Matrix-Nullspace Wavelet Construction

Subendran Sivalingam

September 30, 1994

Abstract

"Wavelets" have been a very popular topic of conversation in many scientific and engineering gatherings these days. Some view wavelets as a new basis for representing functions, some consider it a technique for time-frequency analysis, and others think of it as a new mathematical subject. All of them are right, since "wavelet theory" is a versatile tool with very rich mathematical content and great potential for applications, such as data compression. On the other hand polynomial spline functions are among the simplest functions for both computational and implementational purposes. Hence, they are most attractive for analyzing and constructing wavelets.

This thesis explores a simple, general tool for constructing wavelets from "box splines," which are natural generalization of B-splines. This tool involves use of the inner product, a matrix formulation, an associated homogeneous system of equations and the determination of null space. This tool is applicable for both univariate and multivariate settings.

Contents

1 Introduction **1**
- 1.1 Motivation and Related work 3

2 An Overview of Wavelets **5**
- 2.1 The Integral Wavelet Transform 5
- 2.2 Multiresolution Analysis and Wavelets 9
- 2.3 Decomposition and Reconstruction Sequences 12
- 2.4 Pyramid Algorithms and Linear Phase Filters 18

3 Box Splines **22**
- 3.1 Bivariate Box Splines 22
- 3.2 Multivariate Box Splines 27

4 Compactly Supported Box Spline Wavelets **35**
- 4.1 Box Splines and Multiresolution Analysis 35
- 4.2 Construction of Wavelet Decompositions 38
- 4.3 Construction of Wavelets from Cardinal Splines 40

5 Construction of Wavelets from Box Splines **52**
- 5.1 Bivariate Three Directional Box splines Wavelets 52
- 5.2 Analysis of The Geometric Representation of Matrix $[B]$ . . . 60
  - 5.2.1 Derivation of Permutation Matrices $[X]$ and $[Y]$ 62

6 Conclusion **72**

A Example 1 in Chapter 4 **74**

B Example 2 in Chapter 4 **81**

C Example 3 in Chapter 4 **87**

D Example 1 in Chapter 5 **91**

E Example 2 in Chapter 5 **103**

List of Tables

4.1 Results of Examples 1 and 2 50
4.2 Result of Example 3 51

5.1 Result of Example 1 70
5.2 Result of Example 2 71

List of Figures

3.1 The support of the Courant finite element 32
3.2 Two, three and four directional grids 32
3.3 The support of some box splines 33
3.4 Various box spline grids 34

4.1 The geometric representation for computation of [B] 49

5.1 Indices of p_{r_1,r_2}, q_{k_1,k_2} and $b_{\theta_{k_1},\theta_{k_2}}$ 64
5.2 The supports of $p_{\theta_{r_1},\theta_{r_2}}$ and $\bar{N}(i,j \mid \Xi \bigcup \Xi)$ 65
5.3 The geometric representation for computation of $b_{\theta_{k_1},\theta_{k_2}}$ . . . 66
5.4 The geometric representation for computation of [B]: case $m_1 < m_2$ 67
5.5 The geometric representation for computation of [B]: case $m_1 > m_2$ 68
5.6 A useful transformation for determining the permutation matrices [X] and [Y] 69

Chapter 1

Introduction

While spline functions have been used successfully for analyzing and approximating both exact and noisy data, the technique of Fourier transformation is the standard tool for studying the corresponding spectral behavior in the frequency domain. Indeed, for many applications, such as image and signal analyses, only spectral information can be observed. Although the Fourier techniques are very powerful, there is a serious deficiency in the integral Fourier representation: time-evolution of frequencies is not reflected. In other words, the formulation of the Fourier transform requires global information of the function in the time-domain. This shortcoming was observed by D. Gabor who, in his 1946 paper [19], introduced a time and frequency localization method by applying the Gaussian function to "window" the Fourier integral. Other window functions have been studied since then, and this method is usually called the *windowed Fourier transform* or *short time Fourier transform* (STFT). There are still defects in all of the STFT methods, mainly due to a undesirable computational burden arising from repeatedly having to narrow the window for good localization and then having to wide the window for a more global picture.

The integral wavelet transform (IWT), on the other hand, has the capability of simultaneously "zooming-in" on short-lived high-frequency phenomena and "zooming-out" for low-frequency observations. Hence, the IWT is suitable for a wide variety of applications such as radar, sonar, acoustics, edge-detection, etc. This transform has its origin in seismic analysis, and the first published work was by Grossman and Morlet [21] in 1984. IWT is based on the idea of analysing a signal with dilations and translations of a

single window function, known as the *basic wavelet*. Dilation corresponds to a change of frequency, while translation localizes time or position. In addition to being a window function as in STFT, the basic wavelet ψ, must satisfy a very mild condition: namely it must have zero mean. Since the wavelets must also have somewhat concentrated mass, it behaves like a "small wave", and so the terminology of "wavelet", or "ondelette" in French, is quite appropriate. The zero-mean property of ψ , or equivalently the vanishing of its Fourier transform at the origin, $\hat{\psi}(0) = 0$, can be weakened a little to $\hat{\psi}(w)/ \mid w \mid^{1/2}$ being in L^2. Therefore, the idea and techniques of IWT can be traced back to the work of Calderon [3] in 1964 on singular integral operators. In particular, an integral reproducing formula allows us to reconstruct the function from its IWT.

Such a reconstruction requires global information of the IWT at all dilations and translations. If the basic wavelet ψ is orthonormal (o.n.), and by this we mean that $\{\psi(\cdot - n) : n \in Z\}$ is an orthonormal family, i.e., $\{\psi(\cdot - n)\}$ is orthonormal in both dilation and translation. Then under very mild conditions on ψ, a "wavelet series" can also be used to recover the original function. This observation revolutionizes certain aspects of harmonic analysis in that instead of Fourier series, which represents periodic functions, we now have an orthogonal series, which represents functions defined on the real line. The importance of this representations is that with a good choice of the basic o.n. wavelet ψ, we have a representation that localizes both time and frequency.

There are many o.n. wavelets in the literature. The oldest one is the Haar function which unfortunately cannot be used to localize frequency. Orthonormal wavelets that are less well known but have been influential to the development of the subject of wavelets are given by Stromberg [33], Meyer [27], Lemarie [23] , and Battle [2]. The o.n. wavelets ψ that have had the greatest impact on this subject are the compactly supported o.n. wavelets introduced by Daubechies [15] in 1988.

More recently, Chui and Wang [11, 10] gave yet another approach to construct a function from its IWT, again by a wavelet series. This wavelet ψ, however, does not have to be orthonormal as described above, but still gives an *orthogonal* wavelet decomposition of the function. The idea is to introduce a dual wavelet $\tilde{\psi}$ of ψ, namely:$< \psi, \tilde{\psi} >= 0$. The advantage of this approach is that it gives us more freedom to construct wavelets with other desirable properties. Perhaps the most important desirable property, at least in many applications to signal analysis, is the property of *linear*

phase. Linear phase requires the wavelet to be symmetric or antisymmetric. Since compactly supported o.n. wavelets different from the Haar function can not be symmetric or antisymmetric (an important result established by Daubechies [15]) we have to give up orthogonality at least within the same scale levels if a compactly supported continuous wavelet with linear phase is desired. Such a sacrifice is worthwhile in many situations since the extra freedom also allows us to give explicit expressions for the wavelets. In fact, even compactly supported polynomial spline wavelets with linear phase have very simple expressions (cf. Chui and Wang [11]). However, Daubechie's approach for constructing o.n. wavelets and Chui's and Wang's approach for constructing symmetric or antisymmetric wavelets involve us deeply the rigorous of functional analysis.

The applications of wavelets, whether orthonormal or not, go far beyond the IWT and the recovery of functions from their IWT. The main reason is that very efficient algorithms, usually called pyramid algorithms, introduced by Mallet [24] in 1988 yield orthogonal wavelet decompositions and reconstructions in $O(n \log\ n)$ time. A wavelet decomposition separates and localizes the spectral information in different frequency bands (or octaves), and hence filtering, detection, data reduction, enhancements, etc., can be easily implemented before applying the wavelet reconstruction algorithm. Both the decomposition and reconstruction algorithms utilize the formulas that describe the intimate relationship between the wavelet of interest and the "spline" function that is used for approximation. This relationship has a very beautiful mathematical description, called the *multiresolution analysis*, introduced by Meyer [29] and Mallet [24] .

Wavelets also have great promises in other areas of mathematics such as approximation theory, harmonic analysis, operator theory, and numerical partial differential equations. We will not go into the discussion of these aspects but refer the interested reader to the literature listed in the references. In particular, we would like to mention Meyer's two volumes [28] and Daubechies' CBMS monograph [16] as well as Chui's introductory text [5].

1.1 Motivation and Related work

In the above section, we briefly discussed the history of wavelets, how the IWT distinguishes itself from the Fourier transform, the concept of the basic

wavelet, the o.n. wavelets, Chui's and Wang's approach to construct the wavelets (not orthonormal) and desirable properties of wavelets such as linear phase. Now we pause and ask the following questions:

- Is there any other simple method, not involving the rigorous of functional analysis, available to construct the compactly supported wavelets?
- Is this method applicable to both the univariate and multivariate settings ?
- Does this method produce the wavelets with the desirable property of linear
  phase ?
- Is it possible to produce the wavelets corresponding to different dilated scale levels by applying this method ?

