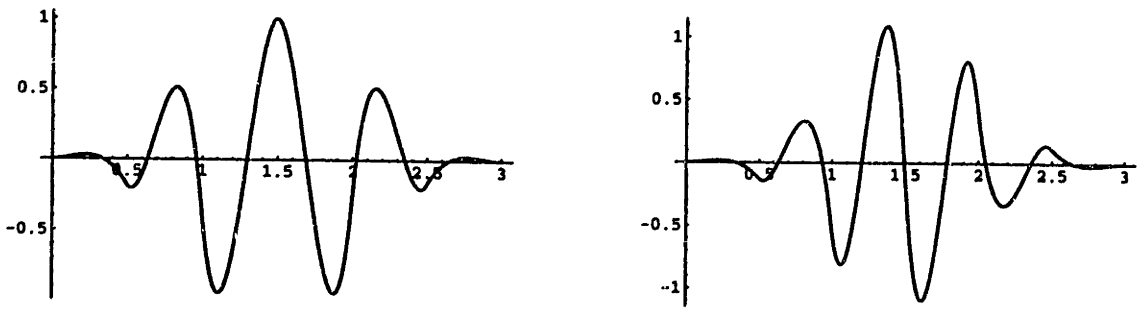


Figure 2.3: Hermite semiorthogonal multiwavelet

where $w(t) = [w_0(t) \ \dots \ w_{r-1}]^T$ is the multiwavelet. We compute D's from C's. Figure 2.3 shows the Hermite semiorthogonal wavelets w_0 and w_1.

The first question is, why five D's? The support of $\phi(2t-k)$ is $[\frac{k}{2}, \frac{k}{2}+1]$. With coefficients $D_0, \dots, D_4$, the support of the wavelets in (2.20) will be $[0,3]$. Orthogonality against $\phi(t)$ and its translates will give four matrix equations for the D's:

$$\begin{aligned} w(t) \cdot \phi^T(t+1) &= 0 \\ w(t) \cdot \phi^T(t) &= 0 \\ w(t) \cdot \phi^T(t-1) &= 0 \\ w(t) \cdot \phi^T(t-2) &= 0, \end{aligned} \tag{2.21}$$

where $[w(t) \cdot \phi^T(t-l)]_{jk} = \int w_j(t)\phi_k(t-l)dt$, $j,k = 0, \dots, r-1$. There is a nontrivial solution which we want. If we assume a shorter support with fewer D's the homogeneous system becomes square and leads to $D_k = 0$.

Substitute the w's from wavelet equation (2.20) into (2.21). Use the dilation equation (2.9) for the ϕ's. Then we integrate $\phi_i(2t-k)\phi_j(2t-m)$. A change of variables brings these inner product integrals back to

$$\int \phi_i(2t-k)\phi_j(2t-m)dt = \frac{1}{2}\int \phi_i(t)\phi_j(t-m+k)dt. \tag{2.22}$$

The ϕ_i are supported on $[0,2]$, so the only inner products we need are in the two matrices

$$X = \phi(t) \cdot \phi^T(t) \quad \text{and} \quad Y = \phi(t) \cdot \phi^T(t-1) = \phi(t+1) \cdot \phi^T(t). \tag{2.23}$$

We substitute the dilation equation into (2.23), and use (2.22) to bring all arguments involving $2t$ back to t. The result is two matrix equations for the matrices X and Y:

$$\begin{aligned} 2X &= C_0XC_0^T + C_1Y^TC_0^T + C_0YC_1^T + C_1XC_1^T + C_2Y^TC_1^T + C_1YC_2^T + C_2XC_2^T \\ 2Y &= C_1YC_0^T + C_2XC_0^T + C_2YC_1^T. \end{aligned} \tag{2.24}$$

This determines X and Y. Actually the block vector of inner products $[Y \ \ X \ \ Y]^T$ is the 1 eigenvector of the transition operator studied in [65]. This means that X and Y are unique up to a scalar factor.

Then X and Y enter the orthogonality conditions (2.21). Remember that substituting in (2.21) for $w(t)$ introduced the unknown D's, and substituting for $\phi(t)$ introduced the known C's. After these substitutions (2.21) becomes

$$\begin{aligned} &D_0(Y^TC_1^T + XC_2^T) + D_1Y^TC_2^T = 0 \\ &D_0(XC_0^T + YC_1^T) + D_1(Y^TC_0^T + XC_1^T + YC_2^T) + D_2(Y^TC_1^T + XC_2^T) + D_3Y^TC_2^T = 0 \\ &D_1YC_0^T + D_2(XC_0^T + YC_1^T) + D_3(Y^TC_0^T + XC_1^T + YC_2^T) + D_4(Y^TC_1^T + XC_2^T) = 0 \\ &D_3YC_0^T + D_4(XC_0^T + YC_1^T) = 0. \end{aligned} \tag{2.25}$$

We solve this system of $4r^2$ homogeneous equations for $5r^2$ entries of $\boldsymbol{D}_0, \ldots, \boldsymbol{D}_4$. We pick out the solution that has $\boldsymbol{D}_2 = \boldsymbol{I}$. Then the symmetry-antisymmetry of the ϕ's is true also for the wavelets.

The property $\boldsymbol{C}_0 = \boldsymbol{S}\boldsymbol{C}_2\boldsymbol{S}^{-1}$ of the $\boldsymbol{C}$'s extends to the $\boldsymbol{D}$'s with the sign matrix $\boldsymbol{S} = \mathrm{diag}(1, -1, 1, \ldots, (-1)^{r-1})$:

$$\boldsymbol{D}_0 = \boldsymbol{S}\boldsymbol{D}_4\boldsymbol{S}^{-1} \quad \text{and} \quad \boldsymbol{D}_1 = \boldsymbol{S}\boldsymbol{D}_3\boldsymbol{S}^{-1}. \tag{2.26}$$

The first two equations (2.25) become identical to the last two, using this pattern for the $\boldsymbol{D}$'s and verifying the corresponding pattern $\boldsymbol{X} = \boldsymbol{S}\boldsymbol{X}\boldsymbol{S}^{-1}$ and $\boldsymbol{Y} = \boldsymbol{S}\boldsymbol{Y}^T\boldsymbol{S}^{-1}$:

$$[\boldsymbol{X}]_{kj} = [\boldsymbol{X}]_{jk} = \int \phi_k(t)\phi_j(t)dt = \int \phi_k(2-t)\phi_j(2-t)dt =$$

$$(-1)^{k+j}\int \phi_k(t)\phi_j(t)dt = (-1)^{k+j}[\boldsymbol{X}]_{kj}$$

$$[\boldsymbol{Y}]_{kj} = \int \phi_k(t)\phi_j(t-1)dt = \int \phi_k(3-t)\phi_j(2-t)dt =$$

$$(-1)^{k+j}\int \phi_k(t-1)\phi_j(t)dt = (-1)^{k+j}[\boldsymbol{Y}]_{jk}$$

Thus (2.21) reduces to two matrix equations for two unknowns $\boldsymbol{D}_3$ and $\boldsymbol{D}_4$. Our Mathematica subroutine constructs dilation coefficients $\boldsymbol{C}_0$, $\boldsymbol{C}_1$, $\boldsymbol{C}_2$, solves these equations, applies the wavelet equations (2.20), and draws the wavelets. Figure 2.4 shows the quintic finite elements ϕ_0, ϕ_1, ϕ_2 and Figure 2.5 shows the corresponding wavelets w_0, w_1, w_2.

By construction, the finite elements are orthogonal to the wavelets and their translates. The two spaces are orthogonal complements in $V_1 = V_0 \oplus W_0$. This is invariant under dilation so $V_2 = V_1 \oplus W_1 = V_0 \oplus W_0 \oplus W_1$. Therefore the wavelets in W_0 are orthogonal to their dilates and translates in W_1. This is true at all scaling levels, $W_k \perp W_j$, except at the same level $k = j$. The wavelets are therefore called "semiorthogonal", or "pre-wavelets". They have $2r$ vanishing moments and $r-1$ continuous derivatives, with symmetry and support $[0, 3]$.

Other piecewise polynomial multiwavelet constructions can be found in [27, 28, 55, 56]

2.3 Condition of Orthogonality in Time Domain

Suppose the multi-scaling function $\phi(t)$ generates an orthogonal basis of V_0. This means that all $\phi_j(t)$ are orthogonal to all translates $\phi_l(t-k)$, $j, l = 0, \ldots, r-1$, $k \in \mathbf{Z}$

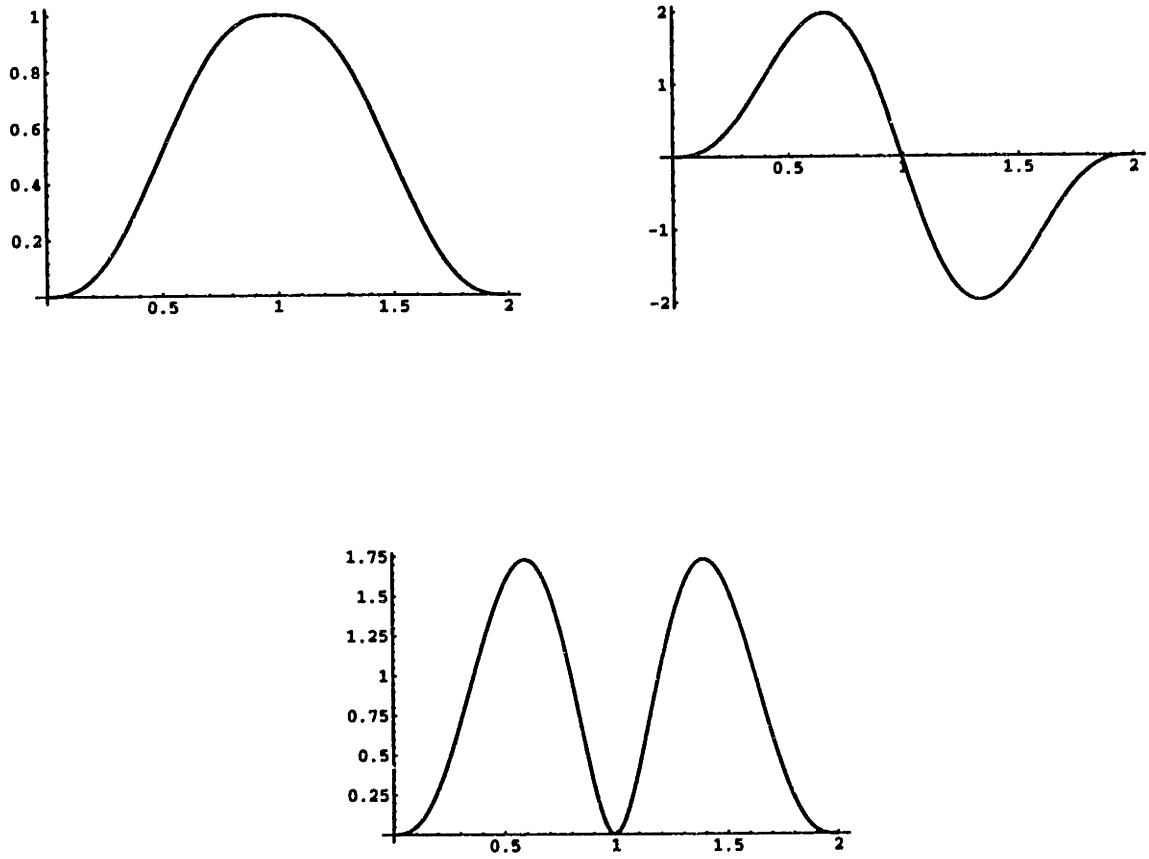


Figure 2.4: Quintic finite element scaling functions ϕ_0, ϕ_1, ϕ_2

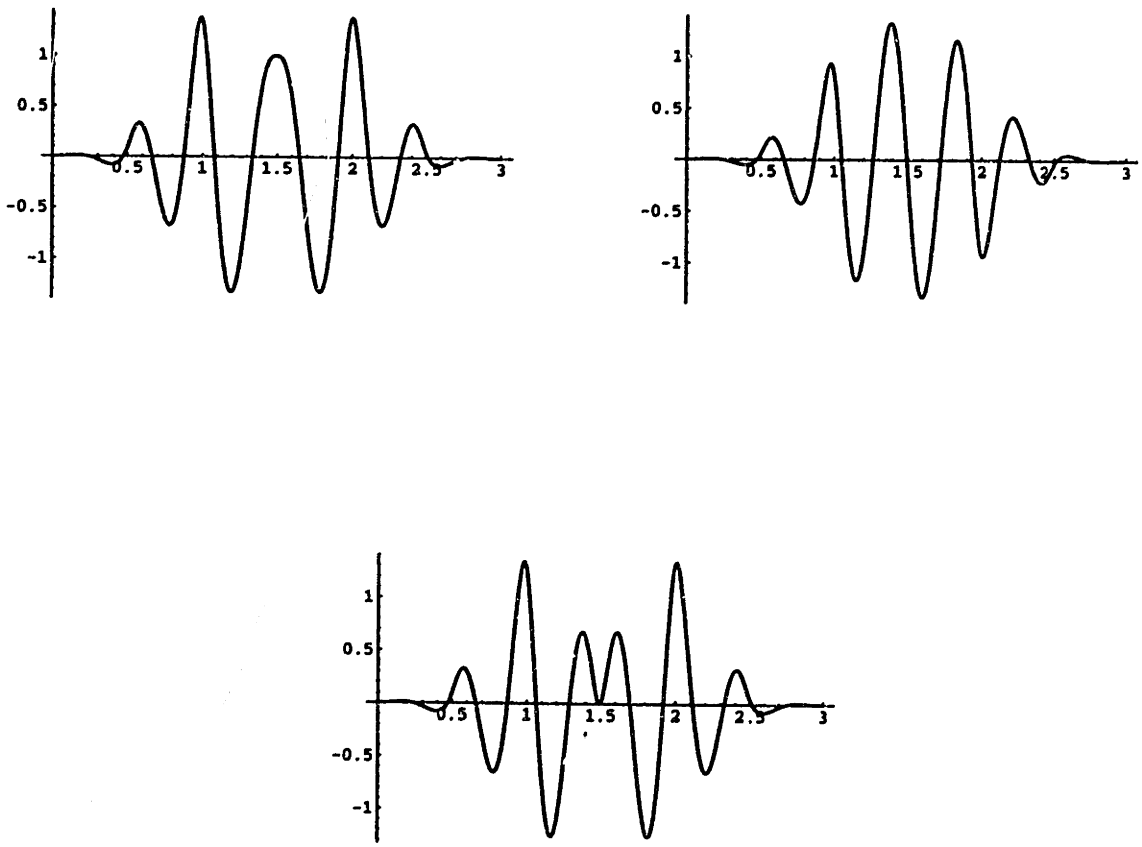


Figure 2.5: Quintic finite element wavelets w_0, w_1, w_2

and

$$\int \phi(t)\phi^T(t-l)dt = \delta_{0,l} \cdot I, \quad l \in \mathbf{Z}.$$

Substitute here the dilation equation for ϕ and change the variable of integration from t to $2t$:

$$\begin{aligned}
&\int \sum_k C_k \phi(2t-k) \sum_j \phi^T(2t-2l-j) C_j^T dt \\
&= \frac{1}{2} \sum_{kj} C_k (\int \phi(t-k)\phi^T(t-2l-j)) C_j^T dt \\
&= \frac{1}{2} \sum_{kj} C_k (\delta_{k,2l+j} \cdot I) C_j^T = \frac{1}{2} \sum_j C_{2l+j} C_j^T = \delta_{0,l} \cdot I. \qquad (2.27)
\end{aligned}$$

Equation (2.27) gives a necessary condition of orthogonality:

$$\sum_k C_k C_{2l+k}^T = 2\delta_{0,l} \cdot I, \quad k, l \in \mathbf{Z}. \qquad (2.28)$$

Not surprisingly it is just a condition of orthogonality for the matrix L. Moreover, (2.28) looks absolutely the same as in the scalar case [66].

Condition (2.28) also turns out to be sufficient if the corresponding cascade algorithm converges [65, 61]. In other words (2.28) is a necessary and sufficient condition of orthogonality if I is an eigenmatrix corresponding to the simple eigenvalue $\lambda = 1$ of the transition operator and absolute values of all its other eigenvalues are less than 1.

It is easy to check that the GHM coefficients satisfy condition (2.28).

Let us also say a few words about orthogonal multiwavelet $w(t) = [w_0(t) \ \ldots \ w_{r-1}(t)]^T$ corresponding to $\phi(t)$. In order to generate an orthogonal basis of W_0, the translates $w(t-l)$ should satisfy the following equation

$$\int w(t)w^T(t-l)dt = \delta_{0,l} \cdot I, \qquad l \in \mathbf{Z}. \qquad (2.29)$$

On the other hand, $V_{-1} = V_0 \oplus W_0$, and $W_0 \perp V_0$, so

$$\int \phi(t)w^T(t-l)dt = 0, \qquad l \in \mathbf{Z}. \qquad (2.30)$$

$w(t)$ satisfies wavelet equation (1.7), so (2.29) and (2.30) can be rewritten similarly to (2.27):

$$\begin{aligned}
&\textstyle\sum_k D_k D_{2l+k}^T = \delta_{0,l} \cdot I \\
&\textstyle\sum_k C_k D_{2l+k}^T = 0
\end{aligned} \qquad l \in \mathbf{Z}. \qquad (2.31)$$

These are conditions on matrices D_k producing orthogonal multiwavelets. Similarly to the conditions of approximation in time domain, they are not very useful for the

construction, because they give a large system of quadratic equations on the entries of the D's. However, for the GHM example, equations (2.31) can be solved explicitly [26, 69].

(2.28) in combination with (2.31) implies orthogonality of the doubly infinite block Toeplitz matrix

$$H_{\text{poly}} = \begin{bmatrix} \cdot & \cdot & \cdot & \cdot & & & & \\ & C_0 & C_1 & C_2 & C_3 & \cdot & \cdot & \cdot \\ & D_0 & D_1 & D_2 & D_3 & \cdot & \cdot & \cdot \\ & & & C_0 & C_1 & C_2 & C_3 & \cdot \\ & & & D_0 & D_1 & D_2 & D_3 & \cdot \\ & & & & & \cdot & \cdot & \cdot \end{bmatrix}.$$

This matrix is the time domain analog of the polyphase symbol in frequency domain, which will be used in Section 3.11 for the construction of orthogonal multiwavelet starting from a given orthogonal multi-scaling function.

Orthogonality and biorthogonality of multiwavelets will be considered in more detail in Chater 3.

Chapter 3

Frequency Domain Methods

As we have seen in Chapter 2, time domain methods are interesting and important from the theoretical point of view, but they do not open an efficient way to construct scaling functions and wavelets. Moreover, the experience of scalar wavelet theory suggests that Fourier methods could allow not only easy construction of scaling functions, but also would be convenient for investigation of the regularity, symmetry, orthogonality and biorthogonality of multiwavelets. Thus, we turn to the frequency domain and the Fourier transformation of the dilation equation (2.1):

$$\widehat{\boldsymbol{\phi}}(\omega) = \boldsymbol{P}\left(\frac{\omega}{2}\right)\widehat{\boldsymbol{\phi}}\left(\frac{\omega}{2}\right). \tag{3.1}$$

Here $\widehat{\boldsymbol{\phi}} = [\widehat{\phi}_0 \ \ldots \ \widehat{\phi}_{r-1}]^T$ and $\widehat{\phi}_k(\omega) = \int_{-\infty}^{\infty} \phi_k(t)\, e^{-i\omega t}\, dt$. $\boldsymbol{P}(\omega)$ is the refinement mask corresponding to $\boldsymbol{\phi}(t)$,

$$\boldsymbol{P}(\omega) := \frac{1}{2}\sum_{k=0}^{N} \boldsymbol{C}_k\, e^{-i\omega k}. \tag{3.2}$$

In the scalar case, $P(\omega)$ is a trigonometric polynomial. In the vector case, $\boldsymbol{P}(\omega)$ becomes a matrix trigonometric polynomial.

Iterating equation (3.1) we get

$$\widehat{\boldsymbol{\phi}}(\omega) = \boldsymbol{P}(\tfrac{\omega}{2})\boldsymbol{P}(\tfrac{\omega}{4})\widehat{\boldsymbol{\phi}}(\tfrac{\omega}{4}) = \ldots = \prod_{j=1}^{\infty} \boldsymbol{P}(\tfrac{\omega}{2^j})\widehat{\boldsymbol{\phi}}(0). \tag{3.3}$$

From now on (unless otherwise stated) we assume that $\boldsymbol{P}(0)$ has a simple eigenvalue $\lambda = 1$ and all its other eigenvalues are strictly less than 1 in absolute value. This condition ensures uniform convergence of the infinite product (3.3) (see [31, 15]).

3.1 Condition of Approximation in Fourier Domain

By recalling the formulas for the derivatives of $\boldsymbol{P}(\omega)$, the conditions of approximation (2.7) can be transferred from time to frequency domain.

Theorem 3.1.1 [32, 57] *Let $\boldsymbol{\phi}(t)$ be a refinable vector of compactly supported functions ϕ_k, $k = 0, \ldots, r-1$. Further, assume that integer translates $\phi_k(t-l)$, $l \in \mathbf{Z}$ are linearly independent. Then $\boldsymbol{\phi}(t)$ provides approximation order m if and only if the refinement mask $\boldsymbol{P}(\omega)$ of $\boldsymbol{\phi}$ satisfies the following conditions: There are vectors $\boldsymbol{u}_k \in \mathbf{R}^r$; $\boldsymbol{u}_0 \neq \mathbf{0}$ $(k = 0, \ldots, m-1)$ such that for $n = 0, \ldots, m-1$,*

$$\sum_{k=0}^{n} \binom{n}{k} \boldsymbol{u}_k (2i)^{k-n} (D^{n-k}\boldsymbol{P})(0) = 2^{-n} \boldsymbol{u}_n, \tag{3.4}$$

$$\sum_{k=0}^{n} \binom{n}{k} \boldsymbol{u}_k (2i)^{k-n} (D^{n-k}\boldsymbol{P})(\pi) = \mathbf{0}. \tag{3.5}$$

Here $\mathbf{0}$ denotes the zero vector.

If a matrix trigonometric polynomial $\boldsymbol{P}(\omega)$ satisfies (3.4) and (3.5) for $n = 0, \ldots, m-1$ with vectors $\boldsymbol{u}_0, \ldots, \boldsymbol{u}_{m-1}$ $(\boldsymbol{u}_0 \neq \mathbf{0})$, we say that $\boldsymbol{P}(\omega)$ provides approximation order m with $\boldsymbol{u}_0, \ldots, \boldsymbol{u}_{m-1}$.

In the scalar case ($r = 1$), when there is only one scaling function, the m-th approximation order implies m zeros of $P(\omega)$ at $\omega = \pi$ [19]. In the vector case, $\boldsymbol{P}(\omega)$ is a matrix. The situation becomes more complicated and we need conditions in *two* points, $\omega = 0$ and $\omega = \pi$. But still, for $r = 1$, equations (3.4), (3.5) can be simplified to

$$P(0) = 1; \qquad \mathrm{D}^n P(\pi) = 0, \quad n = 0, \ldots, m-1 \tag{3.6}$$

implying m zeros of $P(\omega)$ at $\omega = \pi$. Thus, Theorem 3.1.1 is a natural generalization of the scalar case.

Now the question is how to construct multiscaling functions with given approximation order. In the scalar case such constructions usually start from factoring out the zeros at $\omega = \pi$ from $P(\omega)$. We would like to get an analog of this procedure in the matrix case. On the other hand we need a tool to influence the approximation order of $\boldsymbol{P}(\omega)$.