All the goals related to above questions are accomplished in our *Matrix Nullspace* method. Our method involves only use of the inner product, a matrix formulation, an associated homogeneous system of equations and the determination of the null space, in contrast other functional analysis approaches.

An overview of wavelets, including the integral wavelet transform, multiresolution analysis and wavelet algorithms for decomposition and reconstruction of functions, is presented in the second chapter. This thesis is intended to be self contained. The only prerequisite is a basic knowledge of box spline theory and elementary numerical linear algebra. For this reason, an introductory study of box splines is presented in Chapter 3. In Chapter 4, the basic steps of our approach and the construction of cardinal spline wavelets are discussed and some numerical examples are given. We study the construction of box spline wavelets and illustrate this construction with some numerical examples in Chapter 5. In Chapter 6, we summarize our work and mention possible future research. The writing of this thesis was greatly influenced by the pioneering work of C.K. Chui, J. Stockler, J.D. Ward and I. Daubechies in wavelets, as well as the research work of T. Lyche and M. Daehlen in box splines.

Chapter 2

An Overview of Wavelets

This chapter presents the integral wavelet transforms, multiresolution analysis and wavelet algorithms for decomposition and reconstruction of functions. The objective of this chapter is not to go into any depth but only to convey a general impression of what wavelet analysis is about. In this chapter, we have restricted our attention to functions of one variable, although some of the ideas, techniques and results easily carry over to the multi-dimensional setting. The material presented in this chapter also includes C. K. Chui's key ideas for constructing wavelets. The reader is advised to refer to Chui's introductory text [6] for further study.

2.1 The Integral Wavelet Transform

The Fourier transform

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

of a sufficiently well-behaved function $f(t)$ contains a lot of information of the "rule" governed by the function $f(t)$. To expedite our discussion, we will simply assume that $f \in \mathcal{L}^1 \bigcap \mathcal{L}^2$, where $\mathcal{L}^p = \mathcal{L}^p(\mathcal{R}) =$ The $p-$integrable functions of a real variable. If a physical phenomenon is described by $f(t)$ where t may be thought of as position in space, then $\hat{f}(\omega)$ describes the spectral behavior of this phenomenon in the momentum space with variable ω. If, on the other hand, $f(t)$ represents a signal in the time variable, then the spectral information given by $\hat{f}(\omega)$ describes both the magnitude and

the phase of the signal at frequency ω. In order to present a unified point of view, the function $f(t)$ will always be considered as a time signal in the following discussion.

Observe that to define $\hat{f}(\omega)$, we need $f(t)$ for all $t \in \mathcal{R}$. This is an ineffective way to study the spectral behavior. For instance, the time-evolution of a signal whose spectrum changes with time in a significant way is certainly not reflected in the Fourier transform. This motivates the introduction of a window function to the Fourier integral. In general a function $g(t) \in \mathcal{L}^2(\mathcal{R})$ is called window function if $tg(t) \in \mathcal{L}^2(\mathcal{R})$ and must have finite width, which is defined to be $2\Delta_g$, where

$$\Delta_g = \left[\frac{\int_{-\infty}^{\infty} (t - t_0) \mid g(t) \mid^2 dt}{\int_{-\infty}^{\infty} \mid g(t) \mid^2 dt} \right]^{\frac{1}{2}} \tag{2.1}$$

is the standard deviation of $g(t)$ with t_0 being the "center" of the window function, defined by

$$t_0 = \frac{\int_{-\infty}^{\infty} t \mid g(t) \mid^2 dt}{\int_{-\infty}^{\infty} \mid g(t) \mid^2 dt}. \tag{2.2}$$

In order to window both time and frequency, the Fourier transform of a window function must be a window function also. A perfect example is the Gaussian function which is "invariant" under the Fourier transform. This was actually the original window function that Gabor used [19].

It should be remarked that no matter which window function $g(t)$ is used, the product of the widths of the time-window $g(t)$ and the frequency-window $g(\omega)$ is bounded below by $\frac{1}{4\pi}$; that is

$$\Delta_g \Delta_{\hat{g}} \geq \frac{1}{4\pi}, \tag{2.3}$$

and equality is attained only by the Gaussian function. This is the so-called Heisenberg Uncertainty Principle. Hence, if a desirable window function is chosen so that the size of the frequency-window is appropriate to localize certain frequencies (such as high enough frequency for edge-detection), then the time-window can not be too narrow. Consequently very high-frequency signals can not be accurately identified. Furthermore, since Δ_g and $\Delta_{\hat{g}}$ are unchanged when $g(t)$ and $\hat{g}(\omega)$ are shifted, the same time-frequency window is used for both high-frequency and low-frequency signals. In many applications, such as edge detection, the time-window must be very narrow at a

very high-frequency band (the location of an edge) for accuracy, and very wide at a low-frequency band for efficiency. The STFT does not have this property and its performance suffers.

The integral wavelet transform (IWT) introduced by Grossman and Morlet [21] has the so-called zoom-in and zoom-out effect. It automatically zooms in at high-frequencies (with a very narrow time-window) and zooms out at low-frequencies (with a very wide time-window). More precisely, let $\psi(t)$ be a window function such that its Fourier transform $\hat{\psi}(\omega)$ is also a window function (i.e. Δ_ψ , $\Delta_{\hat{\psi}}$ are both required to be finite). In addition, we also assume that $\hat{\psi}(0) = 0$ and that the center of the frequency-window $\hat{\psi}$ is non-zero. The reasons for these two extra conditions will be clear later. Let us first pause for a moment to observe that $\hat{\psi}(0) = 0$ is equivalent to the fact that ψ has zero mean, so that since ψ is a window function, it behaves like a wave packet. This is why ψ is called a wavelet. The wavelet transform, with the wavelet ψ as a window function, is defined to be

$$(W_\psi f)(b,a) := \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} f(t) \overline{\psi\left(\frac{t-b}{a}\right)} dt \tag{2.4}$$

where $a > 0$ and $b \in \mathcal{R}$. We will assume, without loss of generality, that the window ψ is centered at the origin (cf. Equation 2.2 for the definition of the center). By the convolutional duality of Fourier transform , we have

$$(W_\psi f)(b,a) = \frac{\sqrt{a}}{2\pi} \int_{-\infty}^{\infty} e^{ib\omega} \hat{f}(\omega) \overline{\hat{\psi}(a\omega)} d\omega. \tag{2.5}$$

If $g(\omega) := \hat{\psi}(\omega + \omega_0)$, then $g(\omega)$ is a window function with center at 0, and

$$g\left(a\omega - \omega_0\right) = \hat{\psi}(a\omega). \tag{2.6}$$

That is, with the exception of the multiplicative constant $\sqrt{a}$, the wavelet transform $(W_\psi f(b,a))$ agrees with the window inverse Fourier transform evaluated at $t = b$, with the window function given by Equation 2.6, which is centered at $\omega = \frac{\omega_0}{a}$ and has width given by $2/a\Delta_{\hat{\psi}}$. Since the wavelet transform already windows the function $f(t)$ with center at b and width given by $2a\Delta_\psi$, we now have a time-frequency localization, where the time-frequency window in the $t - \omega$ plane is given by

$$[b - a\Delta_\psi, b + a\Delta_\psi] \times \left[\tfrac{\omega_0}{a} - 1/a\Delta_{\hat{\psi}}, \tfrac{\omega_0}{a} + 1/a\Delta_{\hat{\psi}}\right] \tag{2.7}$$

Now, we see that it is important to center ψ at $\omega_0 \neq 0$ (and for practical purposes with real signal, since only positive frequency is important, we should pick $\omega_0 > 0$). If we identify $\frac{1}{a}$ with a constant multiple of the frequency, it follows from Equation 2.7 that the time-window narrows at high frequencies and widens at low frequencies. This is the zoom-in and zoom-out capability of the wavelet transform. For more details see the works of Grossmann, Martinet and Morlet [20] and Chui [5].

Localization of time and frequency of a signal not only makes it possible for filtering, detection, enhancement, etc., but also facilitates tremendously the process of reduction of the signal information for the purpose of transmission or storage. In any case, the modified, enhanced, or reduced signal must be reconstructed again. That is, we need a formula to recover $f(t)$ from the wavelet transform $(W_\psi)f(b,a)$ in Equation 2.5. Here, the condition $\hat{\psi}(0) = 0$ is needed, or at least the slightly weaker version of it is needed, namely:

$$C_\psi := \int_0^\infty \frac{\mid \hat{\psi}(\omega) \mid^2 d\omega}{\mid \omega \mid} < \infty. \tag{2.8}$$

Indeed, if Equation 2.8 holds and ψ is real-valued, then for any real-valued $f \in \mathcal{L}^1 \bigcap \mathcal{L}^2$, we have

$$f(t) = \frac{2\pi}{C_\psi} \int_0^\infty \left[\int_{-\infty}^\infty (W_\psi f)(b,a) \psi \left(\frac{(t-b)}{a} \right) db \right] \frac{da}{a^{5/2}} \tag{2.9}$$

(cf. Grossmann, Martinet and Morlet [20], Chui [5], and Daubechies [16]).

This elegant formula is not very useful, since the integral has to be approximated by a finite sum. Hence, it is preferable to have a series representation instead. A series representation, as we will see, not only reconstructs $f(t)$ via what will be called a *reconstruction algorithm*, it also decomposes $f(t)$ into components with the wavelet transforms as coefficients. More precisely, if a good approximant $f_N(t)$ of $f(t)$ is taken, then the wavelet transform $(W_\psi f_N)(b,a)$ at the dyadic values $b = j/2^k$ and $a = 2^{-k}$ $j,k \in \mathcal{Z}$ is easily determined via what will be called a *decomposition algorithm*. Both the decomposition and reconstruction algorithms are very efficient pyramid algorithms introduced by Mallet [24]. They can be easily implemented on the computer. Since only simple moving-average operations are required, it should not be difficult to produce computer chips for wavelet decompositions and reconstructions.

To give a series representation with the wavelet transform $(W_\psi)(b,a)$ at dyadic values as coefficients, we have to choose the wavelet window function ψ properly. In particular, the collection of $\psi_{k,j}$, obtained from integer translates by j and dyadic dilations by 2^k of ψ, where $j,k \in \mathcal{Z}$, should be dense in $\mathcal{L}^2$. If $\{2^{k/2}\psi_{k,j}\}$, $j,k \in \mathcal{Z}$ is also orthonormal, then we obviously have the desirable result. If $\psi_{k,j}$ is only orthogonal to $\psi_{l,m}$, for $l \neq k$, then a "dual" $\tilde{\psi}$ of ψ is required [10], and if no orthogonality is given, then a "biorthogonal basis" is needed [13]. A more complete mathematical formulation of these concepts will be given later by using the notion of multiresolution analysis, introduced by Meyer [29] and Mallet [24].

2.2 Multiresolution Analysis and Wavelets

We again restrict our attention to the $\mathcal{L}^2$ space, although our discussion applies to a much more general setting.

A nested sequence of closed subspaces

$$\cdots \subset V_{-1} \subset V_0 \subset V_1 \subset \cdots$$

of $\mathcal{L}^2$ is said to form a *multiresolution analysis* of $\mathcal{L}^2$ if the following conditions are satisfied:

(i)
$$clos_{\mathcal{L}^2}\left(\bigcup_{n\in\mathcal{Z}} V_n\right) = \mathcal{L}^2;$$

(ii)
$$\bigcap_{n\in\mathcal{Z}} V_n = \{0\};$$

(iii) *for each* $n \in \mathcal{Z}$,
$$f \in V_n \;\Leftrightarrow f(2\cdot) \in V_{n+1};\ \textit{and}$$

(iv) *there exists a* $\phi \in V_0$ *such that* $\{\phi(\cdot - n) : n \in \mathcal{Z}\}$ *is an unconditional basis of* V_0.

We remark that by defining

$$\phi_{k,j}(t) = \phi(2^k t - j); \quad k,j \in \mathcal{Z}, \tag{2.10}$$

it is clear that if (iv) holds, then $\{\phi_{k,j} : j \in \mathcal{Z}\}$ is also an unconditional basis of V_k; by which we mean, there exist positive constants A and B such that

$$2^k A \| \{c_n\} \|_{l^2}^2 \leq \| \sum_{n \in \mathcal{R}} c_n \phi_{k,n} \|_{L^2}^2 \leq 2^k B \| \{c_n\} \|_{l^2}^2, \tag{2.11}$$

for all $\{c_n\} \in l^2$ and $k \in \mathcal{Z}$.
The most typical examples are:

1. The mth order B-spline N_m, with integer knots and supp $N_m = [0, m]$, whose integer translates and dyadic dilations $N_{m;k,j}$, as defined by Equation 2.10 , span the closed spline subspaces V_k, with knots at $2^{-k}\mathcal{Z}$; and

2. the compactly supported ${}_N\phi$, with orthonormal integer translates ${}_N\phi(\cdot - n) : n \in \mathcal{Z}$, constructed by Daubechies [15].

Later, we will refer to these two examples quite frequently.

For each $k \in \mathcal{Z}$, let W_k be the orthogonal complementary subspace of V_{k+1} relative to V_k; that is, $V_{k+1} = V_k + W_k$ and $W_k \perp V_k$. The notation

$$V_{k+1} = V_k \oplus W_k \tag{2.12}$$

will be used to denote the orthogonal complement. Then it clear that

$$W_k \perp W_n, \text{ all } k \neq n, \tag{2.13}$$

and from (i) and (ii), we also have the infinite sum

$$\mathcal{L}^2 = \bigoplus_{k \in \mathcal{Z}} W_k \tag{2.14}$$

Hence, every $f \in \mathcal{L}^2$ has a (unique) orthogonal decomposition into an infinite number of terms:

$$f = \sum_{k \in \mathcal{Z}} g_k, \quad g_k \in W_k \tag{2.15}$$

Now suppose that a wavelet window function as defined in Section 1 can be chosen from W_0. Then for any $k, j \in \mathcal{Z}$, we have

$$(W_\psi f)\left(\frac{j}{2^k}, 2^{-k}\right) = 2^{\frac{k}{2}} \langle f, \psi_{k,j} \rangle \tag{2.16}$$

This allows us to make the following observations on representing any function $f \in \mathcal{L}^2$ as a "wavelet series" with coefficients given by the integral wavelet transform of f at the dyadic values $j/2^k$ in Equation 2.16.
(1°) Suppose that $\psi \in W_0$ satisfies:

$$\langle \psi_{k,j}, \psi_{n,l} \rangle = 2^{-k} \delta_{k,n} \delta_{j,l} \tag{2.17}$$

for all $k, j, n, l \in \mathcal{Z}$. Then

$$\begin{aligned} f &= \sum_{k,j \in \mathcal{Z}} 2^k \langle f, \psi_{k,j} \rangle \psi_{k,j} \\ &= \sum_{k,j \in \mathcal{Z}} 2^{\frac{k}{2}} (W_\psi f) \left(\frac{j}{2^k}, 2^{-k} \right) \psi_{k,j} \end{aligned} \tag{2.18}$$

for all $f \in \mathcal{L}^2$.
(2°) Suppose that $\psi \in W_0$, and $\tilde{\psi} \in W_0$ is its "dual" in the sense that

$$\langle \psi, \tilde{\psi}(\cdot - n) \rangle = \delta_{n,0}, \ n \in \mathcal{Z} \tag{2.19}$$

Then

$$\begin{aligned} f &= \sum_{k,j \in \mathcal{Z}} 2^k \langle f, \psi_{k,j} \rangle \tilde{\psi}_{k,j} \\ &= \sum_{k,j \in \mathcal{Z}} 2^{\frac{k}{2}} (W_\psi f) \left(\frac{j}{2^k}, 2^{-k} \right) \tilde{\psi}_{k,j}, \end{aligned} \tag{2.20}$$

where it is clear that ψ and $\tilde{\psi}$ can be interchanged.

In view of the above observations, it is natural to call the orthogonal subspaces W_k wavelet spaces, and since the orthogonal decomposition given by Equation 2.15 is achieved by using Equation 2.18 or 2.20, with

$$g_k = \sum_{j \in \mathcal{Z}} 2^k \langle f, \psi_{k,j} \rangle \psi_{k,j}$$

if ψ satisfies Equation 2.17, or

$$g_k = \sum_{j \in \mathcal{Z}} 2^k \langle f, \psi_{k,j} \rangle \tilde{\psi}_{k,j}$$

if $\tilde{\psi}$ is the dual of ψ in the sense of Equation 2.19, we will call Equation 2.15 a *complete wavelet decomposition.* If ψ satisfies Equation 2.17, it is called an

orthonormal (or o.n.) *wavelet.* Of course, any o.n. wavelet is self-dual in the sense that $\tilde{\psi} = \psi$. In general any $\psi \in W_0$ that provides a wavelet window function is called a *wavelet*. In choosing a wavelet ψ, it is important to have finite values of both Δ_ψ and $\Delta_{\hat{\psi}}$, since ψ is used to localize time and $\hat{\psi}$ to localize frequency (cf. Equations 2.4, 2.5 and 2.7), and for optimal efficiency in time-frequency localization, $\psi \in W_0$ should be chosen to attain

$$\Delta_\psi \Delta_{\hat{\psi}} = \inf \{\Delta_\eta \Delta_{\hat{\eta}} : \eta \in W_0\} \tag{2.21}$$

A characterization of all wavelets for an arbitrary multiresolution analysis has recently been given by Chui and Wang [10], and so this minimization problem seems to be solvable, at least numerically.