Example 3.1.2 Suppose that a scalar trigonometric polynomial $P(\omega)$ has m zeros at $\omega = \pi$ and $P(0) = 1$:

$$P(\omega) = \left(\frac{1+e^{-i\omega}}{2}\right)^m P_0(\omega), \quad P_0(0) = 1.$$

Then the scaling function $\phi(t)$ corresponding to $P(\omega)$ has approximation order m [19]. Consider another symbol $\tilde{P}(\omega)$

$$\tilde{P}(\omega) = M^{-1}(2\omega)P(\omega)M(\omega) = \frac{M(\omega)}{M(2\omega)}P(\omega), \tag{3.7}$$

where $M(\omega)$ is a trigonometric polynomial. We are dealing with scalar polynomials and they commute! Clearly there are two principally different cases $M(0) \neq 0$ and $M(0) = 0$.

If $M(\omega) \neq 0$, then

$$\tilde{P}(\omega) = \left(\frac{1+e^{-i\omega}}{2}\right)^m \frac{M(\omega)}{M(2\omega)}P_0(\omega), \quad \tilde{P}(0) = P_0(0) = 1,$$

and $\tilde{P}(\omega)$ has the same approximation order as $P(\omega)$. The scaling function corresponding to $\tilde{P}(\omega)$ is $\hat{\psi}(\omega) = M(\omega)\hat{\phi}(\omega)$.

If $M(\omega)$ has a simple zero at $\omega = 0$, it can be written as $M(\omega) = (1-e^{-i\omega})m_0(\omega)$ with $m_0(0) \neq 0$ and

$$\begin{aligned}\tilde{P}(\omega) &= 2M^{-1}(2\omega)P(\omega)M(\omega) = 2\frac{(1-e^{-i\omega})m_0(\omega)}{(1-e^{-2i\omega})m_0(2\omega)}P(\omega) \\ &= \frac{1}{2^{m-1}}(1+e^{-i\omega})^{m-1}\frac{m_0(\omega)}{m_0(2\omega)}P_0(\omega)\end{aligned}$$

which means that $\tilde{P}(\omega)$ has approximation order $m-1$.

It turns out that transformations of type (3.7) play a very important role in the whole theory of multiwavelets, so we devote the next section to the investigation of their properties.

3.2 Two-scale Similarity Transform

$\boldsymbol{Q}(\omega)$ is a two-scale similarity transform (TST) of $\boldsymbol{P}(\omega)$ if

$$\boldsymbol{Q}(\omega) = \boldsymbol{M}(2\omega)\boldsymbol{P}(\omega)\boldsymbol{M}^{-1}(\omega). \tag{3.8}$$

$\widetilde{P}(\omega)$ is an inverse two-scale similarity transform (ITST) of $P(\omega)$ if

$$\widetilde{P}(\omega) = M^{-1}(2\omega)P(\omega)M(\omega). \tag{3.9}$$

We call $M(\omega)$ the transform matrix.

In this section we study how a TST changes the eigenvalues and eigenvectors of a matrix. The first result is obvious but useful.

Lemma 3.2.1 *Suppose $M(\omega)$ is invertible for all ω. Then TST (and the ITST) of $P(\omega)$ do not change the eigenvalues of $P(0)$.*

Proof. $Q(0) = M(0)P(0)M^{-1}(0)$ is a similarity transform of $P(0)$, so the eigenvalues of $Q(0)$ and $P(0)$ coincide. ∎

The situation changes when $M(0)$ is degenerate.

Lemma 3.2.2 *Suppose $M(\omega)$ is invertible for all ω except $\omega = 0$, $M(0)$ has a simple eigenvalue $\lambda_M(0) = 0$ corresponding to an eigenvector $r(0)$ and left eigenvector $l(0)$:*

$$M(0)r(0) = 0, \qquad l^T(0)M(0) = 0.$$

Moreover, $P(0)$ shares the eigenvector $r(0)$ with $M(0)$:

$$P(0)r(0) = \lambda_P(0)r(0),$$

and $\lambda_P(0)$ is a simple eigenvalue of $P(0)$. Then TST preserves all eigenvalues of $P(0)$ except $\lambda_P(0)$. This eigenvalue changes to $2\lambda_P(0)$ and has left eigenvector $l(0)$:

$$l^T(0)Q(0) = 2\lambda_P(0)l^T(0).$$

Proof. Consider Jordan factorizations of $M(\omega)$ and $P(\omega)$ near $\omega = 0$

$$M(\omega) = R_M(\omega)J_M(\omega)R_M^{-1}(\omega), \qquad P(\omega) = R_P(\omega)J_P(\omega)R_P^{-1}(\omega).$$

and a matrix $A(\omega) = R_M^{-1}(\omega)R_P(\omega)J_P(\omega)R_P^{-1}(\omega)R_M(\omega)$. As $M(0)$ and $P(0)$ share an eigenvector, the first column of both $R_M^{-1}(0)R_P(0)$ and $R_P^{-1}(0)R_M(0)$ is $[1\ 0\ \ldots\ 0]^T$. This means that

$$A(0) = R_M^{-1}(0)R_P(0)J_P(0)R_P^{-1}(0)R_M(0) = \begin{bmatrix} \lambda_P(0) & a_{01}(0) & \cdots & a_{0n-1}(0) \\ 0 & & & \\ \vdots & & \widetilde{A}(0) & \\ 0 & & & \end{bmatrix}$$

On the other hand $\widetilde{A}(0)$ is similar to $P(0)$, so the eigenvalues of $\widetilde{A}(0)$ are actually the same as $n-1$ eigenvalues of $P(0)$, which are not equal to $\lambda_P(0)$. Furthermore,

$$B(\omega) = J_M(2\omega)A(\omega)J_M^{-1}(\omega) = \begin{bmatrix} \frac{\lambda_M(2\omega)}{\lambda_M(\omega)}\lambda_P(\omega) & \lambda_M(2\omega)a_{01}(\omega) & \dots & \lambda_M(2\omega)a_{0n-1}(\omega) \\ \frac{a_{10}(\omega)}{\lambda_M(\omega)} & & & \\ \vdots & & \tilde{J}_M(\omega)\widetilde{A}(\omega)\tilde{J}_M^{-1}(\omega) & \\ \frac{a_{n-10}(\omega)}{\lambda_M(\omega)} & & & \end{bmatrix} \quad (3.10)$$

When ω goes to zero, we have

$$B(0) = \lim_{\omega\to 0} J_M(2\omega)A(\omega)J_M^{-1}(\omega) = \begin{bmatrix} 2\lambda_P(0) & 0 & \dots & 0 \\ b_{10}(0) & & & \\ \vdots & & \tilde{J}_M(0)\widetilde{A}(0)\tilde{J}_M^{-1}(0) & \\ b_{n-10}(0) & & & \end{bmatrix}.$$

The eigenvalues of $\tilde{J}_M(0)\widetilde{A}(0)\tilde{J}_M^{-1}(0)$ coincide with the eigenvalues of $\widetilde{A}(0)$, so the eigenvalues of $B(0)$ and $P(0)$ are the same with exception that $\lambda_P(0)$ of $P(0)$ is changed to $2\lambda_P(0)$ in $B(0)$.

Finally, $Q(0) = R_P(0)B(0)R_P^{-1}(0)$ is similar to $B(0)$. Hence it has the same eigenvalues and

$$l^T(0)Q(0) = l^T(0)R_P(0)B(0)R_P^{-1}(0) =$$
$$[1\ 0\ \dots\ 0]\begin{bmatrix} 2\lambda_P(0) & \dots & 0 \\ \vdots & \tilde{B}(0) & \\ b_{n-10}(0) & & \end{bmatrix} R_P^{-1}(0) = 2\lambda_P(0)l^T(0).$$

Thus the eigenvalue $2\lambda_P(0)$ of $Q(0)$ corresponds to the left eigenvector $l(0)$. ∎

Corollary 3.2.3 *If $M(0)$ is singular, then $Q(0)$ can have bounded components only if $P(0)$ and $M(0)$ share an eigenvector corresponding to the zero eigenvalue of $M(0)$.*

Proof. From (3.10) follows that $Q(0)$ can have bounded components only if

$$b_{j0} = \lim_{\omega\to 0}\frac{a_{j0}(\omega)}{\lambda_M(\omega)} < \infty, \qquad 0 \le j \le n-1.$$

But $\lambda_M(0) = 0$ so $a_{j0}(0)$ also must be zero for all $0 \le j \le n-1$, which means that $P(0)$ and $M(0)$ share an eigenvector corresponding to the zero eigenvalue of $M(0)$.

∎

Absolutely analogously can be proved a similar result for ITST:

Lemma 3.2.4 *Suppose $M(\omega)$ is invertible for all ω except $\omega = 0$, $M(0)$ has a simple eigenvalue $\lambda_M(0) = 0$ corresponding to an eigenvector $r(0)$ and left eigenvector $l(0)$:*

$$M(0)r(0) = 0, \qquad l^T(0)M(0) = 0.$$

Moreover, $P(0)$ shares the left eigenvector $l(0)$ with $M(0)$:

$$l^T(0)P(0) = \lambda_P(0)l^T(0).$$

Then ITST (3.9) preserves all eigenvalues of $P(0)$ except $\lambda_P(0)$. This eigenvalue changes to $\lambda_P(0)/2$ and corresponds to eigenvector $r(0)$:

$$\widetilde{P}(0)r(0) = \frac{1}{2}\lambda_P(0)r(0).$$

Corollary 3.2.5 *If $M(0)$ is singular, then $\widetilde{P}(0)$ can have bounded components only if $P(0)$ and $M(0)$ share a left eigenvector corresponding to the zero eigenvalue of $M(0)$.*

The Corollaries 3.2.3 and 3.2.5 give a necessary condition for existence of TST and ITST at $\omega = 0$. Let us also prove a sufficient condition.

Lemma 3.2.6 *Suppose the conditions of Lemma* 3.2.2 (*Lemma* 3.2.4) *are satisfied, and $D(\lambda_M)(0) \neq 0$. Then TST* (3.8) (*ITST* (3.9)) *has bounded components at $\omega = 0$.*

Proof. We will give the reasoning only for TST. As $M(\omega)$ is invertible for all $\omega \neq 0$, $Q(\omega)$ may be unbounded only at a single point $\omega = 0$. The difficulty in (3.10) could be only with $b_{j0} = \lim_{\omega\to 0} \frac{a_{j0}(\omega)}{\lambda_M(\omega)}$. Lemma 3.2.2 gives $\lambda_M(0) = 0$. Then $a_{j0}(0) = 0$, because $M(0)$ and $P(0)$ share the corresponding eigenvector. Use the rule of l'Hospital to compute the limit:

$$\lim_{\omega\to 0} \frac{a_{j0}(\omega)}{\lambda_M(\omega)} = \frac{\mathrm{D}(a_{j0})(0)}{\mathrm{D}(\lambda_M)(0)}.$$

$\mathrm{D}(\lambda_M)(0) \neq 0$ which means that $\lim_{\omega\to 0} \frac{a_{j0}(\omega)}{\lambda_M(\omega)} < \infty$ for all $j < n$. ∎

Remark 3.2.7 If $M(0)$ is degenerate and 2π periodic, then ITST with $M(\omega)$ is unbounded at $\omega = \pi$, unless $M(0)$ and $P(\pi)$ share an eigenvector in the sense of Lemma 3.2.4. Analog of Lemma 3.2.6 also holds for ITST at $\omega = \pi$.

3.3 Factorization of the Refinement Mask

Now, when we know the properties of our main tool — TST, we are ready to proceed with factoring out the approximation orders from $\boldsymbol{P}(\omega)$. This will give us an easy way to construct multi-scaling functions with given approximation order. In this section we find a whole class of factorizations of the symbol $\boldsymbol{P}(\omega)$. Our results generalize those achieved by Plonka in [57].

Suppose we are given a dilation equation in frequency domain (3.1), and symbol $\boldsymbol{P}(\omega)$ providing approximation order m, such that for $k = 0, \ldots, m-1$ relations (3.4),(3.5) are satisfied with some starting vectors $\boldsymbol{u}_0, \ldots, \boldsymbol{u}_{m-1}$

Example 3.1.2 suggests that in the matrix case, the approximation order of $\widetilde{\boldsymbol{P}}(\omega) = 2\boldsymbol{M}^{-1}(2\omega)\boldsymbol{P}(\omega)\boldsymbol{M}(\omega)$ with some singular $\boldsymbol{M}(\omega)$ is less by one than the approximation order of $\boldsymbol{P}(\omega)$. On the contrary, a TST with nonsingular $\boldsymbol{M}(\omega)$, preserves the approximation order. It turns out to be the case.

Theorem 3.3.1 *If the transform matrix $\boldsymbol{M}(\omega)$ is 2π periodic and invertible for all ω then both*

$$\boldsymbol{Q}(\omega) = \boldsymbol{M}(2\omega)\boldsymbol{P}(\omega)\boldsymbol{M}^{-1}(\omega) \quad \textit{and},$$

$$\widetilde{\boldsymbol{P}}(\omega) = \boldsymbol{M}^{-1}(2\omega)\boldsymbol{P}(\omega)\boldsymbol{M}(\omega) \tag{3.11}$$

have the same approximation order as $\boldsymbol{P}(\omega)$. The starting vectors for $\boldsymbol{Q}(\omega)$ and $\widetilde{\boldsymbol{P}}(\omega)$ are

$$\boldsymbol{q}_k = \sum_{l=0}^{k} \binom{k}{l} i^{l-k} \boldsymbol{u}_l (D^{k-l}\boldsymbol{M}^{-1})(0) \quad \boldsymbol{q}_0 \neq \boldsymbol{0}, \quad k = 0, \ldots, m-1,$$

$$\tilde{\boldsymbol{u}}_k = \sum_{l=0}^{k} \binom{k}{l} i^{l-k} \boldsymbol{u}_l (D^{k-l}\boldsymbol{M})(0), \quad \tilde{\boldsymbol{u}}_0 \neq \boldsymbol{0}, \quad k = 0, \ldots, m-1. \tag{3.12}$$

Proof. We give the proof only for $\widetilde{\boldsymbol{P}}(\omega)$. For $\boldsymbol{Q}(\omega)$ it is absolutely similar. Modify (3.11) to $\boldsymbol{P}(\omega)\boldsymbol{M}(\omega) = \boldsymbol{M}(2\omega)\widetilde{\boldsymbol{P}}(\omega)$ and differentiate this equality $k-j$ times:

$$(D^{k-j}\boldsymbol{P})(\omega)\boldsymbol{M}(\omega) = \boldsymbol{S}_1^{k-j}(\omega) - \boldsymbol{S}_2^{k-j}(\omega), \tag{3.13}$$

where

$$\boldsymbol{S}_1^{k-j}(\omega) = \sum_{l=0}^{k-j} \binom{k-j}{l} 2^{k-j-l} (D^{k-j-l}\boldsymbol{M})(2\omega)(D^l\widetilde{\boldsymbol{P}})(\omega) \tag{3.14}$$

$$\boldsymbol{S}_2^{k-j}(\omega) = \sum_{l=1}^{k-j} \binom{k-j}{l} (D^{k-j-l}\boldsymbol{P})(\omega)(D^l\boldsymbol{M})(\omega), \quad \boldsymbol{S}_2^0 = \boldsymbol{0}. \tag{3.15}$$

Suppose $\boldsymbol{P}(\omega)$ provides approximation order m. This means that (3.4), (3.5) hold for $k = 0, \ldots, m-1$ and a combination of (3.15) and (3.5) gives zero:

$$\sum_{j=0}^{k-1} \binom{k}{j} \boldsymbol{u}_j (2i)^{j-k} \boldsymbol{S}_2^{k-j}(\pi) =$$
$$\sum_{l=1}^{k} \binom{k}{l} \left(\sum_{j=0}^{k-l} \binom{k-l}{j} \boldsymbol{u}_j (2i)^{j-k+l} (D^{k-j-l}\boldsymbol{P})(\pi) \right) (2i)^{-l} (D^l \boldsymbol{M})(\pi) =$$
$$\sum_{l=1}^{k} \binom{k}{l} 0 (2i)^{-l} (D^l \boldsymbol{M})(\pi) = \boldsymbol{0}. \tag{3.16}$$

Analogously, by (3.15) and (3.4) we get

$$\sum_{j=0}^{k-1} \binom{k}{j} \boldsymbol{u}_j (2i)^{j-k} \boldsymbol{S}_2^{k-j}(0) =$$
$$\sum_{l=1}^{k} \binom{k}{l} \left(\sum_{j=0}^{k-l} \binom{k-l}{j} \boldsymbol{u}_j (2i)^{j-k+l} (D^{k-j-l}\boldsymbol{P})(0) \right) (2i)^{-l} (D^l \boldsymbol{M})(0) =$$
$$\sum_{l=1}^{k} \binom{k}{l} 2^{-k+l} (u_0^{k-l})^T (2i)^{-l} (D^l \boldsymbol{M})(0) =$$
$$2^{-k} \sum_{l=0}^{k-1} \binom{k}{l} i^{l-k} \boldsymbol{u}_l (D^{k-l}\boldsymbol{M})(0). \tag{3.17}$$

Recall the definition (3.12) of $\tilde{\boldsymbol{u}}_k$ for $k = 1, \ldots, m-1$ and substitute it into (3.17):

$$\sum_{j=0}^{k} \binom{k}{j} \boldsymbol{u}_j (2i)^{j-k} \boldsymbol{S}_2^{k-j}(0) = 2^{-k} k \tilde{\boldsymbol{u}}_{k-1} - 2^{-k} \boldsymbol{u}_k \boldsymbol{M}(0). \tag{3.18}$$

Now, multiply (3.5) by $\boldsymbol{M}(\pi)$, substitute (3.13) for $(D^{k-j}\boldsymbol{P})(\pi)\boldsymbol{M}(\pi)$ and (3.14) for $\boldsymbol{S}_k^{k-j}(\pi)$, and take into account (3.16) and (3.12):

$$\sum_{j=0}^{k} \binom{k}{j} \boldsymbol{u}_j (2i)^{j-k} (\boldsymbol{S}_1^{k-j}(\pi) - \boldsymbol{S}_2^{k-j}(\pi)) =$$
$$\sum_{j=0}^{k} \binom{k}{j} \boldsymbol{u}_j (2i)^{j-k} \sum_{l=0}^{k-j} \binom{k-j}{l} 2^{k-j-l} (D^{k-j-l}\boldsymbol{M})(0) (D^l \widetilde{\boldsymbol{P}})(\pi) =$$
$$\sum_{l=0}^{k} \binom{k}{l} \left(\sum_{j=0}^{k-l} \binom{k-l}{j} \boldsymbol{u}_j i^{j-k+l} (D^{k-j-l}\boldsymbol{M})(0) \right) (2i)^{-l} (D^l \widetilde{\boldsymbol{P}})(\pi) =$$
$$\sum_{l=0}^{k-1} \binom{k}{l} \tilde{\boldsymbol{u}}_{k-l} (2i)^{-l} D^l \widetilde{\boldsymbol{P}}(\pi) = \boldsymbol{0}, \tag{3.19}$$

which means that for $k = 0, \ldots, m-1$, (3.5) is satisfied for $\widetilde{P}(\omega)$ and starting vectors $\tilde{u}_0 \ldots \tilde{u}_{m-1}$.

Now check that condition (3.4) is valid for $P(\omega)$ with the same starting vectors. Multiply (3.4) by $M(0)$, substitute (3.13) for $(D^{k-j}P)(\pi)M(\pi)$ and take into account (3.18):

$$\sum_{j=0}^{k} \binom{k}{j} u_j (2i)^{j-k} S_1^{k-j}(0) = 2^{-k} u_k M(0) + \sum_{j=0}^{k} \binom{k}{j} u_j (2i)^{j-k} S_2^{k-j}(0) = 2^{-k} u_k M(0) + 2^{-k}\tilde{u}_k - 2^{-k} u_k M(0) = 2^{-k}\tilde{u}_0^k. \quad (3.20)$$

Analogously to (3.19)

$$\sum_{j=0}^{k} \binom{k}{j} u_j (2i)^{j-k} S_1^{k-j}(0) = \sum_{l=0}^{k} \binom{k}{l} \tilde{u}_{k-l} (2i)^{-l} D^l \widetilde{P}(\pi).$$

and by using (3.20) we get

$$\sum_{l=0}^{k} \binom{k}{l} (2i)^{-l} \tilde{u}_{k-l} (D^l \widetilde{P})(0) = 2^{-k}\tilde{u}_k.$$

Therefore, for $k = 0, \ldots, m-1$, (3.4) is also satisfied with the same starting vectors $\tilde{u}_0 \ldots \tilde{u}_{m-1}$.

Finally, $\tilde{u}_0 \neq 0$ because by (3.12), $\tilde{u}_0 = u_0 M(0) \neq 0$. ∎

Corollary 3.3.2 *If $M(0)$ is degenerate, but $\widetilde{P}(\omega)$ and $Q(\omega)$ are bounded, and $u_0 M(0) \neq 0$, then Theorem* 3.3.1 *still holds.*

Theorem 3.3.3 *Suppose $P(\omega)$ has approximation order $m \geq 1$ and* (3.4),(3.5) *are satisfied with starting vectors $u_0, \ldots u_{m-1}$. Let $M(\omega)$ be 2π periodic and invertible for all $\omega \neq 0$. Assume also that $M(0)$ has a simple eigenvalue $\lambda_M(0) = 0$ corresponding to the left eigenvector u_0 and $D(\lambda_M)(0)$.*

Then

$$\widetilde{P}(\omega) = 2M^{-1}(2\omega)P(\omega)M(\omega) \quad (3.21)$$

provides approximation order $m-1$ and conditions (3.4),(3.5) *are satisfied for $k = 0, \ldots, m-2$ with starting vectors $\tilde{u}_0, \ldots \tilde{u}_{m-2}$:*

$$\tilde{u}_k = \frac{1}{k+1} \sum_{j=0}^{k+1} \binom{k+1}{j} i^{j-k-1} u_j (D^{k+1-j} M)(0), \quad \tilde{u}_0 \neq 0. \quad (3.22)$$

Proof of Theorem 3.3.3 is similar to the proof of Theorem 3.3.1 and can be found in [71].

Remark 3.3.4 As in the scalar case we need 2 in the front of ITST in order to preserve the eigenvalue 1 in $\widetilde{P}(0)$.

Theorem 3.3.3 establishes a factorization of $P(\omega)$ similar to scalar factoring out the zeros at $\omega = \pi$.

Corollary 3.3.5 *If $P(\omega)$ has approximation order m, then it can be factored by TST's:*

$$\begin{aligned} P(\omega) = P_m(\omega) &= \frac{1}{2} M_m(2\omega) P_{m-1}(\omega) M_m^{-1}(\omega) \\ &= \frac{1}{4} M_m(2\omega) M_{m-1}(2\omega) P_{m-2}(\omega) M_{m-1}^{-1}(\omega) M_m^{-1}(\omega) \\ &= \ldots = \frac{1}{2^m} M_m(2\omega) \ldots M_1(2\omega) P_0(\omega) M_1^{-1}(\omega) \ldots M_m^{-1}(\omega), \end{aligned} \tag{3.23}$$

where $M_i(\omega)$ is invertible for all $\omega \neq 0$ and $M_i(0)$ has a simple eigenvalue $\lambda_{M_i}(0) = 0$, corresponding to the left eigenvector $l_i^T(0)$, such that $l_i^T(0) P_i(0) = l_i^T(0)$.

In the scalar case, (3.23) becomes the famous factorization of symbol $P(\omega)$ [19]

$$P(\omega) = \frac{1}{2^m}(1 - e^{-2i\omega})^m P_0(\omega)(1 - e^{-i\omega})^{-m} = \frac{1}{2^m}(1 + e^{-i\omega})^m P_0(\omega) \tag{3.24}$$

which ensures that $P(\omega)$ has approximation order m. In the matrix case the role of factors $(1 + e^{-i\omega})$ play TST's with transformation matrices $M_j(\omega)$.

For the construction of multi-scaling functions with given approximation order, the the key is the inverse of Theorem 3.3.3: the factorization (3.23) implies the approximation order m for $P(\omega)$. This statement is proved in [58].