Dual wavelets ψ were first introduced by Chui and Wang for polynomial splines [11], and for the general setting [10]. For the polynomial B-splines N_m of order m as in the above example (1), an "interpolatory" wavelet is given by Chui and Wang [9] as

$$\eta_m = L_{2m}^{(m)}(2 \cdot -1) = m^{th} \text{ derivative of } L_{2m}, \tag{2.22}$$

where L_{2m} is the $(2m)^{th}$ order fundamental spline with integer knots defined by $L_{2m}(n) = \delta_{n,0}$; and the wavelet with minimum support is given by Chui and Wang [11] as

$$\psi_m = \frac{1}{2^{m-1}} \sum_{j=0}^{2m-2} (-1)^j N_{2m}(j+1) \sum_{l=0}^{m} (-1)^l \begin{pmatrix} m \\ l \end{pmatrix} N_m(2 \cdot -j-l). \tag{2.23}$$

Note that the wavelet ψ_m is an mth order spline with supp $\psi_m = [0, 2m-1]$. The duals $\tilde{\eta}_m$ and $\tilde{\psi}_m$ of η_m and ψ_m, respectively, can be easily calculated using the two scale sequence as the decomposition sequence (to be defined in the next section) as shown in [9] and [10].

2.3 Decomposition and Reconstruction Sequences

The main advantage of a wavelet series representation, such as the wavelet specified by Equation 2.18 for an o.n. wavelet, or Equation 2.20 for an arbitrary wavelet ψ with dual $\tilde{\psi}$ over the integral representation given by Equation 2.9 is that a decomposition algorithm can be applied to find the

integral wavelet transform in Equation 2.16 at the dyadic values, and a reconstruction algorithm can be used to recover the function from these values of the integral wavelet transform. For decomposition, we need *decomposition sequences*, and for reconstruction, the two scale sequences are required. For this reason, the two scale sequences are also called reconstruction sequences. Let us now discuss the significance of these sequences.

Let $\phi \in V_0$ generate the given multiresolution analysis. Then since $V_0 \subset V_1$ and $W_0 \subset V_1$, it follows from (iii) in Section 2 that, for any wavelet $\psi \in W_0$, there exist two l^2-sequences $\{p_n\}$ and $\{q_n\}$, such that ϕ and ψ satisfy:

$$\phi(t) = \sum_{n \in \mathcal{R}} p_n \phi(2t - n); \tag{2.24}$$

$$\psi(t) = \sum_{n \in \mathcal{R}} q_n \phi(2t - n); \tag{2.25}$$

Furthermore, since $\{\phi_{1,n}\}$ is an unconditional basis of V_1, these two sequences are unique. Equations 2.24 and 2.25 are called the *two-scale formulas* for ϕ and ψ, respectively; and the two sequences $\{p_n\}$ and $\{q_n\}$ are called their corresponding *two-scale (or reconstruction) sequences*. Next, since $V_1 = V_0 + W_0$, we also have two l^2-sequences $\{a_n\}$ and $\{b_n\}$ such that ϕ and ψ satisfy:

$$\phi(2t - l) = \sum_{n \in \mathcal{Z}} a_{l-2n} \phi(t - n) + \sum_{n \in \mathcal{Z}} b_{l-2n} \psi(t - n) \tag{2.26}$$

for all $l \in \mathcal{Z}$. The uniqueness of $\{a_n\}$ and $\{b_n\}$ follows from the orthogonality of $\phi(\cdot - n)$ with $\psi(\cdot - m)$ and the fact that ϕ and ψ yield unconditional bases of V_0 and W_0, respectively. Equation 2.26 is called the *decomposition formula* for ϕ and ψ, and the sequences $\{a_n\}$ and $\{b_n\}$ are called their corresponding *decomposition sequences*. This generalization of the orthonormal decomposition and reconstruction was first introduced by Chui and Wang [9] for polynomial splines and wavelets (see also Chui and Wang [10] for the general setting).

Perhaps the easiest way to characterize these two pairs of sequences ($\{p_n\}$, $\{q_n\}$) and ($\{a_n\}$, $\{b_n\}$) is to consider the Fourier transform formulations of the two-scale and decomposition formulas. Hence, we need the "symbols" or Z-transforms of the sequences. For notational convenience (mainly due to

the two-scale property), let us set

$$\begin{cases} P(z) = \frac{1}{2}\sum_{n\in\mathcal{Z}} p_n z^n \\ Q(z) = \frac{1}{2}\sum_{n\in\mathcal{Z}} q_n z^n \end{cases} \tag{2.27}$$

and

$$\begin{cases} G(z) = \sum_{n\in\mathcal{Z}} a_n z^{-n} \\ H(z) = \sum_{n\in\mathcal{Z}} b_n z^{-n}. \end{cases} \tag{2.28}$$

Then using the Fourier transforms, Equations 2.24 and 2.25 become

$$\begin{cases} \hat{\phi}(\omega) = P(e^{-i\frac{\omega}{2}})\hat{\phi}(\frac{\omega}{2}) \\ \hat{\psi}(\omega) = Q(e^{-i\frac{\omega}{2}})\hat{\psi}(\frac{\omega}{2}), \end{cases} \tag{2.29}$$

and Equation 2.26 becomes

$$\begin{cases} \hat{\phi}(\frac{\omega}{2}) = \left[G(e^{-i\frac{\omega}{2}}) + G(-e^{-i\frac{\omega}{2}})\right]\hat{\phi}(\omega) + \left[H(e^{-i\frac{\omega}{2}}) + H(-e^{-i\frac{\omega}{2}})\right]\hat{\psi}(\omega) \\ \hat{\phi}(\frac{\omega}{2}) = \left[G(e^{-i\frac{\omega}{2}}) - G(-e^{-i\frac{\omega}{2}})\right]\hat{\phi}(\omega)] + \left[H(e^{-i\frac{\omega}{2}}) - H(-e^{-i\frac{\omega}{2}})\right]\hat{\psi}(\omega) \end{cases} \tag{2.30}$$

In particular, Equations 2.29 and 2.30 give the pair of identities:

$$\begin{cases} P(z)G(z) + Q(z)H(z) = 1, \quad |z| = 1; \\ P(z)G(-z) + Q(z)H(-z) = 0, \quad |z| = 1, \end{cases} \tag{2.31}$$

which must be satisfied by $\{P, Q\}$ and $\{G, H\}$ (cf. Chui and Wang [11, 10]).

For orthonormal ϕ and ψ (with appropriate shift of ψ), it can be shown that, for $|z| = 1$,

$$\begin{cases} Q(z) &= z\overline{P(-z)}; \\ G(z) &= \overline{P(z)}; \\ H(z) &= \bar{z}P(-z), \end{cases} \tag{2.32}$$

(cf. Chui and Wang [10]), so that Equation 2.31 becomes a single identity

$$|P(z)|^2 + |P(-z)|^2 = 1, \quad |z| = 1. \tag{2.33}$$

From Equation 2.32, we can write down all other sequences in terms of $\{p_n\}$, namely:

$$\begin{cases} q_n &= (-1)^n \bar{p}_{1-n} \\ a_n &= 1/2\bar{p}_n \\ b_n &= 1/2(-1)^n p_{1-n} \end{cases} \tag{2.34}$$

(cf. Mallet [25] and Daubechies [15]). For orthonormal ϕ and ψ, all we need is to find $\{p_n\}$.

Let us now return to the general setting where ϕ and ψ are not required to be orthonormal, and recall that to compute the sequences ($\{p_n\}$, $\{q_n\}$) and ($\{a_n\}$, $\{b_n\}$), it is sufficient to determine the functions (P,Q) and (G,H), defined in Equations 2.27 and 2.28 respectively. We are only interested in ϕ with finite two-scale sequences, namely:

$$\phi(t) = \sum_{n=0}^{N_\phi} p_n \phi(2t-n); \quad p_0, p_{N_\phi} \neq 0.$$

Hence, it is not difficult to see that supp $\phi = [0, N_{\phi+1}]$ (cf. Daubechies [15]). Again the two typical examples are:

1. For the mth order B-spline N_m, the two-scale (or reconstruction) sequence p_n is given by

$$p_n = 2^{-m+1} \begin{pmatrix} m \\ n \end{pmatrix}, \quad 0 \leq n \leq m$$

   and $p_n = 0$ otherwise: and

2. for the orthonormal ${}_2\phi$ of Daubechies, we have

$$\begin{aligned} p_0 &= 1/4(1+\sqrt{3}) \\ p_1 &= 1/4(3+\sqrt{3}) \\ p_2 &= 1/4(3-\sqrt{3}) \\ p_3 &= 1/4(1-\sqrt{3}) \end{aligned}$$

   and $p_n = 0$ otherwise. (The sequences $\{p_n\}$ for ${}_N\phi$, $N = 2, \cdots, 10$, are listed in Daubechies [[15]; p.980]. Here, a multiplicative factor of $\sqrt{2}$ is used since we have a different normalization than Daubauchies.)

The key idea to constructing the other three sequences $\{q_n\}$, $\{a_n\}$ and $\{b_n\}$ from $\{p_n\}$ is the *generalized Euler-Frobenius* polynomial Π_ϕ introduced by Chui and Wang [10]. To define Π_ϕ, we need the correllation sequence

$$\gamma_\phi(n) = \int_{-\infty}^{\infty} \phi(n+x)\overline{\phi(x)}dx$$

which clearly satisfies $\gamma_\phi(-n) = \overline{\gamma_\phi(n)}$ and supp $\gamma_\phi \subseteq [-N_\phi, N_\phi]$. Note that $\gamma_\phi(n) = \delta_{n,0}$ if ϕ is orthonormal ; otherwise, there is a positive integer k_ϕ such that supp $\gamma_\phi = [-k_\phi, k_\phi]$. Then the generalized Euler-Frobenius polynomial is defined by

$$\Pi_\phi(z) = \sum_{n=0}^{2k_\phi} \gamma_\phi(n-k_\phi)z^n.$$

Let us again consider the two typical examples:

1. For $\phi = N_m$, we have $k_\phi = m-1$, and Π_ϕ becomes

$$\Pi_\phi(z) = \sum_{n=0}^{2m-2} N_{2m}(n+1)z^n$$

   which is a $1/(2m-1)!$ multiple of the classical Euler-Frobenious polynomial of degree $2m-2$ (cf. Schoenberg [32]); and

2. for any orthonormal ϕ, since $k_\phi = 0$, $\Pi_\phi(z)$ is the constant 1.

Corresponding to an arbitrary ϕ, there is still a lot of freedom for choosing a desirable wavelet. It was shown by Chui and Wang [10], however, that, with the exception of an arbitrary shift, there is a unique wavelet ψ with minimum support, and the two-scale sequence $\{q_n\}$ for this wavelet has the symbol (defined by Equation 2.27 with a factor of 1/2), given by

$$Q(z)\mu_\phi(-z), \tag{2.35}$$

where $\mu_\phi(-z)$ is the highest-degree polynomial factor of $z^{N_\phi - k_\phi - 1}\Pi_\phi(z)\check{P}(z)$ without any symmetric zeros. Here, $\check{P}$ denotes the reciprocal polynomial of P and z_0 is called a symmetric zero of a polynomial f if $f(z_0) = f(-z_0) = 0$ for $z_0 \neq 0$, and $f(0) = f'(0) = 0$ for $z_0 = 0$. Note that the wavelet ψ has

compact support and the size of its support is dictated by the degree of the polynomial Q.

For example if $\phi = N_m$, since the two-scale symbol is given by

$$P(z) = 1/2 \sum_{n=0}^{m} 2^{-m+1} \begin{pmatrix} m \\ n \end{pmatrix} z^n, \tag{2.36}$$

and since the classical Euler-Frobenius polynomial does not vanish at 1 and has no symmetric roots, then we have

$$Q(z) = \Pi_\phi(-z)\check{P}(-z) = \left(\frac{1-z}{2}\right)^m \Pi_\phi(-z). \tag{2.37}$$

Once the polynomials P and Q are determined, the two decomposition sequences $\{a_n\}$ and $\{b_n\}$ can be computed by using Equation 2.31 . They are given by

$$\begin{cases} G(z) = z^{-N_\phi + k_\phi} \frac{\Pi_\phi(z)\check{P}(z)}{\Pi_\phi(z^2)}; \\ H(z) = -z^{-N_\phi + k_\phi} \frac{\Pi_\phi(z)\check{P}(z)P(-z)}{\mu(z)\Pi_\phi(z^2)}. \end{cases} \tag{2.38}$$

Here, we remark that the generalized Euler-Frobenius polynomial Π_ϕ never vanishes on the unit circle (cf. Chui and Wang [10]). Hence, both the decomposition sequences $\{a_n\}$, $\{b_n\}$ have exponential decay, and are actually finite sequences, if and only if ϕ and ψ are both orthonormal. So, although ϕ and ψ both have compact supports and their two-scale (or reconstruction) sequences are finite, their decomposition sequences are necessarily infinite unless they are both orthonormal.

To obtain finite decomposition sequences, we may use their duals $\tilde{\phi} \in V_0$ and $\tilde{\psi} \in W_0$ defined (uniquely) by

$$\begin{cases} \langle \phi(\cdot - m), \tilde{\phi}(\cdot - n) \rangle = \delta_{m,n}; \\ \langle \psi(\cdot - m), \tilde{\psi}(\cdot - n) \rangle = \delta_{m,n}, \end{cases} \tag{2.39}$$

for all $m, n \in \mathcal{Z}$. The two-scale relations of $\tilde{\phi}$ and $\tilde{\psi}$ are given by

$$\begin{cases} \hat{\tilde{\phi}}(\omega) = \overline{G(e^{-i\omega/2})}\hat{\tilde{\phi}}(\omega/2); \\ \hat{\tilde{\psi}}(\omega) = \overline{H(e^{-i\omega/2})}\hat{\tilde{\phi}}(\omega/2); \end{cases} \tag{2.40}$$

Hence, the decomposition formulas for $\tilde{\phi}$ and $\tilde{\psi}$ are determined by the finite sequences $\{p_n\}$ and $\{q_n\}$, namely:

$$\tilde{\phi}(2t-l) = 1/2 \sum_n \tilde{p}_{l-2n}\tilde{\phi}(t-n) \ + 1/2 \sum_n \tilde{q}_{l-2n}\tilde{\psi}(t-n) \tag{2.41}$$

for all $l \in \mathcal{Z}$. For more details, the reader is referred to Chui and Wang [11], [10] .

2.4 Pyramid Algorithms and Linear Phase Filters

We have already seen the relationship between the two pairs of sequences ($\{p_n\}$, $\{q_n\}$) and ($\{a_n\}$, $\{b_n\}$), and how they relate ϕ and ψ in Equations 2.24 - 2.26. Of course, for orthonormal ϕ and ψ, they are simply related by Equation 2.34. Now, with the notation $\phi_{k,j}$ defined in Equation 2.10 and the analogous notation for $\psi_{k,j}$, we see that $\{\phi_{k,j} : j \in \mathcal{Z}\}$ and $\{\psi_{k,j} : j \in \mathcal{Z}\}$ are unconditional bases of V_k and W_k, respectively; and furthermore, Equations 2.24 - 2.26 become:

$$\begin{cases} \phi_{k,l} = \sum_n p_{n-2l}\phi_{k+1,n} \\ \psi_{k,l} = \sum_n q_{n-2l}\phi_{k+1,n} \end{cases} \tag{2.42}$$

and

$$\phi_{k,l} = \sum_n a_{l-2n}\phi_{k,n} \ + \sum_n b_{l-2n}\psi_{k,n} \tag{2.43}$$

for all $k, l \in \mathcal{Z}$.

Now, for any $f \in \mathcal{L}^2$, in view of the first condition (i) on the multiresolution analysis, there exist sufficiently large values of N, such that $\|f - f_N\|_{\mathcal{L}^2}$ is arbitrarily small, where $f_N \in V_N$. Here, if some knowledge of ϕ is known (such as the B-spline N_m), we can even determine f_N by interpolation or quasi-interpolation using discrete data information of F (see Chui [12] and the references therein). Since the space V_N has the orthogonal decomposition:

$$\begin{aligned} V_N &= V_{N-1} \bigoplus W_{N-1} \\ &= W_{N-1} \bigoplus V_{N-2} \bigoplus W_{N-2} \end{aligned}$$

$$\begin{aligned} &= \cdots \\ &= \cdots \\ V_N &= W_{N-1} \bigoplus \cdots \bigoplus W_{N-M} \bigoplus V_{N-M} \end{aligned}$$

for an arbitrarily positive integer M, the approximant $f_N \in V_N$ has the (unique) orthogonal decomposition:

$$f_N = \sum_{j=N-M}^{N-1} g_j + f_{N-M}, \quad g_j \in W_j, \ f_{N-M} \in V_{N-M}. \tag{2.44}$$

This is called a *wavelet decomposition* of f_N as opposed to the notion of complete wavelet decomposition in Equation 2.15. Set

$$\begin{cases} f_k = \sum_j c_j^k \phi_{k,j} \\ g_k = \sum_j d_j^k \psi_{k,j}. \end{cases} \tag{2.45}$$

Then by using l^2-linear independence and the orthogonality between $\phi_{k,j}$ and $\psi_{k,l}$, we see that the relation $f_{k+1} = f_k + g_k$ is equivalent to

$$\begin{cases} c_j^k = \sum_n a_{n-2j} c_n^{k+1}, \ j \in \mathcal{Z}; \\ d_j^k = \sum_n b_{n-2j} c_n^{k+1}, \ j \in \mathcal{Z}, \end{cases} \tag{2.46}$$

by applying Equation 2.43, and to

$$c_j^{k+1} = \sum_n p_{j-2n} c_n^k \ + \sum_n a_{j-2n} d_n^k, \quad j \in \mathcal{Z}, \tag{2.47}$$

by applying Equation 2.42. By introducing the notations $c^k = \{c_j^k\}$ and $d^k = \{d_j^k\}$, we can diagram the pyramid algorithms for wavelet decomposition (using Equation 2.46), and wavelet reconstruction (using Equation 2.47):

$$\begin{array}{ccccccccc} & & d^{N-1} & & d^{N-2} & & & & d^{N-M} \\ & \nearrow & & \nearrow & & \nearrow & & \nearrow & \\ c^N & \rightarrow & c^{N-1} & \rightarrow & c^{N-2} & \rightarrow & \cdots & \rightarrow & c^{N-M} \end{array}$$

$$\begin{array}{ccccccccc} d^{N-M} & & d^{N-M+1} & & & & d^{N-1} & & \\ & \searrow & & \searrow & & & & \searrow & \\ c^{N-M} & \rightarrow & c^{N-M+1} & \rightarrow & \cdots & & c^{N-1} & \rightarrow & c^N \end{array}$$

For more details, see Mallet [25], Daubechies [15], and Chui and Wang [9].