Example 3.3.6 As an example we give a factorization (3.23) of the refinement mask

$$P(\omega) = \frac{1}{20} \begin{bmatrix} 6 + 6e^{-i\omega} & 8\sqrt{2} \\ (-1 + 9e^{-i\omega} + 9e^{-2i\omega} - e^{-3i\omega})/\sqrt{2} & -3 + 10e^{-i\omega} - 3e^{-2i\omega} \end{bmatrix},$$

corresponding to the multi-scaling pair constructed by Geronimo, Hardin and Massopust in [26]. This $P(\omega)$ has approximation order 2 with starting vectors $u_0 = [\sqrt{2}\ \ 1]$, $u_1 = [1/\sqrt{2}\ \ 1]$.

We choose $M_i(\omega)$ in the form $M_i(\omega) = I - e^{-i\omega} \frac{r_i l_i^T}{l_i^T r_i}$. $M_i(\omega)$ is non-degenerate for $\omega \neq 0$ and $M_i(0)$ has a simple eigenvalue $\lambda_{M_i}(0) = 0$ with left eigenvector l_i and right eigenvector r_i. Moreover, $\mathrm{D}(\lambda_{M_i})(0) = i \neq 0$. Thus, to satisfy the conditions

of Theorem 3.3.3 we need only to set $l_2^T = u_0 = [\sqrt{2}\ \ 1]$. We have free choice of r_2 and we put $r_2^T = [0\ \ 1]$. Then (3.21) gives:

$$\widetilde{P}(\omega) = P_1(\omega) = 2M_2^{-1}(2\omega)P(\omega)M_2(\omega) =$$
$$\frac{2}{1-e^{-2i\omega}}\begin{bmatrix} 1-e^{-2i\omega} & 0 \\ \sqrt{2}e^{-2i\omega} & 1 \end{bmatrix} P(\omega) \begin{bmatrix} 1 & 0 \\ -\sqrt{2}e^{-i\omega} & 1-e^{-i\omega} \end{bmatrix} =$$
$$\frac{1}{10}\begin{bmatrix} 6-10e^{-i\omega} & (1-e^{-i\omega})8\sqrt{2} \\ (-1+15e^{-i\omega})/\sqrt{2} & -3+13e^{-i\omega} \end{bmatrix}$$

It is easy to check that $P_1(\omega)$ has approximation order 1 with starting vector $\tilde{u}_0 = [1/\sqrt{2}\ \ 1]$. In correspondence with (3.22) we find $\tilde{u}_0 = \frac{1}{i}u_0 \mathrm{D}(M)(0) + u_1 M(0)$.

As $P_1(\omega)$ has approximation order 1, we can pull out one more "factor" based on $M_1(\omega)$. Now l_1 must be equal to $\tilde{u}_0 = [1/\sqrt{2}\ \ 1]$ and we again choose $r_1^T = [0\ \ 1]$. Then (3.21) gives

$$P_0(\omega) = 2M_1^{-1}(2\omega)P_1(\omega)M_1(\omega) =$$
$$\frac{2}{1-e^{-2i\omega}}\begin{bmatrix} 1-e^{-2i\omega} & 0 \\ \frac{e^{-2i\omega}}{\sqrt{2}} & 1 \end{bmatrix} P_1(\omega) \begin{bmatrix} 1 & 0 \\ -\frac{e^{-i\omega}}{\sqrt{2}} & 1-e^{-i\omega} \end{bmatrix} =$$
$$\frac{1}{5}\begin{bmatrix} 6-18e^{-i\omega}+8e^{-2i\omega} & (1-e^{-i\omega})^2 8\sqrt{2} \\ (-1+18e^{-i\omega}-8e^{-2i\omega})/\sqrt{2} & -3+16e^{-i\omega}-8e^{-2i\omega} \end{bmatrix}$$

$P_0(0)$ has approximation order 0, because

$$uP_0(0) = \frac{1}{5}[1/\sqrt{2}\ \ 1]\begin{bmatrix} -4 & 0 \\ \frac{9}{\sqrt{2}} & 5 \end{bmatrix} = [1/\sqrt{2}\ \ 1] = u, \quad \text{but}$$

$$uP_0(\pi) = \frac{1}{10}[1/\sqrt{2}\ \ 1]\begin{bmatrix} 32 & 32\sqrt{2} \\ -27\sqrt{2} & -27 \end{bmatrix} \neq 0.$$

We cannot continue and the final factorization (3.23) has the desired form:

$$P(\omega) = \frac{1}{4}M_2(2\omega)M_1(2\omega)P_0(\omega)M_1^{-1}(\omega)M_2^{-1}(\omega).$$

In the end of this section we prove one more Lemma which will be used in Section 3.8.

Lemma 3.3.7 *Suppose a symbol $P(\omega)$ provides approximation order m. Then,*

1. $\widetilde{P}(\omega) = \frac{1+e^{-i\omega}}{2}P(\omega)$ *provides approximation order* $m+1$.
2. $\widetilde{P}(\omega) = q(\omega)P(\omega)$, $q(0) = 1$, $q(\omega) \neq 0$, $\omega \in \mathbf{R}$, *provides approximation order* m.

Proof. 1. Approximation order m is achieved if and only if vector $\phi(t)$ satisfies Strang-Fix conditions [67]: there is a sequence of vectors a_l, $l \in \mathbf{Z}$ such that Fourier transform of

$$f(t) = \sum_{l \in \mathbf{Z}} a_l \phi(t-l)$$

has the following property

$$\hat{f}(0) \neq 0; \qquad D^n \hat{f}(2\pi l) = 0 \quad n = 0, \ldots, m-1. \tag{3.25}$$

Multiplication of $P(\omega)$ by $\frac{1+e^{-i\omega}}{2}$ leads to multiplication of $\hat{\phi}(\omega)$ by $\frac{1-e^{-i\omega}}{\omega}$ (see [65]):

$$\hat{\tilde{\phi}}(\omega) = \frac{1-e^{-i\omega}}{\omega} \hat{\phi}(\omega).$$

This means that $\tilde{f}(t) = \sum_{l \in \mathbf{Z}} a_l \tilde{\phi}(t-l)$ satisfies (3.25) for $n = 0, \ldots, m$ and $\widetilde{P}(\omega)$ provides approximation order $m+1$.

2. Consider $q_0(\omega) = \prod_{j=1}^{\infty} q(\omega/2)$. $q(\omega) < \infty$ and $q(0) = 1$, so this infinite product converges uniformly for all ω, $q_0(\omega) \neq 0$, $\omega \in \mathbf{R}$. Moreover, $q(\omega) = \frac{q_0(2\omega)}{q_0(\omega)}$, hence $\widetilde{P}(\omega) = q_0(2\omega) P(\omega) q_0^{-1}(\omega)$, and the new scaling function $\hat{\tilde{\phi}}(\omega) = q_0(\omega)\hat{\phi}(\omega)$ satisfies Strang-Fix conditions (3.25) with the same approximation order m. ∎

3.4 Estimate of Pointwise Regularity of Multi-scaling Functions

In this section we study how a TST influences the corresponding scaling function. This leads us to an estimate of pointwise regularity of $\phi(t)$.

Let $\phi(t)$ be a refinable vector. Its Fourier transform satisfies a dilation equation in frequency domain (3.1) with symbol $P(\omega)$. *What happens with* (3.1) *if we apply a TST to* $P(\omega)$*?* Let us again start with a scalar example.

Example 3.4.1 In Example 3.1.2 we mentioned that a scalar TST with invertible $M(\omega)$ changes the scaling function from $\hat{\phi}(\omega)$ to $\hat{\psi}(\omega) = M(\omega)\hat{\phi}(\omega)$. On the other hand [65] showed that the multiplication of a symbol $\tilde{P}(\omega)$ by $\frac{1}{2}\frac{1-e^{-2i\omega}}{1-e^{-i\omega}} = (1+e^{-i\omega})/2$ produces a factor $\frac{1-e^{-i\omega}}{\omega}$ in $\hat{\psi}(\omega)$. This means that if $P(\omega) = \frac{1}{2^m}(1+e^{-i\omega})^m P_0(\omega)$ than $\hat{\phi}(\omega) = \frac{1}{\omega^m}(1-e^{-i\omega})^m \hat{\phi}_0(\omega)$ and each order of approximation "adds" one more derivative to $\phi(t)$.

Now we are ready to start with the multi case. Assume that the transform matrix $M(\omega)$ is invertible for all ω. Apply a TST with $M(\omega)$ to $P(\omega)$. Then (3.1) still holds,

but for other scaling vector $\widehat{\psi}(\omega) = \boldsymbol{M}(\omega)\widehat{\phi}(\omega)$ and other symbol $\boldsymbol{Q}(\omega)$. Really, from (3.1) follows that

$$\boldsymbol{M}(\omega)\widehat{\phi}(\omega) = \boldsymbol{M}(\omega)\boldsymbol{P}(\omega/2)\boldsymbol{M}^{-1}(\omega/2)\boldsymbol{M}(\omega/2)\widehat{\phi}(\omega/2), \quad \text{or}$$

$$\widehat{\psi}(\omega) = \boldsymbol{Q}(\omega/2)\widehat{\psi}(\omega/2).$$

(Absolutely analogously to the scalar case!) By Lemma 3.2.1, $\boldsymbol{Q}(0)$ has a simple eigenvalue 1 with nonzero eigenvector $\widehat{\psi}(0) = \boldsymbol{M}(0)\widehat{\phi}(0)$.

As one could expect, the situation is different when $\boldsymbol{M}(0)$ is not invertible.

Lemma 3.4.2 *Suppose $\widehat{\phi}(\omega)$ satisfies* (3.1) *with $\boldsymbol{P}(\omega)$ providing approximation order at least* 1. *If $\boldsymbol{M}(\omega)$ is invertible for all $\omega \neq 0$ and $\boldsymbol{M}(0)$ has eigenvector $\widehat{\phi}(0)$ corresponding to a simple eigenvalue $\lambda_{\boldsymbol{M}}(0) = 0$, such that $D(\lambda_{\boldsymbol{M}})(0)$, then*

$$\widehat{\psi}(\omega) = \frac{1}{\omega}\boldsymbol{M}(\omega)\widehat{\phi}(\omega) \tag{3.26}$$

is the solution of the dilation equation

$$\widehat{\psi}(\omega) = \boldsymbol{Q}(\omega/2)\widehat{\psi}(\omega/2), \tag{3.27}$$

where

$$\boldsymbol{Q}(\omega) = \frac{1}{2}\boldsymbol{M}(2\omega)\boldsymbol{P}(\omega)\boldsymbol{M}^{-1}(\omega). \tag{3.28}$$

Moreover, $\widehat{\psi}(0) \neq \mathbf{0}$ and is bounded.

Proof. By our previous assumption, $\boldsymbol{P}(0)$ has a simple eigenvalue 1, corresponding to the eigenvector $\widehat{\phi}(0)$ (and all other eigenvalues of $\boldsymbol{P}(0)$ are less than 1). From Lemma 3.2.2 it follows that $\boldsymbol{Q}(0)$ also has a simple eigenvalue 1 and all its other eigenvalues are two times smaller then the eigenvalues of $\boldsymbol{P}(0)$ (hence, smaller than 1). This means (see [31, 15]) that the dilation equation (3.27) has a unique solution (up to multiplication by a constant).

The only thing we have to do is to check that $\widehat{\psi}(\omega) = \frac{1}{\omega}\boldsymbol{M}(\omega)\widehat{\phi}(\omega)$ satisfies (3.27). Substitute (3.26) and (3.28) into (3.27):

$$\frac{1}{\omega}\boldsymbol{M}(\omega)\widehat{\phi}(\omega) = \frac{1}{2}\boldsymbol{M}(\omega)\boldsymbol{P}(\omega/2)\frac{2}{\omega}\boldsymbol{M}^{-1}(\omega/2)\boldsymbol{M}(\omega/2)\widehat{\phi}(\omega/2). \tag{3.29}$$

By the assumption $\widehat{\phi}(\omega) = \boldsymbol{P}(\omega/2)\widehat{\phi}(\omega/2)$, both sides of (3.29) are equal for all ω.

Finally we mention that by the Corollary to Lemma 3.2.2, $\mathrm{D}(\lambda_{\boldsymbol{M}})(0) \neq 0$ means that $\boldsymbol{Q}(0)$ is bounded, so its eigenvector $\widehat{\psi}(0)$, corresponding to the eigenvalue 1 is also finite and $\widehat{\psi}(0) \neq 0$. ∎

In the previous section we proved that if a refinement mask $\boldsymbol{P}(\omega)$ provides approximation order m, then it can be factored in the form (3.23). Lemma 3.4.2 suggests that the Fourier transform of the scaling function $\widehat{\boldsymbol{\phi}}(\omega)$ corresponding to $\boldsymbol{P}(\omega)$ has the form

$$\widehat{\boldsymbol{\phi}}(\omega) = \frac{1}{\omega^m}\boldsymbol{M}_m(\omega)\dots\boldsymbol{M}_1(\omega)\widehat{\boldsymbol{\psi}}(\omega), \tag{3.30}$$

where $\widehat{\boldsymbol{\psi}}(\omega)$ is a scaling function corresponding to $\boldsymbol{P}_0(\omega)$. Or in other words $\boldsymbol{\phi}(t)$ has "m more derivatives than $\boldsymbol{\psi}(t)$". So, the smoothness of $\boldsymbol{\phi}(t)$ depends on the approximation order m and the smoothness of $\boldsymbol{\psi}(t)$.

Theorem 3.4.3 summarizes all said above and gives an estimate of regularity of $\boldsymbol{\phi}(t)$, analogous to corresponding estimates in the scalar case [19]. More advanced results are obtained in [15, 61] using different techniques.

Let us clear up the notation. We say that $\boldsymbol{\phi}(t) \in C^\alpha$ if for all components $\phi_i(t)$, $\int |\phi_i(\omega)|(1+|\omega|)^\alpha d\omega < \infty$. Following [31], by $||\cdot||$ we understand a matrix norm for which $||\boldsymbol{P}_0(0)|| = 1$. Such a norm can always be found as long as $\boldsymbol{P}_0(0)$ has non-degenerate eigenvalue 1 and absolute values of all other eigenvalues are less than 1.

Theorem 3.4.3 *Let $\boldsymbol{P}_0(\omega)$ be defined by the relation*

$$\boldsymbol{P}_0(\omega) = 2^m\boldsymbol{M}_1^{-1}(2\omega)\dots\boldsymbol{M}_m^{-1}(2\omega)\boldsymbol{P}(\omega)\boldsymbol{M}_m(\omega)\dots\boldsymbol{M}_1(\omega),$$

where $\boldsymbol{M}_1(\omega)\dots\boldsymbol{M}_m(\omega)$ are chosen as in (3.23). *Suppose $\boldsymbol{P}(\omega)$ has approximation order m, $\boldsymbol{P}(0)$ has a simple eigenvalue* 1 *and absolute values of all its other eigenvalues are less than 2^{-m+1}. Then $\boldsymbol{\phi}(t) \in C^\alpha$ if $\log_2 R = \log_2(\sup_{\omega\in R}||\boldsymbol{P}_0(\omega)||) < m-\alpha-1$.*

Proof. We chose $\boldsymbol{P}(\omega)$ such that $\boldsymbol{P}_0(0)$ has a simple eigenvalue 1 and absolute values of all others are less then 1. In [31] it was shown that in this case $||\prod_{j=1}^{\infty}\boldsymbol{P}_0(\omega/2^j)|| \le \text{const}(1+|\omega|)^{\log_2 R}$, so

$$||\widehat{\boldsymbol{\psi}}(\omega)|| = ||\prod_{j=1}^{\infty}\boldsymbol{P}_0(\omega/2^j)\widehat{\boldsymbol{\psi}}(0)|| \le \text{const}(1+|\omega|)^{\log_2 R} < \text{const}(1+|\omega|)^{m-\alpha-1}.$$

From (3.30) it follows for $|\omega| > 1$ that

$$\begin{aligned}||\widehat{\boldsymbol{\phi}}(\omega)|| &= \frac{1}{|\omega|^m}||\boldsymbol{M}_m(\omega)\dots\boldsymbol{M}_1(\omega)\widehat{\boldsymbol{\psi}}(\omega)|| \\ &< \text{const}\frac{1}{|\omega|^m}(1+|\omega|)^{m-\alpha-1} \le \text{const}(1+|\omega|)^{-\alpha-1},\end{aligned}$$

which means that $\boldsymbol{\phi}(t) \in C^\alpha$. ∎

Remark 3.4.4 $\phi(t) \in C^\alpha$ actually means that the component of $\phi(t)$ with the lowest regularity is in C^α. The transform $\widehat{\psi}(\omega) = \frac{1}{\omega} M(\omega)\widehat{\phi}(\omega)$ mixes all components and increases the smoothness of the worst. In principle it could happen, that at the same time the regularity of the most smooth component reduces.

Example 3.4.5 In Example 3.3.6 we found a factorization of the GHM symbol $P(\omega)$. But by Theorem 3.3.3 it is not unique. Another factorization is given in [57]:

$$P(\omega) = \frac{1}{40}\begin{bmatrix} \sqrt{2}/2 & -\sqrt{2}/2 \\ -e^{-2i\omega} & 1 \end{bmatrix}\begin{bmatrix} 1/2 & -1/2 \\ -e^{-2i\omega}/2 & 1/2 \end{bmatrix} \times$$
$$\begin{bmatrix} 10 & 0 \\ -1+20e^{-i\omega}-e^{-2i\omega} & -4-4e^{-i\omega} \end{bmatrix} \times$$
$$\begin{bmatrix} 1/2 & -1/2 \\ -e^{-i\omega}/2 & 1/2 \end{bmatrix}^{-1}\begin{bmatrix} \sqrt{2}/2 & -\sqrt{2}/2 \\ -e^{-i\omega} & 1 \end{bmatrix}^{-1},$$

One can see that here $m = 2$, $\sup_{\omega \in R} ||P_0(\omega)|| = 1$, so $0 < \alpha < 1$ and by Theorem 3.4.3 these scaling functions must be continuous.

Example 3.4.6 Another good example also comes from [57], a factorization of the symbol corresponding to the vector of quadratic B-splines with double knots (they are continuous but not differentiable). This system has approximation order $m = 3$, so one could expect that the scaling functions are differentiable. But it does not happen. The problem is that $P_0(\omega) = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}$ has an eigenvalue 2 which is too big, so Theorem 3.4.3 ensures only continuity.

One can also study L^2 regularity of the multi-scaling functions. Similarly to the the scalar case, it is governed by the eigenvalues of the transition operator (see [65, 61]).

3.5 Symmetry of Multi-scaling Functions

In this section we study properties of symmetric multi-scaling functions. We also find TSTs which preserve symmetry.

Suppose a scalar function $f(t)$ is symmetric about point T, then $\widehat{f}(\omega) = e^{-i\omega 2T}\widehat{f}(-\omega)$. Analogously, $f(t)$ is antisymmetric about T if and only if $\widehat{f}(\omega) = -e^{-i\omega 2T}\widehat{f}(-\omega)$. If $\phi(t)$ is a vector of r functions then all its components are symmetric or antisymmetric if

$$\widehat{\phi}(\omega) = E(\omega)\widehat{\phi}(-\omega), \quad E(\omega) = \mathrm{diag}(\pm e^{-i2T_0\omega}, \ldots, \pm e^{-i2T_{n-1}\omega}). \tag{3.31}$$

Here T_j is the point of symmetry of $\phi_j(t)$. $\boldsymbol{E}(\omega)$ is invertible for all ω.

We say that a vector $\boldsymbol{\phi}(t)$ is symmetric if $\widehat{\boldsymbol{\phi}}(\omega)$ satisfies (3.31) with some $\boldsymbol{E}(\omega)$.

Suppose $\boldsymbol{\phi}(t)$ is symmetric and refinable. Then $\widehat{\boldsymbol{\psi}}(\omega) = \widehat{\boldsymbol{\phi}}(-\omega) = \boldsymbol{E}^{-1}(\omega)\widehat{\boldsymbol{\phi}}(\omega)$ also satisfies a dilation equation:

$$\widehat{\boldsymbol{\psi}}(\omega) = \boldsymbol{Q}(\omega/2)\widehat{\boldsymbol{\psi}}(\omega/2) = \boldsymbol{P}(-\omega/2)\widehat{\boldsymbol{\psi}}(\omega/2).$$

As we have seen in Section 3.4, $\boldsymbol{Q}(\omega)$ must be a TST of $\boldsymbol{P}(\omega)$ with the transform matrix $\boldsymbol{E}^{-1}(\omega)$:

$$\boldsymbol{Q}(\omega) = \boldsymbol{P}(-\omega) = \boldsymbol{E}^{-1}(2\omega)\boldsymbol{P}(\omega)\boldsymbol{E}(\omega).$$

This gives us a condition on $\boldsymbol{P}(\omega)$.

Theorem 3.5.1 *Suppose $\boldsymbol{\phi}(t)$ is symmetric and refinable, $\widehat{\boldsymbol{\phi}}(\omega) = \boldsymbol{P}(\omega/2)\widehat{\boldsymbol{\phi}}(\omega/2)$, then*

$$\boldsymbol{P}(\omega) = \boldsymbol{E}(2\omega)\boldsymbol{P}(-\omega)\boldsymbol{E}^{-1}(\omega), \tag{3.32}$$

where $\boldsymbol{E}(\omega)$ is defined by (3.31).

Now let us find TSTs which preserve symmetry of a multi-scaling function. Suppose that both $\widehat{\boldsymbol{\phi}}(\omega)$ and $\widehat{\boldsymbol{\psi}}(\omega) = \boldsymbol{M}(\omega)\widehat{\boldsymbol{\phi}}(\omega)$ are symmetric and refinable:

$$\widehat{\boldsymbol{\phi}}(\omega) = \boldsymbol{P}(\omega/2)\widehat{\boldsymbol{\phi}}(\omega/2), \qquad \widehat{\boldsymbol{\psi}}(\omega) = \boldsymbol{Q}(\omega/2)\widehat{\boldsymbol{\psi}}(\omega/2).$$

Moreover let $\widehat{\boldsymbol{\psi}}(\omega) = \boldsymbol{M}(\omega)\widehat{\boldsymbol{\phi}}(\omega)$. This means that $\boldsymbol{Q}(\omega) = \boldsymbol{M}(2\omega)\boldsymbol{P}(\omega)\boldsymbol{M}^{-1}(\omega)$. Since $\widehat{\boldsymbol{\psi}}(\omega)$ is symmetric, (3.32) must hold for $\boldsymbol{Q}(\omega)$ with some matrix $\boldsymbol{E}(\omega)$:

$$\boldsymbol{Q}(\omega) = \boldsymbol{M}(2\omega)\boldsymbol{P}(\omega)\boldsymbol{M}^{-1}(\omega) = \boldsymbol{E}(2\omega)\boldsymbol{M}(-2\omega)\boldsymbol{P}(-\omega)\boldsymbol{M}^{-1}(-\omega)\boldsymbol{E}^{-1}(\omega). \tag{3.33}$$

But $\widehat{\boldsymbol{\phi}}(\omega)$ is also symmetric, so $\boldsymbol{P}(\omega) = \widetilde{\boldsymbol{E}}(2\omega)\boldsymbol{P}(-\omega)\widetilde{\boldsymbol{E}}^{-1}(\omega)$. Substitute this equation into (3.33):

$$\boldsymbol{M}(2\omega)\widetilde{\boldsymbol{E}}(2\omega)\boldsymbol{P}(-\omega)\widetilde{\boldsymbol{E}}^{-1}(\omega)\boldsymbol{M}^{-1}(\omega) = \boldsymbol{E}(2\omega)\boldsymbol{M}(-2\omega)\boldsymbol{P}(-\omega)\boldsymbol{M}^{-1}(-\omega)\boldsymbol{E}^{-1}(\omega).$$

This equality is satisfied if

$$\boldsymbol{M}(\omega) = \boldsymbol{E}(\omega)\boldsymbol{M}(-\omega)\widetilde{\boldsymbol{E}}^{-1}(\omega) \tag{3.34}$$

or in other words, $m_{kj}(\omega) = \pm m_{kj}(-\omega)e^{-i\omega(T_k - \widetilde{T}_j)}$.