The decomposition of the approximant f_N into the wavelet components $g_k,\ k = N-M, ..., N-1$, and the blurred component f_{N-M} (where $\|f_{N-M}\|_{\mathcal{L}^2} \to 0$ as $M \to +\infty$), separates and localizes the frequencies of the signal f_N by means of the IWT at dyadic time-intervals, and this information is reflected by the wavelet coefficients d_j^k. Recall from Equation 2.20 that, indeed, d_j^k is precisely a $2^{\frac{k}{2}}$ multiple of the IWT of f_N at the time-scale location $(j/2^k, 2^{-k})$, using the wavelet window function $\tilde{\psi}$, the dual of ψ. (Here, scale is inversely proportional to frequency; and we have interchanged ψ with its dual $\tilde{\psi}$ in Equation 2.20). Hence, we have a perfectly nice filtering tool to reduce, enhance, and even eliminate unwanted frequencies at certain time locations. The filtered signal can later be reconstructed by applying the above wavelet reconstruction algorithm. In many filtering applications, in order to avoid distortion, the filter must have at least approximate linear phase. Let us be more precise on the notion of generalized linear phase and linear phase as follows.

A real-valued function f in $\mathcal{L}^1$ is said to have generalized linear phase if its Fourier transform has the representation:

$$\hat{f}(\omega) = A(\omega)e^{i(a\omega+b)}, \tag{2.48}$$

where $A(w)$ is real-valued, and a and b are real constants. It is said to have linear phase if, in addition,

$$A(\omega) = |\hat{f}(\omega)| \quad \text{and} \quad e^{ib} = \pm 1. \tag{2.49}$$

Similarly, a real-valued l^1-sequence f_n is said to have generalized linear phase if its discrete Fourier transform (or Fourier series) $F(w)$ has the formulation (2.48) (with $\hat{f}(w)$ replaced by $F(w)$); and is said to have linear phase if $A(w) = \mid F(w) \mid$ and $e^{ib} = \pm 1$. It is well known that f has generalized linear phase if and only if it is symmetric or antisymmetric. Hence, by a result of Daubechies [15], any non-trivial orthonormal ${}_N\phi$ or wavelet ${}_N\psi$ does not have generalized linear phase. It was shown in Chui and Wang [10], however, that if the two-scale sequence $\{p_n\}$ is real and has generalized linear phase, then both ϕ and ψ have generalized linear phase; and if this real sequence p_n has linear phase, so do ϕ and ψ. Typical examples include B-splines N_m

and their corresponding B-*wavelets* ψ_m, defined by Equation 2.23, as well as their duals, all have generalized linear phases for all m, and they have linear phases if and only if m is even.

We have seen in this chapter that one of the basic methods for constructing univariate wavelets involves the use of *B-spline functions*. Since we are interested in constructing multivariate wavelets, we choose *Box splines* which are natural generalization of B-splines. The box splines are probably the simplest multivariate spline functions with small supports that are efficient for both software and hardware implementations. The next chapter is devoted to a study of basic theory of box splines with emphasis on the properties that are crucial to construction of wavelets.

Chapter 3

Box Splines

This chapter is intended to cover some basic theory of bivariate box splines and the extension of this theory to the multivariate setting. Since we will restrict our attention to the essential topics that are somewhat related to the material discussed in the next chapters, much of the extensive theory of multivariate splines is not going to be covered. We refer the reader to the text by M.Daehlen and T.Lyche [14] for further study of the box spline theory. A good understanding of univariate spline theory is obviously important for learning this rapidly developing subject of multivariate box splines. We encourage the reader to refer the text by R.H Bartels [1] to get a good understanding of univariate spline theory.

3.1 Bivariate Box Splines

A natural generalization of the univariate B-spline on a uniform mesh to the multivariate setting is the so-called *box spline*. We can define general bivariate box spline as follows:

Definition 3.1 *Suppose* $\Xi_\mu = (\xi^1, \xi^2, \ldots, \xi^\mu)$ *with* $\xi^i = (x_i, y_i)$ *are* $\mu \geq 2$ *vectors in* $\mathcal{R}^2$ *which span* $\mathcal{R}^2$*. The ordering of these vectors is unimportant in the construction of box splines, so assume, without loss of generality, that* ξ^1 *and* ξ^2 *are linearly independent. A bivariate box spline with direction vectors* Ξ_μ*, is the function* $N(u, v|X_\mu) : \mathcal{R}^2 \to \mathcal{R}$ *defined recursively by*

$$N(u, v|\Xi_\mu) = \int_0^1 N(u - tx_\mu, v - ty_\mu|\Xi_{\mu-1})dt, \tag{3.1}$$

with

$$N(u,v|\Xi_2) = \begin{cases} 1/|det(\Xi_2)|, & \text{if } (u,v) \in [\Xi_2[, \\ 0, & \text{otherwise,} \end{cases} \tag{3.2}$$

and where

$$[\Xi_\mu[= \left\{ t_1\xi^1 + \ldots + t_\mu\xi^\mu : 0 \le t_j < 1,\ 1 \le j \le \mu \right\}. \tag{3.3}$$

Note the orientation of square brackets in $[\Xi_\mu[$. This orientation is used to denote which ends of the 0, 1 interval are open or closed.

Example 3.1 *The function $N(u,v|\Xi_2)$ is a constant C times the indicator function $\chi_{[\Xi_2[}$ on the half open parallelogram $[\Xi_2[$ with sides parallel to ξ^1 and ξ^2, where C is the reciprocal of the area of the parallelogram and $\chi_{[\Xi_2[}$ is defined as in Equation 3.6. If for positive integers k_1 and k_2 we set $\mu_1 = k_1 + k_2$ and Ξ_μ consists of the unit vectors e^1 and e^2 repeated k_1 and k_2 times, we obtain the tensor product B-splines. In this case $[\Xi_\mu[$ is a semi-open rectangle with sides k_1 and k_2.*

In Theorem 3.2 below we note that $N(u,v|\Xi_\mu) = 0$ if (u,v) does not belong to $[\Xi_\mu[$.

Definition 3.2 *The set*

$$[\Xi_\mu] = \left\{ t_1\xi^1 + \ldots + t_\mu\xi^\mu : 0 \le t_j \le 1,\ 1 \le j \le \mu \right\}.$$

is called the support of $N(u,v|\Xi_\mu)$.

Example 3.2 *The Courant finite element $N(u,v) = N(u,v|\Xi_3)$ with $\Xi_3 = (\xi^1, \xi^2, \xi^3)$ where $\xi^1 = (1,0)^T$, $\xi^2 = (0,1)^T$, and $\xi^3 = (1,1)^T$. From Equation 3.1 we have*

$$N(u,v) = \int_0^1 N(u-t, v-t| \left((1,0)^T, (0,1)^T\right))dt. \tag{3.4}$$

By Equation 3.2 the integrand is the indicator function of the half open unit square. Therefore, $N(u,v)$ can only be nonzero if (u,v) belongs to the

interior of the hexagonal support set $[\Xi_3]$ *looking as in Figure 3.1 . After some calculation we find*

$$N(u,v) = u\chi_A + v\chi_B + (u+1-v)\chi_C + (v+1-u)\chi_D + (2-v)\chi_E + (2-u)\chi_F. \tag{3.5}$$

where for any region $G \subset \mathcal{R}^2$ *we have the indicator function*

$$\chi_G(u,v) = \begin{cases} 1, & (u,v) \in G \\ 0, & otherwise \end{cases} \tag{3.6}$$

The graph of N is shaped like a hexagonal pyramid and is known as the Courant finite element .

We will now take a closer look at the situation where the direction vectors of a box spline consists of repetitions of a few nonparallel directions.