Lemma 3.5.2 *Suppose $\boldsymbol{\phi}(t)$ and $\boldsymbol{\psi}(t)$ are both symmetric and refinable, and $\widehat{\boldsymbol{\psi}}(\omega) = \boldsymbol{M}(\omega)\widehat{\boldsymbol{\phi}}(\omega)$. Then $\boldsymbol{M}(\omega)$ satisfies* (3.34) *with*
$\boldsymbol{E}(\omega) = \mathrm{diag}(\pm e^{-i2T_0\omega}, \ldots, \pm e^{-i2T_{r-1}\omega})$ *and* $\widetilde{\boldsymbol{E}}(\omega) = \mathrm{diag}(\pm e^{-i2\widetilde{T}_0\omega}, \ldots, \pm e^{-i2\widetilde{T}_{r-1}\omega})$, *where $\widetilde{T}_j$ are points of symmetry of the components $\phi_j(t)$ and T_j are points of symmetry of the components $\psi_j(t)$ $j = 0, \ldots, r-1$.*

As we will see below, Lemma 3.5.2 is very useful in the construction of symmetric multi-scaling functions.

3.6 Construction of Multi-scaling Functions

Finally we have reached the point where we can show how to construct refinement masks which yield multi-scaling functions with desirable properties.

In the scalar case, there is no problem to find a mask providing any given order of accuracy. One can start with a trigonometric polynomial $P(\omega)$ such that $P(0) = 1$, and multiply it by a power of $\frac{1}{2}(1 + e^{-i\omega})$ (see e.g. [65]). In the vector case, a TST with transformation matrix $\boldsymbol{M}(\omega)$ plays the role of the factor $\frac{1}{2}(1 + e^{-i\omega})$.

An algorithm for the construction of multi-scaling functions with given approximation order can be obtained as a consequence of Corollary 3.3.5.

Algorithm 3.6.1 *Start with a matrix trigonometric polynomial $\boldsymbol{P}^{(n)}(\omega)$ providing approximation order n, such that $\rho(\boldsymbol{P}^{(n)}(0)) < 2$. Further, let $\boldsymbol{P}^{(n)}(0)$ possess an eigenvalue 1 with corresponding right eigenvector $\boldsymbol{r}_n$, i.e., $\boldsymbol{P}^{(n)}(0)\boldsymbol{r}_n = \boldsymbol{r}_n$.*

1. *Choose $\boldsymbol{M}_{\boldsymbol{r}_n}(\omega)$ such that:*
   a) $\det \boldsymbol{M}_{\boldsymbol{r}_n}(\omega) \neq 0$ *for* $\omega \neq 0$,
   b) $\mathrm{D}(\det \boldsymbol{M}_{\boldsymbol{r}_n})(0) \neq 0$,
   c) $\boldsymbol{M}_{\boldsymbol{r}_n}(0)\boldsymbol{r}_n = \boldsymbol{0}$.

2. *Construct the matrix $\boldsymbol{P}^{(n+1)}(\omega)$:*

$$\boldsymbol{P}^{(n+1)}(\omega) := \frac{1}{2}\boldsymbol{M}_{\boldsymbol{r}_n}(2\omega)\boldsymbol{P}^{(n)}(\omega)\boldsymbol{M}^{-1}_{\boldsymbol{r}_n}(\omega).$$

3. *Find a right eigenvector $\boldsymbol{r}_{n+1}$ corresponding to the eigenvalue 1 of $\boldsymbol{P}^{(n+1)}(0)$.*

4. *Repeat steps* **1, 2, 3** *as many times as needed.*

By Theorem 3.3.3, the approximation order of $\boldsymbol{P}^{(n+1)}(\omega)$ is $n+1$, and $m-n$ cycles of Algorithm 3.6.1 are needed to get a refinement mask $\boldsymbol{P}^{(m)}$ providing approximation order m. Lemma 3.2.2 shows that $\boldsymbol{P}^{(n+1)}(0)$ has eigenvalue 1, so step 4 is consistent.

One can see that there are two matrices to be chosen in Algorithm 3.6.1, the starting matrix $\boldsymbol{P}^{(n)}(\omega)$ (only once in the beginning) and the transformation matrix $\boldsymbol{M}_{\boldsymbol{r}_n}(\omega)$ (one on each cycle of the algorithm).

Theorem 3.4.3 shows that the regularity of the final function vector (determined by the refinement mask $\boldsymbol{P}^{(m)}(\omega)$) is governed by its approximation order m and by the properties of the starting matrix $\boldsymbol{P}^{(n)}(\omega)$, and each step of Algorithm 3.6.1 increases the number of derivatives by 1.

Now we would like to say a few words how to choose the transformation matrices $M_{r_n}(\omega)$.

In the scalar case, $M(\omega) = (1 - e^{-i\omega})$ is fixed. In the vector case, we are flexible in the choice of M_{r_k}. Actually, only one eigenvalue and one eigenvector are restricted in $M_{r_k}(\omega)$. We can use this freedom to obtain multi-scaling functions with desired properties.

Finite support. A refinement mask $P^{(n+1)}(\omega)$ corresponds to a finitely supported scaling vector if and only if all components of $P^{(n+1)}(\omega)$ are trigonometric polynomials (algebraic polynomials in $z = e^{-i\omega}$). But

$$P^{(n+1)}(\omega) = \frac{1}{2} M_{r_n}(2\omega) P^{(n)}(\omega) M_{r_n}^{-1}(\omega) \tag{3.35}$$

contains $M_{r_n}(2\omega)$ and $M_{r_n}^{-1}(\omega)$ which generally are not matrix trigonometric polynomials at the same time.

Lemma 3.6.2 *If $M_{r_n}(\omega)$ satisfies conditions* a–c *of Algorithm* 3.6.1, *$M_{r_n}(\omega)$ is a matrix trigonometric polynomial, and* det $M_{r_n}(\omega)$ *is linear in $z = e^{-i\omega}$, then the components of $P^{(n+1)}(\omega)$ in* (3.35) *are trigonometric polynomials.*

Proof: Let us use a well-known formula for an inverse matrix:

$$M_{r_n}^{-1}(\omega) = \frac{1}{\det M_{r_n}(\omega)} K_{r_n}(\omega). \tag{3.36}$$

Here $K_{r_n}(\omega)$ is the cofactor matrix. Its (k, j) element is the minor for the (j, k) element of $M_{r_n}(\omega)$ multiplied by $(-1)^{k+j}$ (see [64], page 225). In particular, $K_{r_n}(\omega)$ contains only trigonometric polynomials.

Since det $M_{r_n}(\omega)$ is linear in z, and det $M_{r_n}(0) = 0$, we have

$$\det M_{r_n}(\omega) = c_0(1 - e^{-i\omega}),$$

with a constant $c_0 \neq 0$, and according to (3.36),

$$P^{(n+1)}(\omega) = \frac{1}{2\,c_0\,(1 - e^{-i\omega})} M_{r_n}(2\omega) P^{(n)}(\omega) K_{r_n}(\omega). \tag{3.37}$$

It is easy to see that the components of $M_{r_n}(2\omega) P^{(n)}(\omega) K_{r_n}(\omega)$ are trigonometric polynomials. In Section 3.2, it was proved that $P^{(n+1)}(0)$ is bounded. On the other hand, $\frac{1}{1-e^{-i\omega}}$ is infinite at $\omega = 0$. Thus, all components of $M_{r_n}(2\omega) P^{(n)}(\omega) K_{r_n}(\omega)$ must possess a root at $\omega = 0$ or, in other words, must be divisible by $(1 - e^{-i\omega})$.

Hence, reducing $\boldsymbol{M}_{\boldsymbol{r}_n}(2\omega)\boldsymbol{P}^{(n)}(\omega)\boldsymbol{K}_{\boldsymbol{r}_n}(\omega)$ by $(1-e^{-i\omega})$ we get a matrix trigonometric polynomial $\boldsymbol{P}^{(n+1)}(\omega)$. ∎

Symmetry A reasonable way to get a symmetric multi-scaling functions with high approximation order is to start with $\boldsymbol{P}^{(n)}(\omega)$, yielding a symmetric function vector with low approximation order, and to preserve symmetry on each cycle of Algorithm 3.6.1. It is remarkable that after each cycle, the number of symmetric and antisymmetric components of the multi-scaling function changes, independent of the choice of $\boldsymbol{M}_{\boldsymbol{r}_n}$.

Lemma 3.6.3 *Suppose that $\boldsymbol{M}(\omega)$ satisfies the conditions* a–c *of Algorithm* 3.6.1, *and a* TST *with transformation matrix $\boldsymbol{M}(\omega)$ preserves the symmetry; i.e., $\boldsymbol{\phi} = (t)$ and $\tilde{\boldsymbol{\phi}}(t)$ are two symmetric multi-scaling functions connected by the relation:*

$$\hat{\boldsymbol{\phi}}(\omega) = \boldsymbol{M}(\omega)\hat{\tilde{\boldsymbol{\phi}}}(\omega).$$

Then the difference in the number of antisymmetric components in $\tilde{\boldsymbol{\phi}}$ and $\boldsymbol{\phi}$ is odd.

Proof: By Theorem 3.5.1 we have

$$\boldsymbol{M}(\omega) = \boldsymbol{E}(\omega)\boldsymbol{M}(-\omega)\widetilde{\boldsymbol{E}}^{-1}(\omega). \tag{3.38}$$

Since $\boldsymbol{M}(\omega)$ satisfies the conditions of Algorithm 3.6.1, det $\boldsymbol{M}(\omega)$ has a simple zero at $\omega = 0$, such that

$$f(e^{-i\omega}) = \det\, \boldsymbol{M}(\omega) = (1 - e^{-i\omega})f_0(e^{-i\omega}), \quad f_0(1) \neq 0. \tag{3.39}$$

From (3.38) and the definitions of $\boldsymbol{E}$ and $\widetilde{\boldsymbol{E}}$ (see Lemma 3.5.2), it follows that

$$\det\, \boldsymbol{M}(\omega) = f(e^{-i\omega}) = \det\, \boldsymbol{E}(\omega) \cdot \det\, \widetilde{\boldsymbol{E}}^{-1}(\omega) \cdot \det\, \boldsymbol{M}(-\omega) = e^{-2iT\omega}f(e^{i\omega})(-1)^N \tag{3.40}$$

where $T = \sum_{\nu=0}^{r-1}(T_\nu - \tilde{T}_\nu)$, and N is the difference in the number of antisymmetric functions in $\boldsymbol{\phi}$ and $\tilde{\boldsymbol{\phi}}$. Let $z = e^{-i\omega}$, then by (3.39)

$$f(z) = (1 - z)f_0(z)$$

and by (3.40)

$$f(z) = z^{2T}(-1)^N f(\frac{1}{z}) = z^{2T}\,(-1)^N\,(1 - \frac{1}{z})\,f_0(\frac{1}{z}).$$

Combining these two relations we find

$$(1-z)f_0(z) = -(-1)^N z^{2T-1}(1-z)f_0(\frac{1}{z})$$

and hence

$$f_0(z) = (-1)^{N+1} z^{2T-1} f_0(\frac{1}{z}). \tag{3.41}$$

But (3.41) implies that, if $N+1$ is odd, then $f_0(1) = 0$ and thus $\mathrm{D}(\det \boldsymbol{M})(0) = 0$, which contradicts the assumptions. So $N+1$ must be even and N must be odd. ∎

Example 3.6.4 In this example, we are going to increase the approximation order of the refinement mask $\boldsymbol{P}^{(2)}(\omega)$ corresponding to the Geronimo-Hardin-Massopust multi-scaling function $\boldsymbol{\phi} := [\phi_0\ \phi_1]^T$ (see Figure 1.5):

$$\boldsymbol{P}^{(2)}(\omega) = \frac{1}{20}\begin{bmatrix} 6+6e^{-i\omega} & 8\sqrt{2} \\ (-1+9e^{-i\omega}+9e^{-2i\omega}-e^{-3i\omega})/\sqrt{2} & -3+10e^{-i\omega}-3e^{-2i\omega} \end{bmatrix}. \tag{3.42}$$

$\phi_0(t)$, $\phi_1(t)$ are continuous, symmetric functions providing second order approximation. The integer translates $\phi_0(t-l)$, $\phi_1(t-l)$ $(l \in \mathbf{Z})$ are orthogonal. It is easy to see that a 1-eigenvector of $\boldsymbol{P}^{(2)}(0)$ is $\boldsymbol{r}_2 = [\sqrt{2}\ \ 1]^T$:

$$\boldsymbol{P}^{(2)}(0)\boldsymbol{r}_2 = \frac{1}{20}\begin{bmatrix} 12 & 8\sqrt{2} \\ 8\sqrt{2} & 4 \end{bmatrix}\begin{bmatrix} \sqrt{2} \\ 1 \end{bmatrix} = \begin{bmatrix} \sqrt{2} \\ 1 \end{bmatrix}.$$

Let us apply one cycle of Algorithm 3.6.1 to $\boldsymbol{P}^{(2)}(\omega)$ with transformation matrix $\boldsymbol{M}_{\boldsymbol{r}_2}(\omega)$ preserving symmetry and ensuring short support. Then, $\boldsymbol{M}_{\boldsymbol{r}_2}(\omega)$ must satisfy the assumptions of Lemma 3.6.2 and the following relation:

$$\boldsymbol{M}_{\boldsymbol{r}_2}(\omega) = \boldsymbol{E}(\omega)\boldsymbol{M}_{\boldsymbol{r}_2}(-\omega)\widetilde{\boldsymbol{E}}^{-1}(\omega) \tag{3.43}$$

(see Lemma 3.5.2). The first GHM scaling function is symmetric about $\tilde{T}_0 = 1/2$, and the second is symmetric about $\tilde{T}_1 = 1$, hence $\widetilde{\boldsymbol{E}}(\omega) = \mathrm{diag}\,(e^{-i\omega}, e^{-2i\omega})$. In order to get the supports of the new scaling functions as short as possible, we choose $T_0 = T_1 = 1$. Thus $\boldsymbol{E}(\omega) = \mathrm{diag}\,(e^{-2i\omega}, e^{-2i\omega})$. Let

$$\boldsymbol{M}_{\boldsymbol{r}_2}(\omega) := \begin{bmatrix} 1+e^{-i\omega} & -2\sqrt{2} \\ 1-e^{-i\omega} & 0 \end{bmatrix},$$

then (3.43) is satisfied. Moreover,

$$\boldsymbol{M}_{\boldsymbol{r}_2}(0)\boldsymbol{r}_2 = \begin{bmatrix} 2 & -2\sqrt{2} \\ 0 & 0 \end{bmatrix}\begin{bmatrix} \sqrt{2} \\ 1 \end{bmatrix} = \boldsymbol{0},$$

$\det \boldsymbol{M}_{\boldsymbol{r}_2}(\omega) = 2\sqrt{2}(1 - e^{-i\omega}) \neq 0$ for $\omega \neq 0$, $\mathrm{D}(\det \boldsymbol{M}_{\boldsymbol{r}_2})(0) = i2\sqrt{2} \neq 0$, so $\boldsymbol{M}_{\boldsymbol{r}_2}(\omega)$ satisfies all conditions of Algorithm 3.6.1. $\boldsymbol{M}_{\boldsymbol{r}_2}(\omega)$ is a matrix trigonometric polynomial and $\det \boldsymbol{M}_{\boldsymbol{r}_2}(\omega) = 2\sqrt{2}(1-e^{-i\omega})$ is linear in $z = e^{-i\omega}$, so by Lemma 3.6.2, finite support for the new scaling functions is ensured.

Now we perform step **3** of Algorithm 3.6.1 and compute $\boldsymbol{P}^{(3)}(\omega)$:

$$\begin{aligned} \boldsymbol{P}^{(3)}(\omega) &= \frac{1}{2}\boldsymbol{M}_{\boldsymbol{r}_2}(2\omega)\boldsymbol{P}^{(2)}(\omega)\boldsymbol{M}_{\boldsymbol{r}_2}^{-1}(\omega) \\ &= \frac{1}{40}\begin{bmatrix} -7 + 10e^{-i\omega} - 7e^{-2i\omega} & 15(1 - e^{-2i\omega}) \\ -4(1 - e^{-2i\omega}) & 10(1 + e^{-i\omega})^2 \end{bmatrix}. \end{aligned}$$

The resulting scaling functions are continuously differentiable and provide approximation order 3. They are plotted in Figure 3.1.

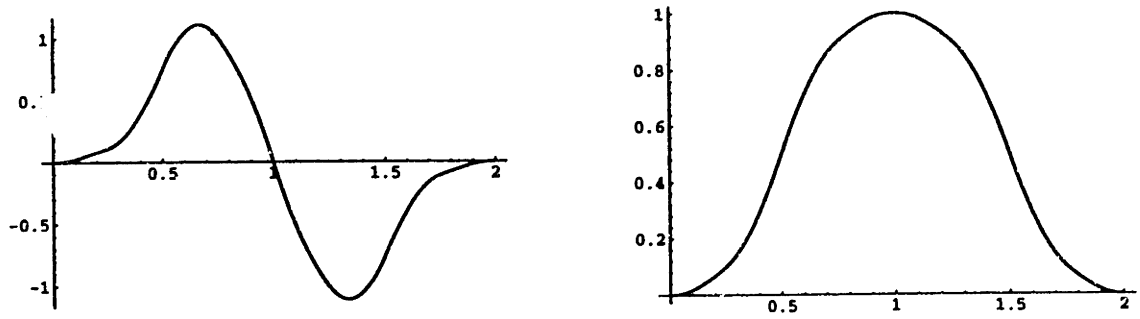


Figure 3.1: Symmetric multi-scaling function with approximation order 3

The mask $\boldsymbol{P}^{(3)}(\omega)$ corresponds to a dilation equation (2.1) with 3 matrix coefficients

$$\boldsymbol{P}_0 = \frac{1}{40}\begin{bmatrix} -7 & 15 \\ -4 & 10 \end{bmatrix}, \quad \boldsymbol{P}_1 = \frac{1}{40}\begin{bmatrix} 10 & 0 \\ 0 & 20 \end{bmatrix}, \quad \boldsymbol{P}_2 = \frac{1}{40}\begin{bmatrix} -7 & -15 \\ 4 & 10 \end{bmatrix}.$$

We mention that the GHM dilation equation has 4 coefficients since GHM functions ϕ_0, ϕ_1 have different supports.

Unfortunately, the new functions are not orthogonal. Observe that, in accordance with Lemma 3.6.3, one scaling function is symmetric and the other is antisymmetric. Moreover, the sum of the supports grows exactly by 1.

3.7 Orthogonalization of Multi-scaling Bases

Now we know how to construct multi-scaling functions with approximation, but still orthogonality is a problem. Suppose that the cascade algorithm converges in L^2 and $\phi(t)$ is not orthogonal but the integer translates $\phi_j(t-k)$, $j=0,\ldots,r-1$, $k \in \mathbf{Z}$ are linearly independent and form a basis in V_0. Can we somehow orthogonalize them and get a multi-scaling function $\psi(t)$ such that translates $\psi_j(t-k)$, $j=0,\ldots,r-1$, $k \in \mathbf{Z}$ form an orthogonal basis of V_0? To answer to this question again helps TST.

We start from the condition of orthogonality in frequency domain — analog of condition (2.28). Let $\phi(t)$ be refinable with a refinement mask $\boldsymbol{P}(\omega)$. We say that $\phi(t)$ is an orthogonal scaling function if all components $\phi_j(t)$ of $\phi(t)$ are orthogonal to the integer shifts of themselves and others:

$$\int \phi(t)\phi^*(t-k)dt = \delta(k)\cdot \boldsymbol{I}, \quad k \in \mathbf{Z}. \tag{3.44}$$

[69, 61] gave a condition of orthogonality of $\phi(t)$ in terms of $\boldsymbol{P}(\omega)$:

$$\boldsymbol{P}(\omega)\boldsymbol{P}^*(\omega) + \boldsymbol{P}(\omega+\pi)\boldsymbol{P}^*(\omega+\pi) = \boldsymbol{I}. \tag{3.45}$$

In the frequency domain, lefthandside of (3.44) becomes an autocorrelation matrix $\boldsymbol{A}(\omega) = \int \widehat{\phi}(\omega)\widehat{\phi}^*(\omega)d\omega$ which equals $\boldsymbol{I}$ if and only if $\phi(t)$ is orthogonal. [65, 61] showed that if the cascade algorithm converges to $\phi(t)$, then $\boldsymbol{A}(\omega)$ is the unique solution of equation

$$\boldsymbol{P}(\omega)\boldsymbol{A}(\omega)\boldsymbol{P}^*(\omega) + \boldsymbol{P}(\omega+\pi)\boldsymbol{A}(\omega+\pi)\boldsymbol{P}^*(\omega+\pi) = \boldsymbol{A}(2\omega). \tag{3.46}$$

If all components of $\phi(t)$ are linearly independent, $\boldsymbol{A}(\omega)$ is positive definite for all ω [61]. Let $\boldsymbol{M}(\omega)$ be 2π-periodic and invertible for all ω and assume that $\widehat{\psi}(\omega) = \boldsymbol{M}(\omega)\widehat{\phi}(\omega)$ is orthogonal. Then, as we saw in Section 3.4, $\widehat{\psi}(\omega) = \boldsymbol{Q}(\omega/2)\widehat{\psi}(\omega/2)$, $\boldsymbol{Q}(\omega) = \boldsymbol{M}(2\omega)\boldsymbol{P}(\omega)\boldsymbol{M}^{-1}(\omega)$. $\widehat{\psi}(\omega)$ is orthogonal and therefore $\boldsymbol{Q}(\omega)$ must satisfy orthogonality condition (3.45):

$$\begin{aligned}
&\boldsymbol{Q}(\omega)\boldsymbol{Q}^*(\omega) + \boldsymbol{Q}(\omega+\pi)\boldsymbol{Q}^*(\omega+\pi) = \\
&\quad \boldsymbol{M}(2\omega)\boldsymbol{P}(\omega)\boldsymbol{M}^{-1}(\omega)(\boldsymbol{M}^{-1}(\omega))^*\boldsymbol{P}^*(\omega)\boldsymbol{M}^*(2\omega) + \\
&\quad \boldsymbol{M}(2\omega+2\pi)\boldsymbol{P}(\omega+\pi)\boldsymbol{M}^{-1}(\omega+\pi)(\boldsymbol{M}^{-1}(\omega+\pi))^*\boldsymbol{P}^*(\omega+\pi)\boldsymbol{M}^*(2\omega+2\pi) = \boldsymbol{I}.
\end{aligned}$$

Take into account the periodicity of $\boldsymbol{M}(\omega)$ and multiply by $\boldsymbol{M}^{-1}(2\omega)$ from the left and by $(\boldsymbol{M}^{-1}(2\omega))^*$ from the right:

$$\begin{aligned}
\boldsymbol{P}(\omega)\boldsymbol{M}^{-1}(\omega)(\boldsymbol{M}^{-1}(\omega))^*\boldsymbol{P}^*(\omega) + \boldsymbol{P}(\omega+\pi)\boldsymbol{M}^{-1}(\omega+\pi)(\boldsymbol{M}^{-1}(\omega+\pi))^*\boldsymbol{P}^*(\omega+\pi) \\
= \boldsymbol{M}^{-1}(2\omega)(\boldsymbol{M}^{-1}(2\omega))^*.
\end{aligned}$$

This formula is actually (3.46) with $\boldsymbol{A}(\omega) = \boldsymbol{M}^{-1}(\omega)(\boldsymbol{M}^{-1}(\omega))^*$. As the cascade algorithm converges, the solution of (3.46) exists and is unique. Moreover, $\boldsymbol{A}(\omega)$ is positive definite because the translates $\phi_j(t-k)$ are linearly independent by the assumption. Thus there is an invertible square root of $\boldsymbol{A}(\omega)$ and the desired TST exists.

Now we can state an algorithm how to find the TST which orthogonalizes $\boldsymbol{\phi}(t)$.