Definition 3.3 *Suppose for an integer* $r \geq 2$, *that*

$$\Xi_\mu = (\overbrace{\xi^1, \ldots, \xi^1}^{k_1 times}, \overbrace{\xi^2, \ldots, \xi^2}^{k_2 times}, \ldots, \overbrace{\xi^r, \ldots, \xi^r}^{k_r times}), \tag{3.7}$$

where $E = (\xi^1, \xi^2, \ldots, \xi^r)$ *are pairwise nonparallel vectors in* $\mathcal{R}^2$, *and* $k_1, k_2, \ldots, k_r$ *are positive integers with* $\mu = \sum_i k_i$. *We write*

$$\Xi_\mu = E^k = E^{(k_1,\ldots,k_r)}$$

and call $N(u,v|E^k)$ *an r directional box spline. If for* $r \leq 4$ *we have the standard directions*

$$\begin{cases} \xi^1 = d^1 = (1,0)^T = \rightarrow, \\ \xi^2 = d^2 = (0,1)^T = \uparrow, \\ \xi^3 = d^3 = (1,1)^T = \nearrow, \\ \xi^4 = d^4 = (1,-1)^T = \searrow, \end{cases} \tag{3.8}$$

then we denote $= N(u,v|E^k)$ *by* $N_k(u,v)$. *We define the value of* N_k *or one of its derivatives on a grid line by taking limits from the right and/or above.*

We see that the Courant finite element $N_{(1,1,1)}$ is a three directional box spline. The standard directions generate regular r directional grids $G_r, r = 2, 3, 4$. To obtain G_r we draw straight lines through each integer grid point in the r first standard directions. These grids are shown in Figure 3.2. In the literature the regular three and four directional grids are sometimes called type-1 and type-2 triangulations. The terms unidiagonal and crisscross partitions are also used by Chui [12].

When $r = 3$ the support of the box spline $N_k = N_{(k_1,k_2,k_3)}$ is

$$[E^{k_1,k_2,k_3}] = \{(t_1 + t_3, t_2 + t_3) : 0 \leq t_i \leq k_i, \quad 1 \leq i \leq 3\}$$

A sequence of such hexagonal sets and corresponding grids are shown from left to right in the second and third rows of Figure 3.3. At the left of second row in Figure 3.3 we find the unit square which is the support of $N_{(1,1)}$. To the right of the unit square we see the support of $N_{(1,1,1)}$. Continuing towards the right we see support of $N_{(1,1,2)}$. The supports of $N_{(1,2,2)}$ and $N_{(2,2,2)}$ are shown from left to right in the third row of Figure 3.3.

The proofs for all the theorems given in this section and the next section are found in Daehlen and Lyche [14].

Theorem 3.1 *N_k is a piecewise polynomial of degree at most $\mu - 2 = \sum_{i=1}^{r} k_i - 2$ on $G_r, r = 2, 3, 4$. Moreover, $N_k \in C^{\mu-2-k_i}$ across a direction d_i.*

In order to construct a C^0 quadratic box spline we have $\mu - 2 = 2, k_1 = k_2 = 1, k_3 = 2$. Therefore, according to Theorem 3.1 we have C^1 continuity across the u and v axes, and C^0 continuity across diagonals as asserted.

Other three directional box splines of practical interest are the three C^1 cubics with direction multiplities (2,2,1), (2,1,2), and the C^2 quartic (2, 2, 2). The support of the functions $N_{(1,2,2)}$ and $N_{(2,2,2)}$ are shown in the third row of Figure 3.3. By using four directions we can obtain more smoothness with lower degree. The function $N_{(1,1,1,1)}(u, v)$ is called the Zwartz-Powell Element. It is a piecewise quadratic of smoothness C^1. The support of this function is shown in the top left position in Figure 3.3. Definition 3.1 gives rise to the construction of box splines defined on a variety of different grids. So far in this section we have considered the important standard grids given by the standard directions d^1, d^2, d^3, and d^4 (Equation 3.8). The grids G_r,

$r = 2, 3, 4$ shown in Figure 3.2 are the most well known because of their regularity and behavior. However, by introducing the four extra directions

$$d^5 = \begin{pmatrix} 1/2 \\ 1 \end{pmatrix}, d^6 = \begin{pmatrix} 1 \\ 1/2 \end{pmatrix}, d^7 = \begin{pmatrix} 1 \\ -1/2 \end{pmatrix}, d^8 = \begin{pmatrix} 1/2 \\ -1 \end{pmatrix}$$

we can build a group of fairly well behaved grids. Figure 3.4(a)-(d) show four grids that are constructed by selecting vectors among $(d^1, \ldots d^8)$,

$$\begin{array}{ll} d^1,\ d^2,\ d^3,\ d^4,\ d^5, & Figure\ 3.4(a), \\ d^1,\ d^2,\ d^4,\ d^5,\ d^6, & Figure\ 3.4(b), \\ d^1,\ d^2,\ d^5,\ d^6,\ d^7,\ d^8 & Figure\ 3.4(c), \\ d^1,\ d^2,\ d^3,\ d^4,\ d^5,\ d^6,\ d^7,\ d^8, & Figure\ 3.4(d). \end{array}$$

We now list some useful properties of box splines:

Theorem 3.2 *For $N(u, v) = N(u, v|\Xi_\mu)$ we have*
(i) $N(u, v) \equiv 0$ for $(u, v) \notin [\Xi_\mu[$. (Local support)
(ii) $N(u, v) > 0$ for $(u, v) \in [\Xi_\mu[$. (Positivity)
(iii) $\int\int_{[\Xi_\mu]} N(u, v)\, du\, dv = 1$. (Normalization)
Here

$$]\Xi_\mu[= \left\{ t_1 x^1 + \ldots + t_\mu x^\mu : 0 < t_j < 1, 1 \le j \le \mu \right\}. \tag{3.9}$$

The next result shows that under the mild restrictions on the direction vectors the translates of box splines form a partition of unity.

Theorem 3.3 *Suppose that ξ^i and ξ^2 in addition to being linearly independent have integer components. Then for each $(u, v) \in \mathcal{R}^2$ we have*

$$(iv) \sum_{(i,j)\in\mathcal{Z}^2} N(u - i, v - j|\Xi_\mu) \equiv 1, \quad (Partition\ of\ Unity)$$

We next turn to differentiation of box splines.

Definition 3.4 *For a sufficiently smooth function f we let*

$$D_x f(u, v) = \lim_{h\to 0,\ h>0} \frac{f(u + hx, v + hy) - f(u, v)}{h} \tag{3.10}$$

denote the one sided derivative in the direction $\xi = (x, y)$.

If $f \in C^1$ then $D_x f = x\partial f/\partial u + y\partial f/\partial v$ is the usual directional derivative of f. To state a differentiation formula we need the reduced direction vectors.

$$\Xi_\mu^i = \Xi_\mu \setminus \left\{\xi^i\right\} = (\xi^1, \ldots, \xi^{i-1}, \xi^{i+1}, \ldots, \xi^\mu)$$

If ξ^1 and ξ^2 are linearly independent then $N(u, v|\Xi_\mu^i)$ is well defined for $i = 3, \ldots, \mu$. However, it is possible that $N(u, v|\Xi_\mu^1)$ and $N(u, v|\Xi_\mu^2)$ are not well defined. For example, if $\Xi_3 = ((1,0)^T, (0,1)^T, (0,1)^T)$ then $\Xi_3^1 = ((0,1)^T, (0,1)^T))$ and these vectors are not linearly independent.

Theorem 3.4 *If Ξ_μ^i contains at least two linearly independent vectors then*

$$D_{\xi^i} N(u, v|\Xi_\mu) = N(u, v|\Xi_\mu^i) - N(u - x_i, v - y_i|\Xi_\mu^i). \tag{3.11}$$

If $\xi = \sum_{i=1}^{\mu} \omega_i \xi^i$ then

$$D_\xi N(u, v|\Xi_\mu) = \sum_{i=1}^{\mu} \omega_i [N(u, v|\Xi_\mu^i) - N(u - x_i, v - y_i|\Xi_\mu^i)], \tag{3.12}$$

provided $N(u, v|\Xi_\mu^i)$ is well defined for all i with $\omega_i \neq 0$.

3.2 Multivariate Box Splines

We will now give a number of useful results on multivariate box splines. We start by stating the general definition.

Definition 3.5 *Suppose for positive integers μ and d that $\Xi_\mu = (\xi^1, \xi^2, \ldots, \xi^\mu)$ where $\xi^i = (x_1^i, x_2^i, \ldots, x_d^i)$ are $\mu \geq d$ vectors in R^d with $det(\Xi_d) \neq 0$. A d-variate box spline with direction vectors Ξ_μ, is a function $N(u \mid \Xi_\mu) : \mathcal{R}^d \to \mathcal{R}$ defined recursively by*

$$N(u|\Xi_\mu) = \int_0^1 N(u - t\xi^\mu|\Xi_{\mu-1})dt, \tag{3.13}$$

$$N(u|\Xi_d) = \begin{cases} 1/|det(\Xi_d)|, & \text{if } u \in [\Xi_d[, \\ 0, & \text{otherwise} \end{cases} \tag{3.14}$$

and where

$$[\Xi_\mu[= \left\{t_1\xi^1 + \ldots t_\mu\xi^\mu \ : \ 0 \leq t_j < 1, 1 \leq j \leq \mu\right\}. \tag{3.15}$$

By induction we see that $N(u \mid \Xi_\mu)$ is positive on

$$[\Xi_\mu] = \left\{ t_1\xi^1 + \ldots t_\mu\xi^\mu : 0 < t_j < 1, 1 \leq j \leq \mu \right\}$$

and zero for any u not in the set

$$[\Xi_\mu] \left\{ t_1\xi^1 + \ldots t_\mu\xi^\mu : 0 \leq t_j \leq 1, \quad 1 \leq j \leq \mu \right\}$$

The set $[\Xi_\mu]$ is called the *support* of $N(u \mid \Xi_\mu)$.

de Boor and Hollig proved that box splines are piecewise polynomials of degree $\mu - d$ in [17]. Moreover, $N(u \mid \Xi_\mu) \in C^{\xi-1}(\mathcal{R}^d)$ where

$$\xi = max\{r : \text{ all selections Y of r elements from } \Xi_\mu \text{ are such that } \Xi_\mu \backslash Y \text{ spans } \mathcal{R}^d \}.$$

The proof of these properties are quite complicated. For proofs refer [14].