Algorithm 3.7.1 1. *Starting from $\boldsymbol{\phi}(t)$ compute the autocorrelation matrix $\boldsymbol{A}(\omega) = \int \hat{\boldsymbol{\phi}}(\omega)\hat{\boldsymbol{\phi}}^*(\omega)d\omega$.*

2. *Find a square root $\widetilde{\boldsymbol{M}}(\omega)$ of $\boldsymbol{A}(\omega)$: $\boldsymbol{A}(\omega) = \widetilde{\boldsymbol{M}}(\omega)\widetilde{\boldsymbol{M}}^*(\omega)$. $\widetilde{\boldsymbol{M}}(\omega)$ exists and is invertible for all ω.*

3. *Compute the Fourier transform $\hat{\boldsymbol{\psi}}(\omega)$ of the orthogonalized scaling function:*

$$\hat{\boldsymbol{\psi}}(\omega) = \widetilde{\boldsymbol{M}}^{-1}(\omega)\hat{\boldsymbol{\phi}}(\omega) = \boldsymbol{M}(\omega)\hat{\boldsymbol{\phi}}(\omega).$$

It satisfies a dilation equation

$$\hat{\boldsymbol{\psi}}(\omega) = \boldsymbol{Q}(\omega/2)\hat{\boldsymbol{\psi}}(\omega/2) = \boldsymbol{M}(\omega)\hat{\boldsymbol{\psi}}(\omega/2)\boldsymbol{M}^{-1}(\omega/2)\hat{\boldsymbol{\psi}}(\omega/2).$$

4. *Compute $\boldsymbol{\psi}(t)$ using inverse Fourier transform.*

Remark 3.7.2 Generally, $\boldsymbol{M}(\omega) = \widetilde{\boldsymbol{M}}^{-1}(\omega)$ is a matrix trigonometric polynomial with an infinite number of terms, which means that $\boldsymbol{\psi}(t)$ has infinite support.

Remark 3.7.3 From the reasoning above, if a TST preserves the orthogonality of a scaling function, then the transform matrix must be orthogonal for all ω: $\boldsymbol{M}(\omega)\boldsymbol{M}^*(\omega) = \boldsymbol{I}$.

One can see that all results of this section are absolutely consistent with the corresponding results in the scalar case.

3.8 Dual Multi-scaling Functions

As we so above, orthogonality is very restrictive and the attempts to orthogonalize translates $\boldsymbol{\phi}(t-k)$ lead to the infinite support of the new scaling function. In the scalar case, orthogonality cannot be combined with finite support, symmetry and approximation order higher than 1 [68]. In the case of multiwavelets, one deals

with several scaling functions which gives much more freedom. Finitely supported, orthogonal multi-scaling functions with high approximation order can be symmetric (see Figure 1.5, [26, 23]). But still, orthogonality causes many problems. In order to avoid them some of the conditions must be relaxed. The usual way is to substitute one orthogonal basis by two biorthogonal ones: first for the decomposition and second for the reconstruction. Such bases still provide perfect reconstruction and can combine symmetry, orthogonality and high approximation order [19]. Besides it, biorthogonal bases are desirable in many applications.

An effective algorithm for the construction of multi-scaling functions with symmetry and approximation was described in Section 3.6. Now we are going to present an algorithm for the construction of dual multi-scaling functions.

Biorthogonality of multiwavelets was already approached in [1, 17, 39, 51] but no systematic procedure for the construction of duals were given. The advantage of our method is that it gives a transparent way to get the dual basis with the same symmetry and approximation properties as the direct one.

In this section we slightly change the notation and denote by $\boldsymbol{H}_0(z)$ the symbol corresponding to the analysis multi-scaling function, and by $\boldsymbol{F}_0(z)$ the symbol corresponding to the synthesis (dual) multi-scaling function. Usually it is more convenient to substitute $e^{-i\omega}$ by z, so from now on, unless otherwise stated, we will work in $z = e^{-i\omega}$ domain. In the scalar case $\boldsymbol{H}_0(z)$ is a polynomial. In the "multi" case $H_0(z)$ is a matrix polynomial.

Suppose we are given a multi-scaling function $\phi(t) = [\phi_0(t) \dots \phi_{r-1}(t)]^T$ such that

$$\widehat{\phi}(\omega) = \boldsymbol{H}_0(\omega/2)\widehat{\phi}(\omega/2). \tag{3.47}$$

$\boldsymbol{H}_0$ can be considered as a low-pass analysis filter. The goal of the following two sections is to complete it to a perfect reconstruction filter bank. The natural first step in this direction is to find the dual synthesis low-pass filter $\boldsymbol{F}_0$, such that the biorthogonality condition is satisfied [68, 61]:

$$\boldsymbol{H}_0(z)\boldsymbol{F}_0^*(z) + \boldsymbol{H}_0(-z)\boldsymbol{F}_0^*(-z) = c\boldsymbol{I}, \quad c = \text{const.} \tag{3.48}$$

In order to do it we develop cofactor method.

Consider a cofactor matrix $\boldsymbol{K}_{\boldsymbol{H}_0}(z)$ corresponding to $\boldsymbol{H}_0(z)$ ((j,k) element of $\boldsymbol{K}_{\boldsymbol{H}_0}$ is the minor complement to (k,j) element of $\boldsymbol{H}_0$ multiplied by $(-1)^{k+j}$). Then, by the properties of $\boldsymbol{K}_{\boldsymbol{H}_0}$ (see [64])

$$\boldsymbol{H}_0(z)\boldsymbol{K}_{\boldsymbol{H}_0}(z) = \boldsymbol{K}_{\boldsymbol{H}_0}(z)\boldsymbol{H}_0(z) = \det\boldsymbol{H}_0(z) \cdot \boldsymbol{I}.$$

This means that substitution of $K_{H_0}(z)$ into (3.48) reduces it to a scalar equation:

$$\det H_0(z)\Delta_0^*(z) + \det H_0(-z)\Delta_0^*(-z) = c, \quad c = \text{const.} \tag{3.49}$$

Here, both $\det H_0(z)$ and $\Delta_0(z)$ are scalar polynomials instead of matrix polynomials $H_0(z)$ and $F_0(z)$ in (3.48).

Solution of (3.49) is well studied, and in particular it is proved that if $\det H_0(z)$ and $\det H_0(-z)$ do not have common roots, then $\Delta_0(z)$ can always be found (see for example [68]). Thus, we can formulate an algorithm for the construction of a dual symbol.

Algorithm 3.8.1 (Cofactor Method) *Given matrix symbol $H_0(z)$ such that $\det H_0(z)$ is not divisible by $z^2 - \lambda^2$ for any λ and the smallest by absolute value eigenvalue of $H_0(1)$ is simple*

1. *Compute Cofactor matrix $K_{H_0}(z)$.*
2. *Find the largest by absolute value eigenvalue λ_0 of $K_{H_0}(1)$.*
3. *Find a polynomial $\Delta_0(z)$ such that $\Delta_0(1) = 1$ and (3.49) is satisfied.*
4. *Compute dual symbol $F_0(z) = \frac{1}{\lambda_0^*}\Delta_0(z)K^*_{H_0}(z)$.*

Remark 3.8.2 Mention rescaling step **2.** It is crucial for the approximation properties of $F_0(z)$ and trivial in the scalar case. It also determines constant c in (3.48), (3.49).

Algorithm 3.8.1 produces only a particular dual symbol. Other, more general solutions of (3.48) are possible. However, cofactor method ensures some nice properties of $F_0(z)$ such as approximation and symmetry.

Theorem 3.8.3 *Suppose matrix symbol $H_0(z)$ has approximation order m, $\det H_0(z)$ is not divisible by $z^2 - \lambda^2$ for any $\lambda \neq 0$ and the smallest by absolute value eigenvalue $\lambda_0 \neq 0$ of $H_0(1)$ is simple. Then, dual symbol $F_0(z) = \frac{1}{\lambda_0^*}\Delta_0(z)K^*_{H_0}(z)$ provides approximation order not less than m.*

Before proving this theorem we would like to discuss a particular case.

Example 3.8.4 Suppose multi-scaling function $\phi(t) = [\phi_0(t) \;\; \phi_1(t)]^T$ has only two components and $H_0(z) = \begin{bmatrix} h_{00}(z) & h_{01}(z) \\ h_{10}(z) & h_{11}(z) \end{bmatrix}$. Here $h_{00}(z), \ldots, h_{11}(z)$ are scalar polynomials. Then $K^*_{H_0}(z) = \begin{bmatrix} h_{11}^*(z) & -h_{10}^*(z) \\ -h_{01}^*(z) & h_{00}^*(z) \end{bmatrix}$. We assume that $\phi(t) \neq 0 \in L^2$,

and the largest eigenvalue of $\boldsymbol{H}_0(1)$ is $\lambda_0 = 1$. One can mention that $\boldsymbol{K}^*_{\boldsymbol{H}_0}(z) = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \boldsymbol{H}_0(\overline{z}) \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$, so the largest eigenvalue of $\boldsymbol{K}^*_{\boldsymbol{H}_0}(1)$ is $\lambda^*_0 = 1$ and no rescaling is needed. On the other hand, Theorem 3.3.1 proves that similarity transform does not change the approximation order of $\boldsymbol{H}_0$. (Actually the multi-scaling function corresponding to $\boldsymbol{K}^*_{\boldsymbol{H}_0}(z)$ is $\boldsymbol{\psi}(t) = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \boldsymbol{\phi}(t) = [\phi_1(-t) \quad -\phi_0(-t)]^T$. Obviously $\boldsymbol{\psi}(t)$ and $\boldsymbol{\phi}(t)$ have the same approximation order.) Finally, according to part **2** of Lemma 3.3.7, multiplication by a scalar polynomial $\Delta_0(z)$ does not change the approximation order as well, so $\boldsymbol{F}_0(z) = \Delta_0(z)\boldsymbol{K}^*_{\boldsymbol{H}_0}(z)$ has the same approximation order as $\boldsymbol{H}_0(z)$.

Proof of Theorem 3.8.3 Assume that $\boldsymbol{H}_0(z)$ provides approximation order $m \geq 1$. Corollary 3.3.5 proves that

$$\boldsymbol{H}_0(z) = \frac{1}{2^m}\boldsymbol{M}_m(z^2)\dots\boldsymbol{M}_1(z^2)\widetilde{\boldsymbol{H}}_0(z)\boldsymbol{M}_1^{-1}(z)\dots\boldsymbol{M}_m^{-1}(z),$$

where $\widetilde{\boldsymbol{H}}_0(z)$ has approximation order 0, $\boldsymbol{M}_j(1)$ $j = 1,\dots,m$ are singular and have special 0-eigenvector. Moreover, $\det\boldsymbol{M}_j(z) = c_j(1-z)$, so $\det\boldsymbol{H}_0(z) = \frac{(1+z)^m}{2^{mr}}\det\widetilde{\boldsymbol{H}}_0(z)$.

By the properties of a cofactor matrix

$$\boldsymbol{K}^*_{\boldsymbol{H}_0}(z) = (\det\boldsymbol{H}_0(z)\cdot\boldsymbol{H}_0^{-1}(z))^* = 2^{-m(r-1)}(1+\overline{z})^m\det\widetilde{\boldsymbol{H}}_0^*(z)\times$$
$$(\boldsymbol{M}_m^{-1}(z^2))^*\dots(\boldsymbol{M}_1^{-1}(z^2))^*(\widetilde{\boldsymbol{H}}_0^{-1}(z))^*(\boldsymbol{M}_1(z))^*\dots(\boldsymbol{M}_m(z))^*. \qquad (3.50)$$

We assume that the initial multi-scaling function $\boldsymbol{\phi}(t)$ is in L^2 and it is the unique solution of (3.47). This means that the eigenvalues of $\boldsymbol{H}_0(1)$ are $1, \lambda_1, \dots, \lambda_{r-1}$ $0 < |\lambda_j| < 1$, $j = 1,\dots,n-1$, $\det\boldsymbol{H}_0(1) = \prod_{j=1}^{r-1}\lambda_j$. Eigenvalues of $\boldsymbol{K}^*_{\boldsymbol{H}_0}(1)$ are $\prod_{j=1}^{r-1}\lambda_j^*, \frac{1}{\lambda_1^*}\prod_{j=1}^{r-1}\lambda_j^*, \dots, \frac{1}{\lambda_{r-1}^*}\prod_{j=1}^{r-1}\lambda_j^*$. Suppose that $|\lambda_k|$ is the smallest, $|\lambda_k| < |\lambda_j|$, $j = 1,\dots,r-1$. Then $\lambda_0^* = \frac{1}{\lambda_k^*}\prod_{j=1}^{r-1}\lambda_j^*$ will have the largest absolute value among all eigenvalues of $\boldsymbol{K}^*_{\boldsymbol{H}_0}(1)$. This is the rescaling factor from the Algorithm 3.8.1.

By the results of Section 3.2 $\widetilde{\boldsymbol{H}}_0(1)$ has eigenvalues $1, 2^m\lambda_1, \dots, 2^m\lambda_{r-1}$ so ${}_0\widetilde{\boldsymbol{H}}_0(1) = \frac{1}{\lambda_0^*}2^{-m(r-2)}\det\widetilde{\boldsymbol{H}}_0^*(1)(\widetilde{\boldsymbol{H}}_0^{-1}(1))^*$ has eigenvalues $\lambda_k^*2^m, \frac{\lambda_k^*}{\lambda_1^*}, \dots, 1, \dots, \frac{\lambda_k^*}{\lambda_{r-1}^*}$.

As we mentioned before, $\widetilde{\boldsymbol{H}}_0(z)$ has approximation order 0, which means that there exists a vector $\boldsymbol{l} \neq \boldsymbol{0}$ such that $\boldsymbol{l}^T\widetilde{\boldsymbol{H}}_0(1) = \boldsymbol{l}^T$, $\boldsymbol{l}^T\widetilde{\boldsymbol{H}}_0(-1) \neq \boldsymbol{0}$ [32, 57]. ${}_0\widetilde{\boldsymbol{H}}_0(1)$ also has eigenvalue 1, so there exists a vector $\boldsymbol{l}_1 \neq \boldsymbol{0}$ such that $\boldsymbol{l}_1^T{}_0\widetilde{\boldsymbol{H}}_0(1) = \boldsymbol{l}_1^T$. If ${}_0\widetilde{\boldsymbol{H}}_0(-1)$ is not degenerate, then $\boldsymbol{l}_1^T{}_0\widetilde{\boldsymbol{H}}_0(-1) \neq \boldsymbol{0}$ and ${}_0\widetilde{\boldsymbol{H}}_0(z)$ has approximation order 0. But if ${}_0\widetilde{\boldsymbol{H}}_0(-1)$ is degenerate, then it could happen that $\boldsymbol{l}_1^T{}_0\widetilde{\boldsymbol{H}}_0(-1) = \boldsymbol{0}$ and ${}_0\widetilde{\boldsymbol{H}}_0(z)$ has approximation order higher than 0.

Consider matrix ${}_1\widetilde{H}_0(z) = (M_1^{-1}(z^2))^* \frac{1+\bar{z}}{2} {}_0\widetilde{H}_0(z)(M_1(z))^*$. Following the construction in Section 3.6 $(M_1(1))^*$ and ${}_0\widetilde{H}_0(1)$ have a common left eigenvector l_2^T, $l_2^T(M_1(1))^* = 0$, $l_2^T {}_0\widetilde{H}_0(1) = \lambda_K^* 2^m$. This means that ${}_1\widetilde{H}_0(z)$ is finite at $z = 1$, although $M_1(1)$ is singular (see Section 3.2). The degenerate eigenvalue of $M_1(z)$ behaves as $(1-z)^{-1}$ near $z = 1$, so $(1+\bar{z})(M_1^{-1}(z^2))^*$ will also be finite at $z = -1$. Hence ${}_1\widetilde{H}_0(z)$ is well defined for all $z = e^{-i\omega}$. Corollary 3.3.2 shows that if $M^{-1}(z^2)P(z)M(z)$ is finite for all $z = e^{-i\omega}$, and the common left eigenvector l^T, $l^T M(1) = 0$ corresponds to an eigenvalue $\lambda \neq 1$ of $P(1)$ then, $M^{-1}(z^2)P(z)M(z)$ has the same approximation order as $P(z)$. This means that ${}_1\widetilde{H}_0(z)$ has the same approximation order as $\frac{1+\bar{z}}{2} {}_0\widetilde{H}_0(z)$ which, according to the previous paragraph and part 1 of Lemma 3.3.7 is not less than 1. Repeating this reasoning m times one can see that ${}_m\widetilde{H}_0(z) = \frac{1}{\lambda_0^*} K_{H_0}^*(z)$ has approximation order not less than m.

The only thing we have to do now, is to apply part 2 of Lemma 3.3.7 one more time and mention that multiplication by $\Delta_0(z)$ does not change the approximation order of $\frac{1}{\lambda_0^*} K_{H_0}^*(z)$. The proof is complete. ∎

Remark 3.8.5 Theorem 3.8.3 does not hold for the scalar case. For $r = 1$, $K_{H_0}(z) = 1$ does not give any approximation at all. This happens because in the scalar case no rescaling can be done, thus no approximation can be moved from one eigenvalue of $H_0(1)$ to another (there is only one eigenvalue!).

Another nice property of the cofactor method is that it preserves symmetry of a multi-scaling function.

Theorem 3.8.6 *Suppose $H_0(z)$ corresponds to a symmetric scaling function $\phi(t)$ (all components of $\phi(t)$ are either symmetric or antisymmetric). Then, $F_0(z) = \frac{1}{\lambda_0^*}\Delta_0(z)K_{H_0}^*(z)$ also corresponds to a symmetric multi-scaling function $\psi(t)$.*

Proof. By Theorem 3.5.1, the fact that $\phi(t)$ is symmetric implies the following condition on $H_0(z)$:

$$H_0(z) = E(z^2)H_0(\frac{1}{z})E^{-1}(z), \tag{3.51}$$

where $E(z) = \mathrm{diag}(\pm z^{2T_0}, \ldots, \pm z^{2T_{n-1}})$, T_j are constants, and "$-$" is chosen if $\phi_k(t)$ is antisymmetric. By our convention $z = e^{-i\omega}$ so $\bar{z} = z^{-1}$, and from (3.51) follows that

$$(H_0^{-1}(z))^* = E(z^2)(H_0^{-1}(\frac{1}{z}))^* E^{-1}(z),$$

$$(\det H_0(z))^* = z^{-T}(\det H_0(\frac{1}{z}))^*, \quad T = 2\sum_{j=0}^{n-1} T_j,$$

and

$$K_{\boldsymbol{H}_0}(z)^* = (\det \boldsymbol{H}_0(z))^* (\boldsymbol{H}_0^{-1}(z))^* = \boldsymbol{E}_1(z^2)(\det \boldsymbol{H}_0(\frac{1}{z})\boldsymbol{H}_0^{-1}(\frac{1}{z}))^* \boldsymbol{E}_1^{-1}(z),$$

where $\boldsymbol{E}_1(z) = \mathrm{diag}(\pm z^{2T_0 - T}, \ldots, \pm z^{2T_{n-1} - T})$. Moreover, if $\det \boldsymbol{H}_0(z) = z^{-T}(\det \boldsymbol{H}_0(\frac{1}{z}))^*$, the solution $\Delta_0(z)$ of (3.49) can be chosen such that $\Delta_0(z) = z^{-T}\Delta_0(\frac{1}{z})$, so

$$\boldsymbol{F}_0(z) = \boldsymbol{E}_2(z^2)\boldsymbol{F}_0(\frac{1}{z})\boldsymbol{E}_2^{-1}(z),$$

where $\boldsymbol{E}_2(z) = \mathrm{diag}(\pm z^{2T_0 - 2T}, \ldots, \pm z^{2T_{n-1}2 - T})$, which means that $\psi(t)$ is also symmetric. ∎

3.9 Construction of Biorthogonal Multiwavelets

Previous section described a way to construct a dual multi-scaling functions. This gives us the low-pass filters $\boldsymbol{H}_0$ and $\boldsymbol{F}_0$. Now we are going to show how the high-pass multi-filters can be constructed, such that the whole filter bank has *perfect reconstruction* (PR). In continuous time high-pass filters will correspond to biorthogonal multiwavelets.

We start from a pair of multi-filters $\boldsymbol{H}_0(z)$, $\boldsymbol{F}_0(z)$ such that

$$\boldsymbol{H}_0(z)\boldsymbol{F}_0^*(z) + \boldsymbol{H}_0(-z)\boldsymbol{F}_0^*(-z) = c\boldsymbol{I}$$

and we need to construct another pair of multi-filters $\boldsymbol{H}_1(z)$, $\boldsymbol{F}_1(z)$ such that PR conditions [68] are satisfied:

$$\boldsymbol{H}_0(z)\boldsymbol{F}_1^*(z) + \boldsymbol{H}_0(-z)\boldsymbol{F}_1^*(-z) = \boldsymbol{0} \tag{3.52}$$

$$\boldsymbol{H}_1(z)\boldsymbol{F}_0^*(z) + \boldsymbol{H}_1(-z)\boldsymbol{F}_0^*(-z) = \boldsymbol{0} \tag{3.53}$$

$$\boldsymbol{H}_1(z)\boldsymbol{F}_1^*(z) + \boldsymbol{H}_1(-z)\boldsymbol{F}_1^*(-z) = c\boldsymbol{I} \tag{3.54}$$

In the scalar case conditions (3.52)—(3.54) can be satisfied by the standard *alternating flip* choice [68]. In the multi case this does not work because matrices do not commute.

Condition (3.52) gives a homogeneous system of linear equations on the coefficients of $\boldsymbol{F}_1^*(z)$. The number of equations is $([\frac{N_1}{2}] + [\frac{N_2}{2}] + 1)r^2$, where N_1 is the degree of $\boldsymbol{H}_0(z)$, N_2 is the degree of $\boldsymbol{F}_1(z)$, and $[\cdot]$ stands for integer part. The number of unknowns is $(N_2 + 1)r^2$, so this system always has a nontrivial solution if $N_2 + 1 > [\frac{N_1}{2}] + [\frac{N_1}{2}] + 1$. We definitely want the shortest possible dual scaling function, so we choose N_2 as small as possible.

From (3.52) follows that

$$\det \boldsymbol{H}_0(z)\det \boldsymbol{F}_1^*(z) = (-1)^r \det \boldsymbol{H}_0(-z)\det \boldsymbol{F}_1^*(-z). \tag{3.55}$$

By our assumption $\det \boldsymbol{H}_0(z)$ and $\det \boldsymbol{H}_0(-z)$ do not have common roots (otherwise (3.49) cannot be satisfied), which means that $\det \boldsymbol{F}_1^*(z) = \det \boldsymbol{H}_0(-z)p(z)$ for some polynomial $p(z) = (-1)^r p(-z)$. Similarly to $\boldsymbol{H}_0(z)$ we don't want common roots except 0 in $\boldsymbol{F}_1(z)$ and $\boldsymbol{F}_1(-z)$, so we need $p(z) = z^a$, where a is even if r is even and is odd if r is odd. Then,

$$\det \boldsymbol{F}_1^*(z) = z^a \det \boldsymbol{H}_0(-z). \tag{3.56}$$

It is easy to check that if $\boldsymbol{H}_0(z)$ is symmetric, than $\boldsymbol{F}_1(z)$ also can be chosen symmetric.

Assume now that (3.52) and (3.56) are satisfied and $\boldsymbol{F}_1(z)$ is found. Then $\boldsymbol{H}_1(z)$ can be obtained using cofactor method.

Theorem 3.9.1 *Suppose we are given $\boldsymbol{H}_0(z)$, $\boldsymbol{F}_0(z)$, $\boldsymbol{F}_1(z)$ such that* (3.48), (3.52), (3.56) *are satisfied and $\boldsymbol{F}_0(z) = \frac{1}{\lambda_0^*}\Delta_0(z)\boldsymbol{K}_{\boldsymbol{H}_0}^*(z)$. Then $\boldsymbol{H}_1(z) = z^{-a}\Delta_0^*(-z)\boldsymbol{K}_{\boldsymbol{F}_1^*}(z)$ will complete the system to a PR filter bank.*

Proof First, let us check condition (3.53). Substitute there the expressions for $\boldsymbol{F}_0$ and $\boldsymbol{H}_1$ and recall that $\boldsymbol{K}_{\boldsymbol{A}} = \det \boldsymbol{A} \cdot \boldsymbol{A}^{-1}$:

$$z^{-a}\Delta_0^*(-z)\det \boldsymbol{F}_1^*(z)(\boldsymbol{F}_1^*(z))^{-1}\frac{1}{\lambda_0}\Delta_0^*(z)\det \boldsymbol{H}_0(z)\boldsymbol{H}_0^{-1}(z)$$
$$+(-z)^{-a}\Delta_0^*(z)\det \boldsymbol{F}_1^*(-z)(\boldsymbol{F}_1^*(-z))^{-1}\frac{1}{\lambda_0}\Delta_0^*(-z)\det \boldsymbol{H}_0(-z)\boldsymbol{H}_0^{-1}(-z) = 0. \tag{3.57}$$

From (3.52) follows that

$$\boldsymbol{H}_0(z)\boldsymbol{F}_1^*(z) = -\boldsymbol{H}_0(-z)\boldsymbol{F}_1^*(-z), \quad (\boldsymbol{F}_1^*(z))^{-1}\boldsymbol{H}_0^{-1}(z) = -(\boldsymbol{F}_1^*(-z))^{-1}\boldsymbol{H}_0^{-1}(-z).$$

This in combination with (3.56) proves that (3.57), and thus (3.53) hold.

(3.54) can be checked analogously by taking into account (3.48), (3.49) and (3.56).

■

Theorem 3.8.6 completes the construction of biorthogonal multi-filter bank starting from a low-pass analysis filter $\boldsymbol{H}_0$.

3.10 Biorthogonal Finite Element Multi-filter Bank

In this section we apply the developed technique to the construction of biorthogonal filter banks starting from FEM functions, described in Section 2.2. There were

constructed semi-orthogonal analysis multiwavelets, but the filter bank was never completed by the synthesis part. Below we give up semi-orthogonality in order to get finite support of analysis functions and obtain FIR filter bank with PR.

Let us begin with the FEM function plotted in Figure 2.2. This multi-scaling function $\phi(t) = [\phi_0(t) \;\; \phi_1(t)]^T$ has two continuously differentiable components ($r = 2$), which are piece-wise Hermit cubics. As was shown in Section 2.2, $\phi(t)$ ensures approximation order 4. We use the symbol corresponding to $\phi(t)$ which was described in [58]:

$$\boldsymbol{H}_0(z) = \frac{1}{16}\begin{bmatrix} 4(1+z)^2 & -2(1-z)(1+z) \\ 3(1-z)(1+z) & -1+4z-z^2 \end{bmatrix}.$$

$\boldsymbol{K}^*_{\boldsymbol{H}_0}(z) = \frac{1}{z^2 16}\begin{bmatrix} -1+4z-z^2 & -3(z-1)(1+z) \\ 2(z-1)(1+z) & 4(1+z)^2 \end{bmatrix}$, and no rescaling is needed because $\lambda_0 = 1$. Further, $\det \boldsymbol{H}_0(z) = \frac{1}{128}(1+z)^4$so (3.49) is satisfied by Daubechies choice of $\Delta_0^*(z) = \frac{-1+4z-z^2}{2z^3}$. Finally the dual symbol is

$$\boldsymbol{F}_0(z) = \Delta_0(z)\boldsymbol{K}^*_{\boldsymbol{H}_0}(z) = \frac{1}{32z}\begin{bmatrix} 1-8z+18z^2-8z^3+z^4 & -3+12z-12z^3+3z^4 \\ 2-8z+8z^3-2z^4 & -4+8z+24z^2+8z^3-4z^4 \end{bmatrix}.$$

The dual multi-scaling function $\psi(t) = [\psi_0(t) \;\; \psi_1(t)]^T$ is plotted in Figure 3.2.

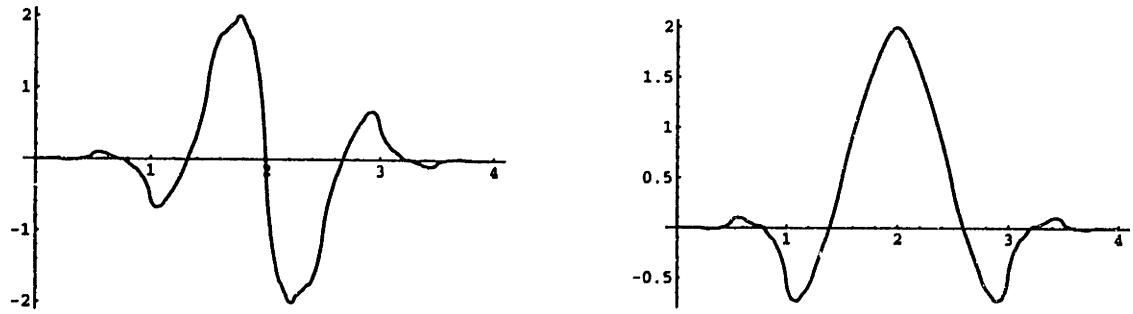


Figure 3.2: Multi-scaling function dual to Hermit cubics

In accordance with Theorem 3.8.6, $\psi(t)$ is symmetric. By Theorem 3.8.3, $\psi(t)$ provides approximation order 4, but it is less smooth then $\phi(t)$. Actually it is point-wise continuous but not continuously differentiable (see Theorem 3.4.3, [58, 15]).

The next step is to find high-pass reconstruction filter $\boldsymbol{F}_1(z)$, such that (3.52), (3.56) hold.

$$\boldsymbol{F}_1(z) = \frac{z}{16}\begin{bmatrix} 4(1-z)^2 & -6(1-z)(1+z) \\ (1-z)(1+z) & -1-4z-z^2 \end{bmatrix}.$$

satisfies this conditions.

To complete the construction we compute $\boldsymbol{H}_1(z)$:

$$\boldsymbol{H}_1(z) = \Delta_0^*(-z)\boldsymbol{K}_{\boldsymbol{F}_1^*}(z) =$$
$$\frac{1}{32}\begin{bmatrix} 1+8z+18z^2+8z^3+z^4 & -1-4z+4z^3+z^4 \\ 6+24z+24z^3-6z^4 & -4-8z+24z^2-8z^3-4z^4 \end{bmatrix}.$$

Figures 3.3, 3.4 display analysis and synthesis wavelets corresponding to $\boldsymbol{H}_1$ and $\boldsymbol{F}_1$.

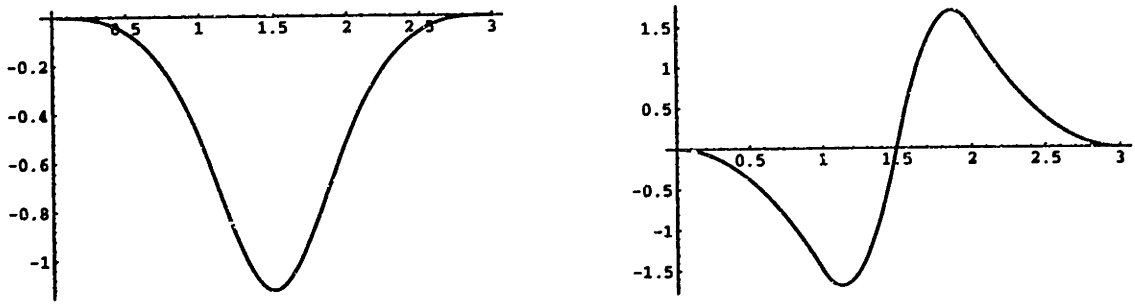


Figure 3.3: Piecewise cubic multiwavelet

It is also worth mentioning that the whole FEM filter bank has dyadic coefficients. This makes exact all floating point computations with it.

The same procedure can be repeated for FEM functions of any order. If $\boldsymbol{H}_0(z)$ corresponds to a FEM of order r then it provides approximation order $2r$ (Section 2.2) and from Corollary 3.3.5 follows that $\det\boldsymbol{H}_0(z) = (1+z)^{2r}q(z)$, where $q(z) \neq 0$ is a polynomial. But as we sow in Section 2.2, $\boldsymbol{H}_0(z)$ has only 3 nonzero coefficients so its degree is 2. Thus $\det\boldsymbol{H}_0(z)$ cannot have degree higher than $2r$ and

$$\det\boldsymbol{H}_0(z) = \text{const} \cdot (1+z)^{2r}. \tag{3.58}$$

From (3.58) follows that $\Delta_0^*(z)$ for FEM of order r is the Daubechies residual polinimial of order r [19]. By Theorems 3.8.3, 3.8.6 dual FEM functions of order r are symmetric, have dyadic coefficients, and ensure approximation order $2r$.

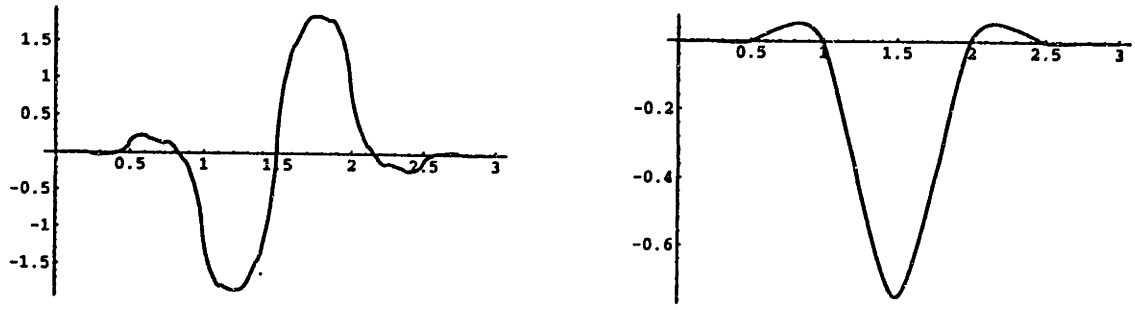


Figure 3.4: Multiwavelet biorthogonal to piecewise cubic

Choice of $F_1(z)$ for FEM also is very easy and transparent. According to Theorem 2.2.1,

$$H_0(z) = C_0 + C_1 z + S C_0 S^{-1} z^2.$$

Then the following F_1 satisfies (3.52) and (3.56):

$$F_1(z) = -C_1^{-1} C_0 B_1 + B_1 z - S C_1^{-1} C_0 B_1 S^{-1} z^2,$$

where C_0, C_1, S are defined in Theorem 2.2.1 and B_1 is any non-degenerate diagonal matrix.

3.11 Orthogonal Multiwavelets

In Section 3.9 we showed how to construct biorthogonal multiwavelets, starting from two dual multi-scaling functions. We gave an explicit expression for three matrices H_0, F_0, H_1 of four. The only symbol which wasn't described explicitly was F_1. It corresponds to the dual wavelets. If the initial multi-scaling function has orthogonal translates, then dual bases coincide, so $H_0 = F_0$, $H_1 = F_1$ and there is a clear way to get F_1 [69]. We present one such method in this section. Another constructions can be found in [38, 42, 78].

It is worth mentioning that in the scalar case, an orthogonal wavelet can be obtained from a given orthogonal scaling function, just by the "flip" construction [68]. Unfortunately matrices don't commute and the construction of an orthogonal multiwavelet becomes a real design problem.

Suppose we are given an orthogonal multi-scaling function

$$\phi(t) = \sum_k h_k^0 \phi(2t - k).$$

Then $H_0(z) = \sum_k h_k^0 z^k$, is the refinement symbol corresponding to $\phi(t)$. As $\phi(t)$ is orthogonal, $F_0(z) = H_0(z)$ and the condition of biorthogonality (3.48) becomes the condition of orthogonality (3.45):

$$H_0(z)H_0^*(z) + H_0(-z)H_0^*(-z) = cI.$$

This equation can be rewritten in more convenient polyphase form [68]

$$L_{\text{poly}}(z)L_{\text{poly}}^*(z) = cI. \tag{3.59}$$

Here $L_{\text{poly}}(z)$ is an $r \times 2r$ polyphase matrix which separates even and odd powers of z in H_0:

$$L_{\text{poly}}(z) = [h_0^0 \;\; h_1^0] + [h_2^0 \;\; h_3^0]z + \ldots + [h_2^0 k \;\; h_{2k+1}^0]z^k + \ldots.$$

Condition (3.59) means that $L_{\text{poly}}(z)$ is paraunitary (see [80]).

In order to get an orthogonal multiwavelet $w(t)$,

$$w(t) = \sum_k f_k^1 \phi(2t - k),$$

we need to construct such symbol $F_1(z)$,

$$F_1(z) = \sum_k f_k^1 z^k,$$

that

$$\begin{aligned} H_0(z)F_1^*(z) + H_0(-z)F_1^*(-z) &= 0 \\ F_1(z)F_1^*(z) + F_1(-z)F_1^*(-z) &= cI. \end{aligned} \tag{3.60}$$

(see conditions of perfect reconstruction (3.52)).

In the polyphase notation, (3.60) can be rewritten in the following way

$$\begin{aligned} L_{\text{poly}}(z)B_{\text{poly}}^*(z) &= 0 \\ B_{\text{poly}}(z)B_{\text{poly}}^*(z) &= cI, \end{aligned} \tag{3.61}$$

where

$$B_{\text{poly}}(z) = [f_0^1 \;\; f_1^1] + [f_2^1 \;\; f_3^1]z + \ldots + [f_2^1 k \;\; f_{2k+1}^1]z^k + \ldots.$$

Combination of (3.59) and (3.61) means that actually we need to complete a $r \times 2r$ paraunitary matrix $\boldsymbol{L}_{\text{poly}}(z)$ to a square $2r \times 2r$ paraunitary matrix $\boldsymbol{H}_{\text{poly}}(z)$,

$$\boldsymbol{H}_{\text{poly}}(z) = \begin{bmatrix} \boldsymbol{L}_{\text{poly}}(z) \\ \boldsymbol{B}_{\text{poly}}(z) \end{bmatrix}.$$

We suggest to do it following Vaidyanathan's book [80]. There $\boldsymbol{L}_{\text{poly}}(z)$, the top half of the matrix $\boldsymbol{H}_{\text{poly}}(z)$, is factored into $\boldsymbol{L}_{\text{poly}}(1)\boldsymbol{U}(z)$. The paraunitary matrix $\boldsymbol{U}(z)$ is square — in our case $2r \times 2r$. The constant $r \times 2r$ matrix $\boldsymbol{L}_{\text{poly}}(1)$ can be completed by $\boldsymbol{B}_{\text{poly}}(1)$ to a constant unitary matrix. Then we get the desired lower half $\boldsymbol{B}_{\text{poly}}(z)$ from the product $\boldsymbol{B}_{\text{poly}}(1)\boldsymbol{U}(z)$:

$$H(z) = \begin{bmatrix} \boldsymbol{L}_{\text{poly}}(1) \\ \boldsymbol{B}_{\text{poly}}(1) \end{bmatrix} \begin{bmatrix} \boldsymbol{U}(z) \end{bmatrix}. \tag{3.62}$$

Vaidyanathan gives a specific form to $\boldsymbol{U}(z)$. It is a product of Householder-type factors $\boldsymbol{I} - \boldsymbol{v}\boldsymbol{v}^T + z\boldsymbol{v}\boldsymbol{v}^T$. They can be obtained one at a time from $\boldsymbol{L}_{\text{poly}}(z)$, each with $||\boldsymbol{v}|| = 1$ and determinant z. After finitely many steps, making up $\boldsymbol{U}(z)$, we reach the constant matrix $\boldsymbol{L}_{\text{poly}}(1)$.

Example 3.11.1 For the GHM $\boldsymbol{H}_0(z)$ is defined by (3.42) the 2×4 matrix $\boldsymbol{L}_{\text{poly}}(z)$ can be completed to the 4×4 matrix $\boldsymbol{H}_{\text{poly}}(z)$ in a single operation — creating $\boldsymbol{U}(z)$ instead of its factors. This produces the wavelet coefficients $\boldsymbol{f}_k^1$. There will be two families of possible $\boldsymbol{U}(z)$ and therefore two families of wavelets. We chose the only possible symmetric coefficients.

$$\boldsymbol{f}_0^1 = \frac{1}{10}\begin{bmatrix} -\frac{1}{\sqrt{2}} & -3 \\ 1 & 3\sqrt{2} \end{bmatrix} \qquad \boldsymbol{f}_1^1 = \frac{1}{10}\begin{bmatrix} \frac{9}{\sqrt{2}} & -10 \\ -9 & 0 \end{bmatrix}$$

$$\boldsymbol{f}_2^1 = \frac{1}{10}\begin{bmatrix} \frac{9}{\sqrt{2}} & -3 \\ 9 & -3\sqrt{2} \end{bmatrix} \qquad \boldsymbol{f}_3^1 = \frac{1}{10}\begin{bmatrix} -\frac{1}{\sqrt{2}} & 0 \\ -1 & 0 \end{bmatrix}.$$

The corresponding wavelet functions are plotted in Figure 1.6. For more details of this particular construction see [69].

Chapter 4

Application of Multiwavelets to Signal Processing

Wavelets have been proven to be a useful tool in data processing such as image compression, denoising, etc. Before only scalar wavelets were used. Multiwavelets have some advantages in comparison to scalar ones. For example, such features as short support, orthogonality, symmetry, vanishing moments are known to be important in signal processing. A scalar wavelet *cannot* posses all these properties at the same time [19]. On the other hand, a multiwavelet system *can* have all of them simultaneously. This suggests that multiwavelets can provide perfect reconstruction (orthogonality), good performance at the boundaries (symmetry) and high order of approximation (vanishing moments), so they could perform better than the scalar ones.

In this Chapter we describe a technique of implementation of multiwavelets and give some experimental results (in denoising and image compression) proving that multiwavelets are worth studying.

Multiwavelets differ from scalar wavelet systems in requiring several input streams to the filter bank. We describe two methods (repeated row and approximation/deapproximation) for obtaining such a vector input stream from a one-dimensional signal. Algorithms for symmetric extension of signals at boundaries are then developed, and naturally integrated with approximation-based preprocessing. We describe an additional algorithm for multiwavelet processing of two-dimensional signals, two rows at a time, and develop a new family of multiwavelets (the constrained pairs) that is well-suited to this approach. These techniques are then applied to two basic signal processing problems, denoising via thresholding, and data compression. Performance of multiwavelets frequently was better that the performance of comparable scalar wavelet transform. More detailed description of the experiments can be

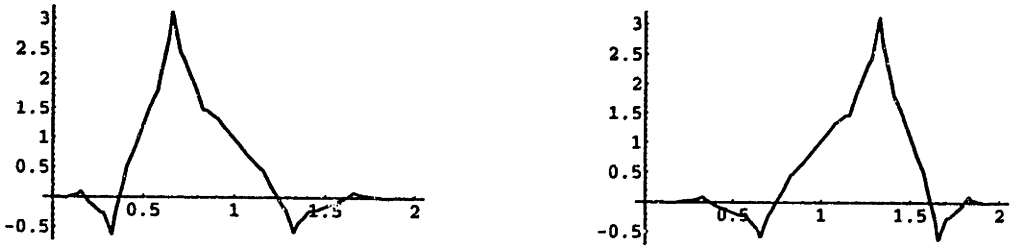


Figure 4.1: Symmetric pair of orthogonal scaling functions

found in [72].

The principal filter bank of this chapter corresponds to GHM scaling functions and wavelets. There are four remarkable properties of GHM scaling functions: **(1)** They are short (first is supported on [0 1], second on [0 2]). **(2)** Both scaling functions are symmetric. **(3)** Translates of scaling functions are orthogonal. **(4)** The system has second order of approximation (i.e. constant and linear functions are in V_0). One scaling function cannot combine symmetry and orthogonality and if it corresponds to a dilation equation with four coefficients, it is supported on [0 3]!

Another useful multiwavelet pair is the symmetric pair determined by the three coefficients

$$C_0 = \begin{bmatrix} 0 & \frac{2+\sqrt{7}}{4} \\ 0 & \frac{2-\sqrt{7}}{4} \end{bmatrix}, \quad C_1 = \begin{bmatrix} \frac{3}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{3}{4} \end{bmatrix}, \quad C_2 = \begin{bmatrix} \frac{2-\sqrt{7}}{4} & 0 \\ \frac{2+\sqrt{7}}{4} & 0 \end{bmatrix}.$$

The scaling functions for this multiwavelet system are shown in Figure 4; observe that one is the reflection of the other about its center point. In this article we will make use of several other *non-symmetric* multiwavelets with desirable properties.

We also will construct a few examples of multiwavelets filter banks with desirable properties, but everywhere in this Chapter we use systems only with 2 scaling functions.

This is one of the first attempts to apply multiwavelets in the real world (see also [88, 79]). Much more work must be done in this direction. Particularly, theory of preprocessing and symmetric extension should be carefully developed. We also hope that the biorthogonal construction of Section 3.9 will find its place in practice. There is a good chance that multiwavelets can be helpful for solution of PDE's, integral equations, problems of physics and fluid dynamics.

4.1 Preprocessing of the Input Data

To compute wavelet coefficients, corresponding to a signal $f[n]$ one usually uses the cascade algorithm and iterates the low pass filter with downsampling [46]. In the "multi" case the situation does not change. In Figure 4.1 is shown the filter bank corresponding to a multiwavelet system with two scaling functions and two wavelets.

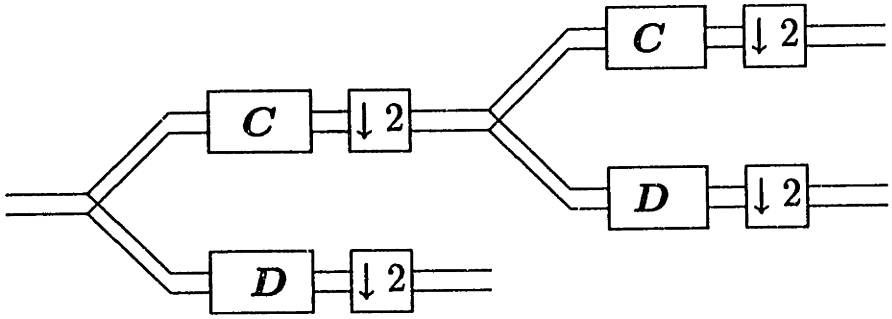


Figure 4.2: A multiwavelet filter bank, iterated once

As in the scalar case, low pass filter $\boldsymbol{C}$ and high pass filter $\boldsymbol{D}$ consist of coefficients, corresponding to the dilation equation (2.1) and wavelet equation (1.7). But now these coefficients are r by r matrices and during the convolution step they must multiply vectors (instead of numbers). This means that multi-filter banks *need r input rows.* Usually there is only one input signal $f[n]$ in the beginning, so some kind of preprocessing of the data must be done before the implementation of a multi-filter bank.

There could be infinitely many ways to do such preprocessing. The most obvious way to get second input row is just to repeat the first one and use two identical rows of length n at the entrance of the cascade. This procedure introduces oversampling of the data by a factor of two. Oversampled representations require more calculation than critically-sampled representations. Furthermore, while compressing data one would like to remove redundancy, not increase it. In the case of one-dimensional signals the "repeated row" scheme is convenient to implement, and our experiments on denoising of one-dimensional signals were encouraging (see Section 4.4.1). In two dimensions the oversampling factor increases to four, making this scheme useful only for applications such as denoising which do not require critically-sampled or near-critically-sampled representation of the data.

A different way to get input rows for the multiwavelet filter bank is to *preprocess* the given scalar signal $f[n]$. For data compression, where one is trying to find compact transform representations for a dataset, it is crucial to find critically sampled multi-wavelet transform schemes. We now describe a preprocessing scheme based on the

approximation properties of the continuous-time multiwavelets which yields a critically sampled signal representation. This scheme (suggested to us by J. Geronimo) is naturally connected with the GHM multiwavelets.

Let function $f(t)$ belong to the first scaling subspace V_0 generated by translates of GHM scaling functions. This means that $f(t)$ is a linear combination of those translates:

$$f(t) = \sum_n v_{1,n}^{(0)} \phi_0(t-n) + v_{2,n}^{(0)} \phi_1(t-n)\,. \tag{4.1}$$

Suppose that the input sequence $f[n]$ contains samples of $f(t)$ at half integers:

$$f[2n] = f(n), \qquad f[2n+1] = f(n+1/2).$$

The support of ϕ_0 is $[0 \;\; 1]$, so it vanishes at all integer points. The support of ϕ_1 is $[0 \;\; 2]$ and it is nonzero at 1. Sampling (4.1) at integers and half integers gives

$$\begin{aligned} f[2n] &= \phi_1(1) v_{2,n} \\ f[2n+1] &= \phi_1(3/2) v_{2,n} + \phi_0(1/2) v_{1,n+1} + \phi_1(1/2) v_{2,n+1}. \end{aligned} \tag{4.2}$$

Coefficients $v_{1,n}$, $v_{2,n}$ can be easily found from (4.2)

$$\begin{aligned} v_{1,n} &= \tfrac{\phi_1(1) f[2n-1] - \phi_1(1/2) f[2n-2] - \phi_1(3/2) f[2n]}{\phi_1(1)\phi_0(1/2)} \\ v_{2,n} &= \tfrac{f[2n]}{\phi_1(1)}. \end{aligned}$$

Taking into account symmetry of $\phi_1(t)$ ($\phi_1(1/2) = \phi_1(3/2)$), we get

$$\begin{aligned} v_{1,n} &= \tfrac{\phi_1(1) f[2n-1] - \phi_1(1/2)(f[2n-2] + f[2n])}{\phi_1(1)\phi_0(1/2)} \\ v_{2,n} &= \tfrac{f[2n]}{\phi_1(1)}. \end{aligned} \tag{4.3}$$

Relations (4.3) give a natural way to get two input rows $v_{1,n}$, $v_{2,n}$ starting from one given signal $f[n]$. Let us also mention that this preprocessing maintains critically sampled representation: the number of data points stays constant and multi-filter deals with two rows of length $N/2$.

Another advantage of this approximation-based preprocessing method is that it fits naturally with symmetric extension for multiwavelets. In other words, if we symmetrically extend a finite length signal $f[n]$ at its boundaries and implement the approximation formulas (4.3), then the two rows $v_{1,n}^{(0)}, v_{2,n}^{(0)}$ from the preprocessor will have the appropriate symmetry (see Section 4.2).

One also can develop a general approximation-type preprocessing, based on the following idea. Suppose again that our given signal f lies in V_0. This implies that

$$f(t) = \sum_{n,k} v_{k,n}^{(0)} \phi_k(t-n). \tag{4.4}$$

The goal of preprocessing is to find the coefficients $v_{k,n}^{(0)}$ from the signal samples.

For the beginning we assume that our multiwavelet system has r scaling functions, all supported on $[0,1]$. Now restrict equation (4.4) to this interval:

$$f(t) = \sum_k v_{k,0}^{(0)} \phi_k(t) , \quad 0 \le t \le 1. \tag{4.5}$$

Suppose that samples $f[0], \ldots, f[r-1]$ are the values of the function $f(t)$ at the points

$$t = \frac{1}{2r}, \frac{3}{2r}, \ldots, \frac{2r-1}{2r}.$$

The representation (4.5) gives a linear system for the coefficients $v_{0,0}^{(0)}, \ldots, v_{r-1,0}^{(0)}$. The following r samples $f[r], \ldots, f[2r-1]$ give the values of $v_{0,1}^{(0)}, \ldots, v_{r-1,1}^{(0)}$. Repeating this procedure we find all the $v_{k,n}^{(0)}$. If some of the scaling functions have support longer than $[0,1]$, we will need several initial (boundary) values of $v_{k,-1}^{(0)}, v_{k,-2}^{(0)}, \ldots$. In the case of finite length signals, these numbers can be obtained from the conditions of periodization or symmetric extension (Section 4.2). Other multiwavelet preprocessing techniques are discussed in [79, 88].

4.2 Symmetric Extension of the Signal at the Boundaries

In practice all signals are of finite length, so we should somehow care about the boundaries. There are two common ways to do it. First is periodization of the data. It introduces discontinuities in signal at the boundaries. The second way is symmetric extension of the data. It keeps signal continuous, but it can be implemented only with symmetric (antisymmetric) filters [6]. GHM multi-filter is symmetric and symmetric data extension is possible with them. A detailed description of this technique is given in [72]. Here we only mention that if we are given two input rows (one of even length, the other of odd length)

$$\begin{matrix} s_0^1 & s_1^1 & s_2^1 & \ldots & s_{N-1}^1 \\ s_0^2 & s_1^2 & s_2^2 & \ldots & s_{N-1}^2 & s_N^2 \end{matrix} ,$$

then we can symmetrically extend them

$$\begin{matrix} \ldots & s_2^1 & s_1^1 & s_0^1 & s_0^1 & s_1^1 & \ldots & s_{N-2}^1 & s_{N-1}^1 & s_{N-1}^1 & \ldots \\ \ldots & s_2^2 & s_1^2 & s_0^2 & s_1^2 & s_2^2 & \ldots & s_{N-1}^2 & s_N^2 & s_{N-1}^2 & \ldots \end{matrix} ,$$

to get four symmetric rows (l_j^k low pass sub-band and h_j^k high pass sub-band) after one step of the cascade algorithm:

$$\begin{array}{cccccccccc} \cdots & l_2^1 & l_1^1 & l_0^1 & l_0^1 & \cdots & l_{N/2-2}^1 & l_{N/2-1}^1 & l_{N/2-1}^1 & \cdots \\ \cdots & l_2^2 & l_1^2 & l_0^2 & l_1^2 & \cdots & l_{N/2-1}^2 & l_{N/2}^2 & l_{N/2-1}^2 & \cdots \end{array}$$

$$\begin{array}{cccccccccc} \cdots & h_2^1 & h_1^1 & h_0^1 & h_1^1 & \cdots & h_{\frac{N}{2}-1}^1 & h_{\frac{N}{2}}^1 & h_{\frac{N}{2}-1}^1 & \cdots \\ \cdots & -h_2^2 & -h_1^2 & 0 & h_1^2 & \cdots & h_{\frac{N}{2}-1}^2 & 0 & -h_{\frac{N}{2}-1}^2 & \cdots \end{array}$$

4.3 Two-dimensional Signal Processing with Multiwavelet Filter Banks

This section is cited from [72] with minor changes.

Multiwavelet filtering of images needs two-dimensional algorithms. One class of such algorithms is derived simply by taking tensor products of the one-dimensional methods described in the previous section. Another class of algorithms comes from using the matrix filters of the multiwavelet system for fundamentally two-dimensional processing. We discuss each of these alternatives.

4.3.1 Separable schemes based on one-dimensional methods

Section 4.1 described two different methods for transforming one-dimensional signals with multiwavelets. Each of these can be turned into a two-dimensional algorithm by taking a tensor product, i.e., by performing the 1D algorithm in each dimension separately. As noted before, the separable product of one-dimensional "repeated row" algorithms leads to a 4:1 data expansion, restricting the use of this approach to applications such as denoising by thresholding, for which near-critical sampling is irrelevant.

The separable product of the approximation-based preprocessing methods described in Section 4.1 yields a critically sampled representation, potentially useful for both denoising and data compression. However, this two-dimensional scheme is not trivial. If we approximate and transform with downsampling, a constant row $f[n] = 1$ would become two different halflength rows. (In the case of GHM scaling functions one row repeats the constant 1, and the other repeats the constant $\sqrt{2}$ in accordance with vector $\boldsymbol{u}_0$. See Example 2.1.7.) To overcome this, we *deapproximate* the multiwavelet filter bank output to create two output rows (one lowpass, one highpass)

before operating in the vertical direction. The vertical transform similarly proceeds as approximate-filter-deapproximate before the next level of the cascade algorithm is applied. This is described schematically in Figure 4.3.1.

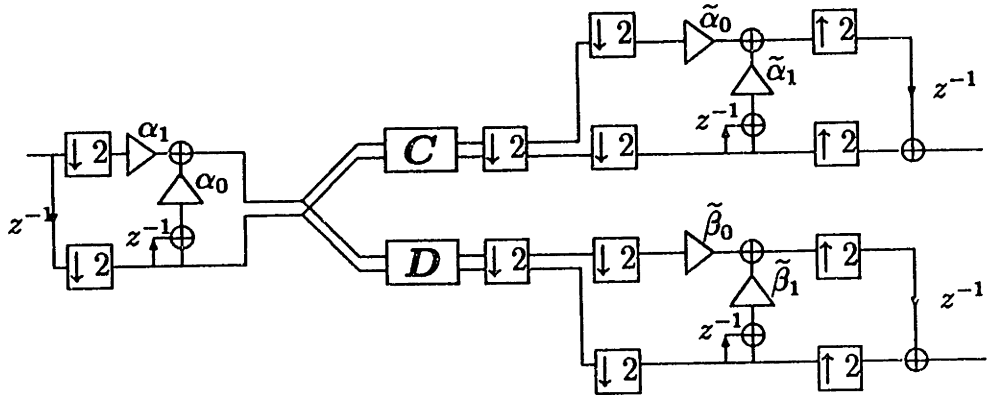


Figure 4.3: Approximation/deapproximation scheme for computing the 2-dimensional multiwavelet transform

4.3.2 Constrained multiwavelets

A different approach to two-dimensional multiwavelet filtering is to make use of the two-dimensionality of the matrix filter coefficients. When processing an image with a scalar filter bank one usually uses as input the rows and columns of the image. For a multiwavelet system we need r input signals. Where can we get them? The first solution which comes to mind is very simple: just use r adjacent rows as the input. For the 2×2 multiwavelets used here, this would mean taking two rows of the image at a time, and applying the matrix filter coefficients to the sequence of 2-element vectors in the input stream.

However, a naive implementation of this approach does not lead to good results (see Table 4.4.4 in Section 4.4.4). This is due to the intricacies of multiwavelet approximation. Approximation of degree m is important for image compression because locally polynomial data can be captured in a few lowpass coefficients. A wavelet system (scalar or multiwavelet) satisfies approximation of degree m (or accuracy m) if polynomials of degree less than m belong to the subspace V_0. Image data is often locally well-approximated by constant, linear, and quadratic functions; thus, such local approximations remain in the lowpass space V_0 after filtering and downsampling. This is one reason why simply retaining the lowpass coefficients of a wavelet decomposition with accuracy m (m vanishing moments) produces good results while compressing the image representation into very few coefficients [90].

When applying multiwavelets to two-dimensional (image) processing, we use this notion of local approximation as a motivation — we wish to capture locally constant and linear features in the lowpass coefficients. Suppose we have a multiwavelet system generated by two scaling functions $\phi_0(t), \phi_1(t)$ with accuracy $m \geq 1$ (this would mean at least one vanishing wavelet moment in the scalar case). Then constant functions $f(t) \equiv c$ locally belong to the scale-limited space V_0. It has been shown Section 2.1 that 1 is an eigenvalue of the filtering and downsampling operator $\boldsymbol{L}$ (see formula (2.2)), and there exists a left eigenvector

$$\boldsymbol{y}^{(0)} = \left[\ldots u_{1,n}\,,\ u_{2,n}\,,\ u_{1,n+1}\,,\ u_{2,n+1}\,,\ \ldots \right]$$

with

$$\boldsymbol{y}^{(0)}\boldsymbol{L} = \boldsymbol{y}^{(0)}\,.$$

In fact, $u_{1,n} = u_{1,0}$ and $u_{2,n} = u_{2,0}$, so that

$$\left[\ldots u_{1,n}\,,\ u_{2,n}\,,\ u_{1,n+1}\,,\ u_{2,n+1}\,,\ \ldots \right] = \left[\ldots u_{1,0}\,,\ u_{2,0}\,,\ u_{1,0}\,,\ u_{2,0}\,,\ \ldots \right]\,.$$

In the continuous-time subspace V_0 this eigenvector leads to the constant function:

$$f(t) = c = c\sum_n \left(u_{1,0}\phi_0(t-n) + u_{2,0}\phi_1(t-n)\right)\,.$$

Assuming that our image is locally constant, we input two equal, constant rows of the image (two-dimensional signal) into the multiwavelet filter bank. The output will be zero in the highpass and a constant vector

$$\begin{bmatrix} c_1 \\ c_2 \end{bmatrix}$$

in the lowpass. If the eigenvector $\boldsymbol{y}^{(0)}$ of $\boldsymbol{L}$ satisfies $u_{1,0} = u_{2,0}$, then we will get $c_1 = c_2$ and the constant input yields a constant lowpass output. However, there is no guarantee of this happy state; for example, in the case of the GHM multiwavelet

$$\begin{bmatrix} u_{1,0} & u_{2,0} \end{bmatrix} \propto \begin{bmatrix} 1 & \sqrt{2} \end{bmatrix}$$

and therefore $c_1 \neq c_2$. Thus the lowpass responses of an arbitrary multifilter to a constant input are *different* constants. Quantization of these lowpass multifilter outputs (for lossy compression) will then introduce a rippled texture in the lowpass part of the image, creating unacceptable artifacts. This is borne out by experiments using the GHM multifilter (Section 4.4.4 below).

Similar arguments hold for linear approximation (Section 2.1): a multiwavelet system has linear approximation (accuracy of order $m = 2$) if and only if there are two left eigenvectors. The first is

$$\boldsymbol{y}^{(0)} = \left[\ \ldots u_{1,0}\,,\ u_{2,0}\,,\ u_{1,0}\,,\ u_{2,0}\,,\ \ldots \right]$$

satisfying

$$\boldsymbol{y}^{(0)}\boldsymbol{L} = \boldsymbol{y}^{(0)}\,.$$

as before. The second eigenvector is

$$\boldsymbol{y}^{(1)} = \left[\ \ldots v_{1,n}\,,\ v_{2,n}\,,\ v_{1,n+1}\,,\ v_{2,n+1}\,,\ \ldots \right]$$

also satisfying

$$\boldsymbol{y}^{(1)}\boldsymbol{L} = \frac{1}{2}\boldsymbol{y}^{(1)}\,.$$

For linear approximation we must have

$$v_{1,n} = y_{1,0} - nu_{1,0}$$

and

$$v_{2,n} = y_{2,0} - nu_{2,0}$$

for some constants $y_{1,0}$ and $y_{2,0}$, so that

$$\boldsymbol{y}^{(1)} = \left[\ \ldots y_{1,0} - nu_{1,0}\,,\ y_{2,0} - nu_{2,0}\,,\ y_{1,0} - (n+1)u_{1,0}\,,\ y_{2,0} - (n+1)u_{2,0}\,,\ \ldots \right]\,.$$

This second eigenvector leads to linear approximation; indeed,

$$g(t) = t = \sum_n v_{1,n}\phi_0(t-n) + v_{2,n}\phi_1(t-n)\,.$$

Again, there is no reason to expect that $y_{1,0} = y_{2,0}$, and so if we input two equal linear rows into the multifilter, they will most likely emerge as two *different* linear rows. Thus the locally linear nature of many images will become distorted under such a multiwavelet transform, and this distortion will lead to unacceptable artifacts under quantization.

One way to avoid this phenomenon is to construct a multiwavelet system in which the eigenvectors have pairwise equal components

$$\boldsymbol{y}^{(0)} = \left[\ \ldots u_{1,0}\,,\ u_{1,0}\,,\ u_{1,0}\,,\ \ldots \right] \tag{4.6}$$

and

$$\boldsymbol{y}^{(1)} = \left[\ \ldots y_{1,0} - nu_{1,0}\,,\ y_{1,0} - nu_{1,0}\,,\ y_{1,0} - (n+1)u_{1,0}\,,\ y_{1,0} - (n+1)u_{1,0}\,,\ \ldots \right]\,, \tag{4.7}$$

which produces two equal linear outputs as the response to two equal linear inputs. Such multiwavelets can be constructed, but as we will see, the restrictions (4.6) and (4.7) imply some constraints on the properties of the multi-scaling functions.

Consider a multiwavelet system with two scaling functions satisfying a matrix dilation equation with four coefficients

$$\phi(t) = C_0\phi(2t) + C_1\phi(2t-1) + C_2\phi(2t-2) + C_3\phi(2t-3)\,. \tag{4.8}$$

It is proven in Section 2.1 that the vectors $u_0 = [u_{1,0} \quad u_{2,0}]$, $u_1 = [y_{1,0} \quad y_{2,0}]$ must satisfy the following system of equations:

$$\begin{aligned} u_0\,(C_0 + C_2) &= u_0 \\ u_0\,(C_1 + C_3) &= u_0 \\ u_1 C_1 + (u_0 + u_1)C_3 &= \tfrac{1}{2}u_1 \\ u_1 C_0 + (u_0 + u_1)C_2 &= \tfrac{1}{2}(u_0 + u_1)\,. \end{aligned} \tag{4.9}$$

We want

$$u_{1,0} = u_{2,0} = u_0\,, \tag{4.10}$$

and

$$y_{1,0} = y_{2,0} = y_0\,, \tag{4.11}$$

i.e., $u_1 = (y_0/u_0)\,u_0$. From the dilation equation (4.8) and the approximation constraints (4.9), it follows that u is a mutual eigenvector of all four matrices C_k:

$$u\,C_k = c_k' u,\ k = 0,\ 1,\ 2,\ 3\,. \tag{4.12}$$

Consider now a scalar function $\phi(t)$

$$\phi(t) = \frac{1}{u_0}u_0\,\phi(t) = \phi_0(t) + \phi_1(t).$$

According to (4.8) and (4.12), $\phi(t)$ satisfies the scalar dilation equation

$$\phi(t) = c_0'\phi(2t) + c_1'\phi(2t-1) + c_2'\phi(2t-2) + c_3'\phi(2t-3).$$

The only solution to this equation with orthogonal translates and second order of approximation is Daubechies' D_4 scaling function [19]. Thus any orthogonal pair $\{\phi_0\,,\ \phi_1\}$ which has second order of approximation, satisfies the dilation equation (4.8), and the eigenvector constraints (4.10) — (4.11) must sum to D_4:

$$\phi_0(t) + \phi_1(t) = D_4(t).$$

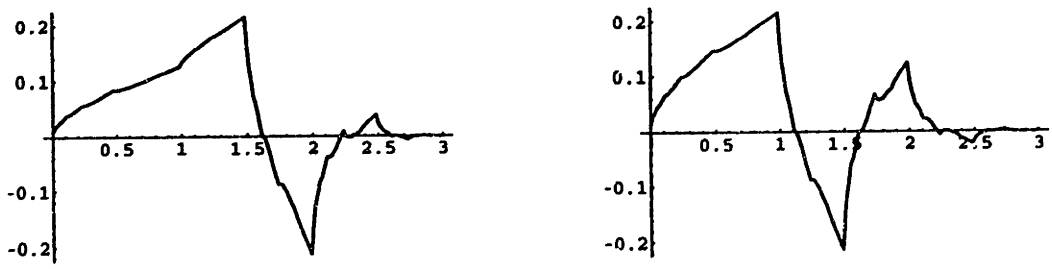


Figure 4.4: Constrained Pair 1

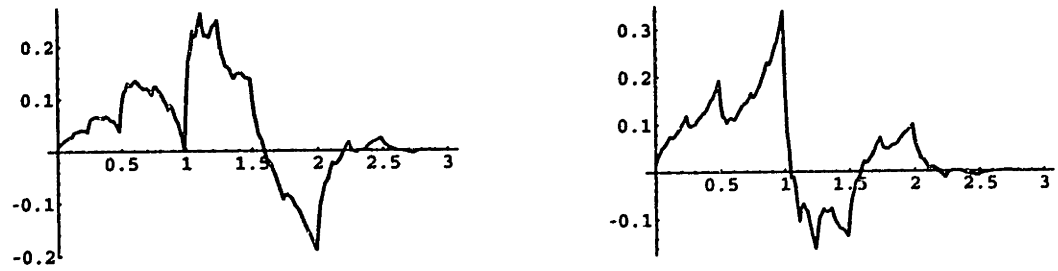


Figure 4.5: Constrained Pair 2

We call such pairs "constrained" multi-scaling functions. There are infinitely many constrained orthogonal solutions of (4.12). Plots of two of them are shown in Figures 4.4 and 4.5.

The implementation of constrained multiwavelets for the two-dimensional wavelet transform is straightforward. In each step of Mallat's algorithm [46], one first processes pairs of rows and then pairs of columns. Because locally constant and linear data are passed through to the lowpass ouputs of a constrained multifilter, the performance of these constrained multiwavelets in image compression is much better than that of the "non-constrained" GHM pair, when applied by using two adjacent rows as the input. This is confirmed by the experiments reported in the next section, as shown in Tables 4.4 and 4.5.

4.4 Signal Processing Applications of Multiwavelets

Finally we are ready to present the results of numerical experiments.

In this section we are going to compare the numerical performance of GHM and constrained multiwavelets with Daubechies D_4 scalar wavelets. D_4 wavelets were chosen because they have two vanishing moments, are orthogonal and have four co-

efficients in the dilation equation — exactly like the GHM and constrained pairs. We perform these comparisons in two standard wavelet applications: signal denoising and data compression. We first develop these applications for one-dimensional signals, then extend them to images.

4.4.1 Denoising by thresholding

Suppose we have a noisy signal

$$g[n] = f[n] + \sigma z[n],$$

where $z[n]$ is Gaussian white noise. We want to recover $f[n]$ from $g[n]$. In [21, 22] was proposed an algorithm to do it:

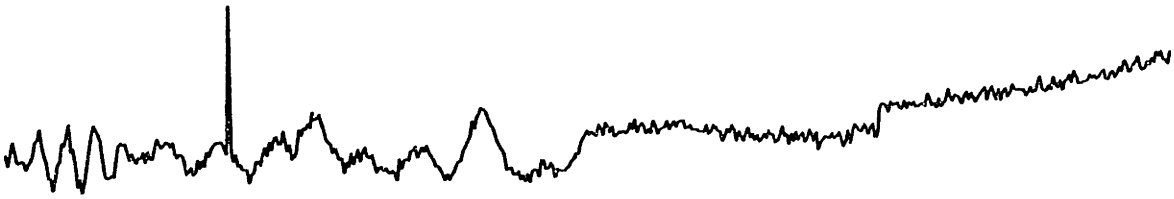
1. Apply the cascade algorithm to get the wavelet coefficients corresponding to $g[n]$.
2. Choose threshold $t_n = \sqrt{2\log(n)}\sigma/\sqrt{n}$ and apply (soft) thresholding to the wavelet coefficients.
3. Invert the cascade algorithm to get denoised signal $\tilde{f}[n]$.

There was proved that $\tilde{f}$ with high probability is at least as smooth as f and $\tilde{f}$ is close to f in the sense of mean square error.

We implemented Donoho's wavelet shrinkage algorithm using several additional remarks from [54]. We compared the performance of the D_4 scalar wavelet transform with oversampled and critically sampled multiwavelet schemes. In the oversampled scheme, the first row is multiplied by $\sqrt{2}$, to better match the first eigenvector of the GHM system. The critically sampled scheme uses the formulas (4.3) to obtain two input rows $v_{1,n}$, $v_{2,n}$ from a single row of data. After reconstruction the two output rows $\tilde{v}_{1,n}$, $\tilde{v}_{2,n}$ are deapproximated using (4.2), to yield the output signal $\tilde{f}[n]$. Boundaries are handled by symmetric data extension for the critically sampled (approximation/deapproximation) and oversampled schemes, and by circular periodization for D_4.

Results of a typical experiment are shown in Table 4.1 and Figure 4.6. In all experiments both types of GHM filter banks performed better than D_4. The "repeated row" usually gave better results than "approximation" preprocessing. This is not surprising, because it is known that oversampled data representations are useful for feature extraction.

Original signal, 512 samples. Range of amplitude $[-3, 10]$



Noisy signal. Noise level $\sigma = 0.3$

Signal reconstructed using GHM with "approximation"

Signal reconstructed using GHM with "repeated row"

Signal reconstructed using D_4

Figure 4.6: Denoising via wavelet-shrinkage

	Noise	GHM with approximation	GHM with repeated row	D_4
mean abs. error	0.243	0.154	0.128	0.213
mean square error	0.093	0.045	0.029	0.062

Table 4.1: 1D Denoising via wavelet soft thresholding

4.4.2 1D signal compression

Scalar wavelets appeared to be useful in the image coding because of their energy compaction property [46]. The following example shows that the multiwavelets could be even better.

As a model problem we took the same signal as in the denoising experiments. It contains 512 data points. We perform 8 iterations of the cascade algorithm to get the wavelet coefficients λ_k, using the same 3 types of filter banks. Then we compare the number of "big" coefficients, those which are bigger then $\lambda^i_{max}/100$ (λ^i_{max}, $i = 1, 2, 3$ is the biggest wavelet coefficient for each filter bank). After this we threshold with $t = \lambda^i_{max}/100$ and apply reversed cascade algorithm to reconstruct the signal. The results are in Table 4.2, Figure 4.7.

	GH with "appr."	GH with "rep. row"	D4
number of "big" coeff.	72	79	92
max abs. value	2.979	4.221	1.984
mean abs. error	0.0572	0.0589	0.0580
mean square error	0.0082	0.0085	0.0072

Table 4.2: 1D Compression

One can see that multiwavelets perform better in terms of number of "big" coefficients. GHM with "approximation" slightly overcome GHM with "repeated row".

The results of this experiment suggest that GH with "approximation" is worth trying in 2D image compression.

4.4.3 Denoising of images

Given the success of the multiwavelet system in denoising of the model one-dimensional signal, we applied multiwavelet denoising to images. We added white Gaussian noise

Original signal, 512 samples. Range of amplitude $[-3, 10]$

Signal reconstructed using GH with "approximation"

Signal reconstructed using GH with "repeated row"

Signal reconstructed using D4

Figure 4.7: 1D Compression

to the *Lenna* image, and applied three wavelet transforms for denoising by wavelet shrinkage: the GHM multiwavelet with approximation, GHM with repeated row, and the Daubechies 4-tap scalar wavelet. The experimental results are shown in Table 4.3 and the resulting images in Figure 4.8 with approximation was superior to D_4 both numerically and subjectively; the approximation-based preprocessing seemed to reduce the Cartesian artifacts present in the scalar wavelet shrinkage. This can be seen, for example, in the facial features (eyes, nose) of the *Lenna* images shown in Figure 4.8. The GHM-repeated row scheme suffered because we had to repeat rows in first the x dimension and then in the y dimension, altering the correlations of the data. This produces the broad stripes in the image denoised with the repeated row scheme.

	Noise	GHM with approximation	GHM with repeated row	D_4
ℓ^1 error	19.93	6.87	9.70	8.09
ℓ^2 error	24.98	9.75	12.6	11.4

Table 4.3: Denoising of *Lenna* image via wavelet-shrinkage

4.4.4 Transform-based image coding

One of the most successful applications of the wavelet transform is image compression. We compared the two-dimensional multiwavelet algorithms of Section 4.3 with a D_4 scalar wavelet in image coding. Five types of wavelet transforms were used:

- D_4 scalar wavelet
- Approximation/deapproximation preprocessing with GHM multiwavelets
- Adjacent rows input with GHM multiwavelets
- Adjacent rows input with symmetric pair
- Adjacent rows input with two different constrained pairs

Each of these wavelet transforms was followed by entropy-constrained scalar quantization. We then entropy-coded the resulting coefficient streams using a combination of zero-run-length coding and adaptive Huffman coding, as in the FBI's Wavelet Scalar Quantization standard [25].

Lenna image with Gaussian noise
MSE 24.98

GHM-with-approximation
multiwavelet denoising, MSE 9.75

Daubechies 4 scalar
wavelet denoising, MSE 11.4

GHM-repeated-row
multiwavelet denoising, MSE 12.6

Figure 12
Multiwavelet denoising

Figure 1.8: 2D Denoising

We applied these wavelet image coders to the Lenna (NITF6) image, as well as a geometric test pattern, at several compression ratios. The results are shown in Tables 4.4 and 4.5, and in Figures 4.9, 4.10. On Lenna, the GHM multiwavelet with approximation mildly outperformed the D_4 scalar wavelet at compression ratios of 32:1 and 64:1. The images in Figure 4.9 show that the GHM-approximation scheme preserves more texture in the hat and, as in the denoising application, produce fewer Cartesian artifacts than the scalar wavelet scheme. The adjacent-row method did not work well on *Lenna*. However, the adjacent-row method produced the best compressions of the test pattern image (Figure 4.10) at intermediate compression ratios (16:1 and 32:1), with the constrained pair #1 "CP-1" outperforming both D_4 and GHM with approximation. When using the adjacent row algorithm, the constrained pairs significantly outperformed the GHM symmetric multiwavelet, demonstrating the importance of the eigenvector constraints (4.6 - 4.7). A close look at the details of the compressed/decompressed test patterns shows that the CP-1 compression did a better job of preserving the checkerboard and giving artifacts over a shorter distance than the D_4 compression.

These results suggest that multiwavelets are worth of further investigation as a technique for image compression. Future work will include design and implementation of new multiwavelets with symmetry, orthogonality and higher orders of approximation, as well as implementation of biorthogonal multiwavelets and elaboration of general preprocessing techniques.

Compression Ratio	8:1	16:1	32:1	64:1
	pSNR	pSNR	pSNR	pSNR
Daubechies 4	35.6	32.3	29.3	26.8
GHM with appr./deappr.	35.3	31.8	29.4	27.1
Adjacent Row Processing:				
GHM	24.1	21.3	19.7	18.4
symmetric pair	31.1	27.3	24.0	21.8
constrained pair #1	32.4	28.5	25.1	23.0
constrained pair #2	31.9	28.2	25.0	22.8

Table 4.4: Peak SNRs for compression of *Lenna*

Lenna original

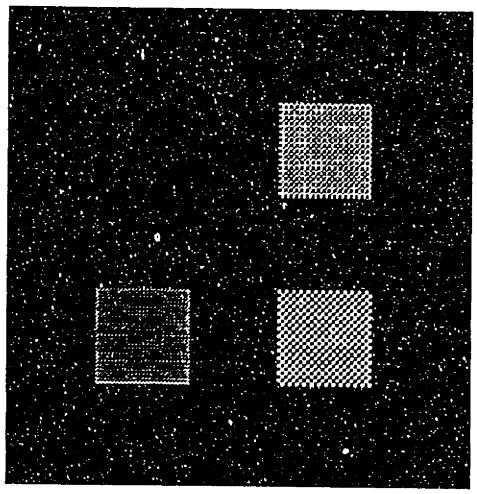
GHM approximation-based
multiwavelet compression
64:1, pSNR 27.1

Daubechies 4
scalar wavelet compression
64:1, pSNR 26.8

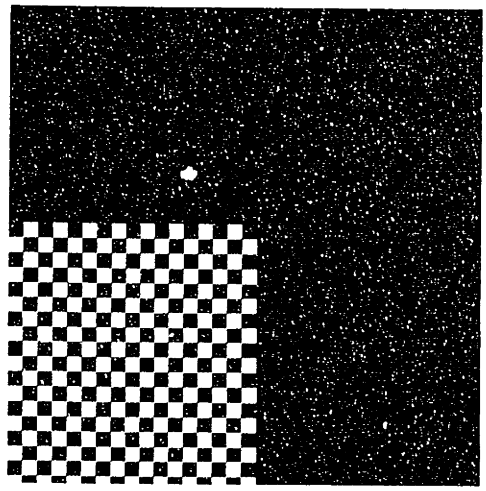
Constrained-Pair #1
multiwavelet compression
64:1, pSNR 23.0

Figure 13
Multiwavelet compression

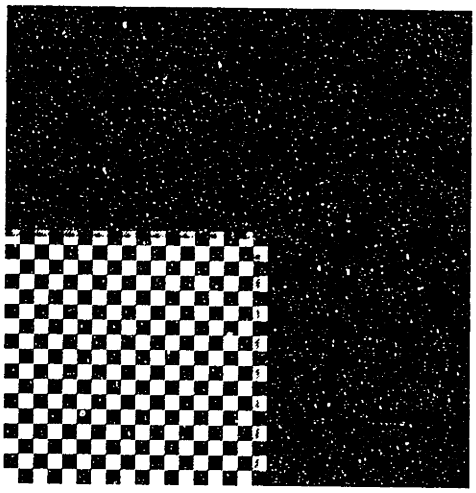
Figure 4.9: Image compression (*Lenna*)



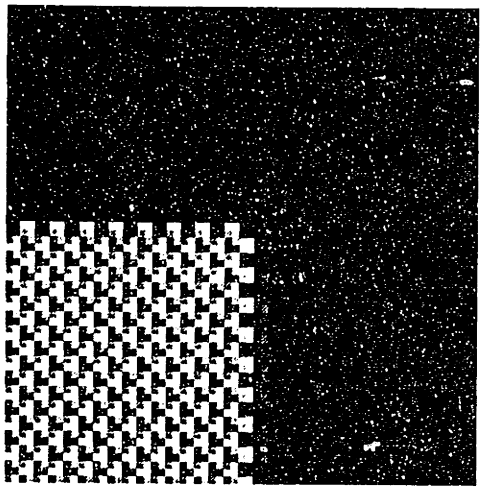
Original geometric pattern



Detail of original pattern
(corner of lower left checkerboard)



Detail of CP-1 multiwavelet
compression (32:1, pSNR 23.9)



Detail of D4 scalar wavelet
compression (32:1, pSNR 23.0)

Figure 14
Multiwavelet compression

Figure 4.10: Image compression (geometric pattern)

Compression Ratio	8:1	16:1	32:1	64:1
	pSNR	pSNR	pSNR	pSNR
Daubechies 4	48.5	31.4	23.0	19.8
GHM with appr./deappr.	52.4	34.0	18.3	16.8
Adjacent Row Processing:				
GHM	29.8	25.4	20.1	15.8
symmetric pair	33.3	28.3	20.9	16.5
constrained pair 1	42.2	32.3	23.9	19.0
constrained pair 2	33.3	30.2	21.6	17.6

Table 4.5: Peak SNRs for compression of geometric test pattern

Bibliography

[1] A. ALDROUBI, Oblique and biorthogonal multiwavelet bases, *preprint* (1996).

[2] B. ALPERT, A class of bases in L^2 for the sparse representation of integral operators, *SIAM J. Math. Analysis* **24** (1993).

[3] M. ANTONINI, M. BARLAUD, P. MATHIEU, AND I. DAUBECHIES, Image coding using the wavelet transform, *IEEE Trans. on Image Processing* **1** (1992) 205-220.

[4] G. BATTLE, A block spin construction of ondelettes. Part I: Lemarié functions, *Commun. Math. Phys.* **110** (1987) 601-615.

[5] G. BEYLKIN, R. COIFMAN, AND V. ROKHLIN, Wavelets in numerical analysis, *Wavelets and Their Applications, M.B. Ruskai ed., Jones and Bartlett, Boston* (1992).

[6] C. BRISLAWN, Classification of symmetric wavelet transforms, *Los Alamos Technical Report* (1993).

[7] A. CAVARETTA, W. DAHMEN AND C. A. MICCHELLI, Stationary Subdivision, *Memoirs Amer. Math. Soc.* **453** (1991) 1–186.

[8] C. K. CHUI, ED., Wavelets: A Tutorial in Theory and Applications, *Academic Press* (1992).

[9] C. K. CHUI AND J. A. LIAN, A study of orthonormal multi–wavelets, *Texas A&M University, CAT Report* **351** (1995).

[10] T. COOKLEV, A. NISHIHARA, M. KATO, AND M. SABLATASH, Two-channel multifilter banks and multiwavelets, *Proc. ICASSP* (1996).

[11] T. COOKLEV, M. KATO, A. NISHIHARA, AND M. SABLATASH, Multifilter banks and multiwavelet bases, *IEICE Report IE95-22* (1995).

[12] A. COHEN, Biorthogonal wavelets, *in Wvelets an Their Application, C.K. Chui ed., Academic Press* (1992).

[13] A. COHEN AND I. DAUBECHIES, A stability criterion for biorthogonal wavelet bases and their related subband coding schemes, *Duke Math. J.* **68** (1992) 313-335.

[14] A. COHEN AND I. DAUBECHIES, A new technique to estimate the regularity of refinable functions, *preprint* (1995).

[15] A. COHEN, I. DAUBECHIES, AND G. PLONKA, Regularity of refinable function vectors, *preprint* (1995).

[16] R. COIFMAN, Y. MEYER, S. QUALE, AND M. V. WICKERHAUSER, Signal processing and compression with wavelet packets, *preprint, Yale Univ.* (1990).

[17] W. DAHMEN AND C. MICCHELLI, Biorthogonal wavelet expansions, *preprint* (1995).

[18] I. DAUBECHIES, Orthonormal bases of compactly supported wavelets, *Comm. Pure Appl. Math.* **41** (1988) 909-996.

[19] I. DAUBECHIES, Ten Lectures on Wavelets, *SIAM, Philadelphia* (1992).

[20] I. DAUBECHIES AND J. LAGARIAS, Two-scale difference equations I. Existence and global regularity of solutions, *SIAM J. Math. Anal.* **22** (1991) 1388-1410; II. Local regularity, infinite products of matrices and fractals, *SIAM J. Math. Anal.* **24** (1992) 1031-1079.

[21] C. DE BOOR, R. A. DEVORE AND A. RON, Approximation from shift-invariant subspaces of $L_2(R^d)$, *Trans. Amer. Math. Soc.* **341** (1994) 787–806.

[22] D. DONOHO, De-noising by soft-thresholding, *IEEE Trans. Inform. Theory* (1994).

[23] D. DONOHO AND I. JOHNSTONE, Ideal spatial adaptation via wavelet shrinkage, *Biometrika,* **81** (1994) 425-455.

[24] G. DONOVAN, J. GERONIMO, D. HARDIN, AND P. MASSOPUST, Construction of orthogonal wavelets using fractal interpolation functions, *SIAM J. Math. Anal.* (1996).

[25] T. EIROLA, Sobolev characterization of solutions of dilation equations, *SIAM J. Math. Anal.* **23** (1992) 1015-1030.

[26] *Fed. Bureau of Investig.*, WSQ Gray-Scale Fingerprint Image Compression Specification,DRAFTED BY T. HOPPER, C. BRISLAWN, AND J. BRADLEY *IAFIS-IC-0110-v2* (1993).

[27] J. GERONIMO, D. HARDIN, AND P. MASSOPUST, Fractal functions and wavelet expansions based on several scaling functions, *J. Approx. Theory* **78** (1994) 373–401.

[28] T. N. T. GOODMAN, Characterizing pairs of refinable splines, *preprint* (1995).

[29] T. N. T. GOODMAN AND S. L. LEE, Wavelets of multiplicity r, *Trans. Amer. Math. Soc.* **342** 1994 307–324.

[30] T. N. T. GOODMAN, S. L. LEE AND W. S. TANG, Wavelets in wandering subspaces, *Trans. Amer. Math. Soc.* **338**, (1993) 639–654.

[31] A. HAAR, Zur Theorie der orthogonalen Funktionen-Systeme, *Math. Ann.* **69** (1910) 331-371.

[32] C. HEIL AND D. COLELLA, Matrix refinement equations: Existence and uniqueness, *preprint* (1994).

[33] C. HEIL, G. STRANG, AND V. STRELA, Approximation by translates of refinable functions, *Numer. Math.* (1996).

[34] L. HERVÉ, Multi–Resolution analysis of multiplicity d. Application to dyadic interpolation, *Comput. Harmonic Anal.* **1** (1994) 299–315.

[35] M. HO, Properties of the slant Toeplitz operator on $L^2(\partial \mathrm{D})$, *preprint* (1995).

[36] T. HOGAN, Stability and independence of the shifts of finitely many refinable functions, *preprint* (1996).

[37] R.-Q. JIA, S. D. RIEMENSCHNEIDER, AND D.-X. ZHOU, Approximation by multiple refinable functions, *preprint* (1996).

[38] J. KAUTSKY, An algebraic construction of discrete wavelet transforms, *preprint* (1995).

[39] J. KAUTSKY AND R. TURCAJOVÁ, Pollen product factorization and construction of higher multiplicity wavelets, *preprint* (1995).

[40] J. KAUTSKY AND R. TURCAJOVÁ, Discrete biorthogonal wavelet transforms as block circulant matrices, *preprint* (1995).

[41] A. KURDILA, T. SUN, AND P. GRAMA, Affine fractal interpolation functions and wavelet-based finite elements, *preprint* (1995).

[42] W. LAWTON, Tight frames of compactly supported wavelets, *J. Math. Phys.* **31** (1990) 1898-1901.

[43] W. LAWTON, S. L. LEE, AND Z. SHEN, An algorithm for matrix extension and wavelet construction, *Mathematics of Computation* (1996).

[44] P. G. LEMARIÉ, Une nouvelle base d'ondelettes de $L^2(R)$, *J. de Math. Pures et Appl.* **67** (1988) 227-236.

[45] J. A. LIAN, Characterization of the order of polynomial reproduction for multi-scaling functions, *Texas A&M University, CAT Report* **349** (1995).

[46] S. MALLAT, Multiresolution approximations and wavelet orthonormal bases of $L^2(\mathbf{R})$, *Trans. Amer. Math. Soc.* **315** (1989) 69–87.

[47] S. MALLAT, A theory for multiresolution signal decomposition: the wavelet representation, *IEEE Trans. PAMI* **11** (1989) 674-693.

[48] H. S. MALVAR, Signal Processing with Lapped Transforms, *Artech House* (1992).

[49] P. MASSOPUST, D. RUCH AND P. VAN FLEET, On the support properties of scaling vectors, *Texas A&M University, CAT Report* **335** (1994).

[50] Y. MEYER Principe d'incertitude, bases Hilbertiennes et algèbres d'opérateurs, *Séminaire Bourbaki* **662** (1985).

[51] Y. MEYER, Ondelettes, functions spliens et analyses graduées. *Rapport CEREMADE* **8914** (1986).

[52] C. MICCHELLI AND Y. XU, Reconstruction and decomposition algorithms for biorthogonal multiwavelets, *preprint* (1995).

[53] C. MICCHELLI AND Y. XU, Using the matrix refinement equation for the construction of wavelets on invariant sets, *Appl. Comp. Harm. Anal.* **1** (1994) 391–401.

[54] C. MICCHELLI AND Y. XU, Using the matrix refinement equation for the construction of wavelets II: smooth wavelets on [0,1], *International Series of Num. Math., Birkhäuser* **119** (1994).

[55] J. ODEGARD, M. LANG, H. GUO, R. GOPINATH, AND C. BURRUS, Nonlinear wavelet processing for enhancement of images, *IEEE SP Letters* (1996).

[56] G. PLONKA, Two-scale symbol and autocorrelation symbol for B-splines with multiple knots, *Advances in Computational Mathematics* **3** (1995) 1–22.

[57] G. PLONKA, Spline wavelets with higher defect, *Curves and Surfaces, P.J. Laurent, A.Le Méhauté, and L.L. Schumaker eds., AKPeters, Boston* (1991).

[58] G. PLONKA, Approximation order provided by refinable function vectors, *Constr. Approx.* (1996).

[59] G. PLONKA AND V. STRELA, Construction of multi-scaling functions with approximation and symmetry, *preprint, Rostock University* (1995).

[60] A. Ron and Z. Shen, Affine systems in $L_2(R^d)$: the analysis of the analysis operator, *preprint* (1996).

[61] J. Shapiro, Embedded image coding using zerotrees of wavelet coefficients, *IEEE Trans. on SP* **41** (1993) 3445-3662.

[62] Z. Shen, Refinable function vectors, *preprint* (1995).

[63] W. So and J. Wang, Estimating the support of a scaling vector, *preprint* (1996).

[64] P. Steffen, P. Heller, R. Gopinath, and C. Burrus, Theory of m-band wavelet bases, *IEEE Trans. SP* **41** (1993) 3497-3511.

[65] G. Strang, Introduction to Linear Algebra, *Wellesley-Cambridge Press* (1993).

[66] G. Strang, Eigenvalues of $(\downarrow 2)H$ and convergence of the cascade algorithm, *IEEE Trans. Sig. Proc.* **44** (1996) 233-238.

[67] G. Strang, Wavelets and dilation equations: a brief introduction, *Siam Review* **31** (1989) 613-627.

[68] G. Strang and G. Fix, An Analysis of the Finite Element Method, *Wellesley-Cambridge Press, Wellesley MA* (1973).

[69] G. Strang and T. Nguyen, Wavelets and Filter Banks, *Wellesley-Cambridge Press* (1996).

[70] G. Strang and V. Strela, Short wavelets and matrix dilation equations, *IEEE Trans. on SP.* **43** (1995) 108-115.

[71] G. Strang and V. Strela, Orthogonal multiwavelets with vanishing moments, *J. Optical Eng.* **33** (1994) 2104-2107.

[72] V. Strela, Multiwavelets: regularity, orthogonality and symmetry via two-scale similarity transform, *Studies in Appl. Math.* (1996).

[73] V. Strela, P. Heller, G. Strang, P. Topiwala, C. Heil, The application of multiwavelet filter banks to image processing, *IEEE Trans. SP* (1996).

[74] V. Strela and G. Strang, Finite element multiwavelets, *Proc. Maratea NATO Conference, Kluwer* (1995).

[75] V. Strela and G. Strang, Biorthogonal Multiwavelets and Finite Elements, *preprint* (1996).

[76] J .O STRÖMBERG, A modified Franklin systems and higher order spline systems on R^n as unconditional bases for hardy spaces, *Conf. in Honor of A. Zygmund, Wadsworth Math. Ser.* II (1982) 475-493.

[77] W. SWELDENS, The lifting scheme: A construction of second generation wavelets, *preprint, Univ. So. Carolina* (1994).

[78] P. TCHAMITCHIAN, Biorthogonalité et théorie des opérateurs, *Rev. Mat. Iberoamer.* **3** (1987) 163-189.

[79] R. TURCAJOVÁ, Factorizations of linear phase paraunitary filter banks and higher multiplicity wavelets, *preprint* (1995).

[80] G. UYTTERHOEVEN, Multiwavelets for image compression, *Masters thesis, Katholieke Universiteit Leuven* (1994).

[81] P. P. VAIDYANATHAN, Multirate Systems and Filter Banks, *Prentice-Hall, Englewood Cliffs* (1993).

[82] M. VETTERLI AND J. KOVACEVIC, Wavelets and Subband Coding, *Prentice-Hall* (1995).

[83] M. VETTERLI AND G. STRANG Time-varying filter banks and multiwavelets, *Sixth IEEE DSP Workshop, Yosemite* (1994).

[84] L. VILLEMOES, Energy moments in time and frequency for two-scale difference equation solutions and wavelets, *SIAM J. Math. Anal.* **23** (1992) 1519-1543; Wavelet analysis of refinement equations, *SIAM J. Math. Anal.* **25** (1994) 1433-1466.

[85] P. WALTERS, An Introduction to Ergodic Theory, *Springer-Verlag, New York* (1982).

[86] J. WANG, Stability and linear independence associated with scaling vectors, *preprint* (1996).

[87] J. WILLIAMS AND K. AMARATUNGA, $O(n)$ multiscale wavelet solvers with arbitrary high accuracy, *MIT IESL Report 93-14* (1993).

[88] J. WILLIAMS AND K. AMARATUNGA, High order wavelet extrapolation schemes for initial value problems and boundary value problems, *MIT IESL Report 94-07* (1994); also *SPIE Proc.* **2491** (1995) 894-902.

[89] X.-G. XIA, J. GERONIMO, D. HARDIN, AND B. SUTER, Computations of multiwavelet transforms, *Proc. SPIE 2569, San Diego, CA* (1995).

[90] X.-G. XIA AND B. SUTER, Multirate filter banks with block sampling, *preprint* (1994).

[91] W. ZETTLER, J. HUFFMAN, AND D. LINDEN, The application of compactly supported wavelets to image compression, *Proc. SPIE* **1244** (1990) 150-160.