It is useful to have criteria for conditions under which translates of box splines are linearly independent. The following theorem states the necessary and sufficient conditions to ensure global linear independence.

Theorem 3.5 *The set*

$$\left\{ N(u - j \mid \Xi_\mu), \quad j \in \mathcal{Z}^d \right\}$$

is linearly independent on $\mathcal{R}^d$ if and only if all selections Y of d vectors from Ξ_μ are such that det Y takes one of the three values -1, 0, or 1.

It is also useful to have criteria for local linear independence of box splines.

Theorem 3.6 *Suppose Ω is any region on which all translates of a box spline are polynomials. Then the set of box splines that are nonzero on Ω is linearly independent on Ω if and only if the condition in Theorem 3.5 holds.*

The formula given in the following theorem is basic and it is often used as the definition of box splines. It defines a box spline as a distribution, or generalized functions.

Theorem 3.7 *Suppose* $det(\Xi_d) \neq 0$. *For any function* f *that is continuous on* $[\Xi_\mu]$

$$\int_{[\Xi_\mu]} N(u|\Xi_\mu) f(u) du = \int_{[0,1]^\mu} f(\Xi_\mu t) dt \tag{3.16}$$

where $t = (t_1, \ldots, t_\mu)$, $[0,1]^\mu$ *is the unit cube in* $\mathcal{R}^\mu$, *and*

$$\Xi_\mu t = \sum_{i=1}^{\mu} t_i \xi^i$$

Corollary 1 *We have*

$$N(u|\Xi_\mu) = N(u|Y_\mu)$$

for any permutation $Y_\mu = (y^1, y^2, \ldots, y^\mu)$ *of* Ξ_μ.

The Fourier transform of a box spline is given next.

Corollary 2 *Suppose* Ξ_μ *contains a basis for* $\mathcal{R}^d$. *Then*

$$\hat{N}(z|\Xi_\mu) = \int_{\mathcal{R}^d} e^{i(u\cdot z)} N(u|\Xi_\mu) du = \Pi_{j=1}^{\mu} \frac{e^{iz\cdot x^j} - 1}{iz \cdot x^j}$$

where $x \cdot y = \sum_{j=1}^{d} x_i y_i$ *for any vectors* $x = (x_1, \ldots, x_\mu)$ *and* $y = (y_1, \ldots, y_\mu)$ *in* $\mathcal{R}^d$.

Theorem 3.7 can also be used to give a geometric interpretation of a box spline.

Corollary 3 *Let* $Y_\mu = (y^1, y^2, \ldots, y^\mu)$ *with* $y^i = (x^i, z^i) \in \mathcal{R}^\mu$ *and* $z^i \in \mathcal{R}^{\mu-d}$, *be a lifting of* Ξ_μ *from* $\mathcal{R}^d$ *to* $\mathcal{R}^\mu$ *such that* $det(Y_\mu) \neq 0$. *Then*

$$N(u \mid \Xi_\mu) = \frac{1}{det(Y_\mu)} \int_{\mathcal{R}^{\mu-d}} \chi[Y_\mu](u, w) dw, \ \ for\ u \in \mathcal{R}^d, \tag{3.17}$$

where $\chi_{[Y_\mu]}$ *is the indicator function of the parallelepiped*

$$[Y_\mu] = \left\{ t_1 y^1 + t_2 y^2 + \ldots + t_\mu y^\mu : 0 \leq t_i \leq 1, i = 1, 2, \ldots, n \right\}.$$

Geometrically we can construct $N(u|\Xi_\mu)$ as follows. We first lift the vectors $x^1, x^2, \ldots, x^\mu$ in $\mathcal{R}^d$ to vectors $Y_\mu = (y^1, y^2, \ldots, y^\mu)$ in $\mathcal{R}^\mu$. A possible lifting is given by $y^i = (x^i, 0),\ i = 1, 2, \ldots, d$ and $y^i = (x^i, e^{i-d}),\ i = d+1, \ldots, n$, where the e^{i-d} are the unit vectors in $\mathcal{R}^{\mu-d}$. Now for each $u \in \mathcal{R}^d$ we obtain the value $N(u|\Xi_\mu)$ as the $\mu - d$ dimensional volume of those points in $[Y_\mu]$ that project to u. In symbols

$$N(u|\Xi_\mu) = \frac{vol_{n-d}(\{w \in \mathcal{R}^{\mu-d} : (u, w) \in [Y_\mu]\})}{vol_\mu([Y_\mu])},\ u \in \mathcal{R}^d, \tag{3.18}$$

where $vol_\mu([Y_\mu]) = det([Y_\mu]) \neq 0$.

Example 3.3 *For the $N_{(1,1,1)}$ bivariate box spline and the lifting $Y_3 = (1,0,0)^T, (1,1,1)^T$ Equation 3.17 takes the form*

$$N_{(1,1,1)}(u, v) = \frac{1}{|det(Y_3)|} \int_{\mathcal{R}} \chi[Y_3](u, v, w)dw, \quad for\ (u, w) \in \mathcal{R}^2.$$

$[Y_3]$ is a parallelepiped obtained form the unit cube in $\mathcal{R}^3$ by moving the top facet horizontally in the (1,1) direction. At (u, v) the value of the box spline is given by the length of that part of the vertical line through $(u, v, 0)$ that lies in $[Y_3]$.

Corollary 2 can be used to derive knot line refinement algorithms for box splines. The following is a result of Cavaretta and Micchelli [4].

Theorem 3.8 *For $\nu \in \mathcal{N}(\equiv$ set of positive integers) and $\xi^l \in \mathcal{Z}^d, l = 1, 2, \ldots, \mu$*

$$N(u|\Xi_\mu) = \sum_{j \in \mathcal{Z}^d} \beta(j|\Xi_\mu) N(\nu u - j|\Xi_\mu), u \in \mathcal{R}^d, \tag{3.19}$$

where the generating function for the $\beta(j|\Xi_\mu)$ is

$$q(z|\Xi_\mu) = \sum_j \beta(j|\Xi_\mu) z^j = \nu^{d-\mu} \Pi_{l=1}^{\mu} (1 + z^{\xi^l} + z^{2\xi^l} + \ldots + z^{(\nu-1)\xi^l}), \tag{3.20}$$

$z = (z_1, z_2, \cdots, z_d)$ and $z^y = {z_1}^{y_1} \cdot {z_2}^{y_2} \cdots {z_d}^{y_d}$ for any $y = (y_1, \cdots, y_d)$

We can also give a recurrence relation for the β's. We define $\beta(j|\Xi_\mu)$ by Equation 3.20 also for $\mu = 1, 2, \ldots, d-1$.

Theorem 3.9 *For $\mu > 1$ and $x^l \in \mathcal{Z}^d, l = 1, 2, \ldots, \mu$ we have*

$$\beta(j|\nu, \Xi_\mu) = \frac{1}{\nu} \sum_{l=0}^{\nu-1} \beta(j - lx^\mu|\nu, \Xi_{\mu-1}). \tag{3.21}$$

The following transformation formula for box splines is sometimes useful. For a $d \times d$ matrix $\mathbf{A}$ we let $A\Xi_\mu$ denote the direction vectors $A\xi^1, A\xi^2, \ldots, A\xi^\mu$.

Theorem 3.10 *Suppose $\boldsymbol{A}$ is a nonsingular $d \times d$ matrix. Then* $N(u|\Xi_\mu) = |detA|N(Au|A\Xi_\mu)$ *(Transformation Formula)*

Next we state a symmetry property of box splines. We need the left continuous versions $\tilde{N}$ of box splines. To define $\tilde{N}$ we use Equation 3.13 and replace $[\Xi_d[$ by $]\Xi_d] = \left\{t_1\xi^1 + \ldots + t_d\xi^d : \ \ 0 < t_i \leq 1, i = 1, 2, \ldots, d\right\}$ in Equation 3.14.

Theorem 3.11 *For any Ξ_μ we have*

$$N(u|\Xi_\mu) = \tilde{N}(\sum_{i=1}^{\mu} \xi^i - u|\Xi_\mu). \tag{3.22}$$

It is also of interest to see what happens if we change sign, or flip one of the direction vectors.

Theorem 3.12 *For $u \in \mathcal{R}^d$ and $i = 1, \ldots, n$ we have*

$$N(u|x^1, \ldots, x^\mu) = N(u - x^i|x^1, \ldots, x^{i-1}, -x^d, x^{i+1}, \ldots, x^\mu). \tag{3.23}$$

Next we will give a formula for the easy calculation of the inner product of two translated box splines.

Theorem 3.13

$$N(u + 2c_{\mu_2}|\Xi_{\mu_1} \bigcup \Xi_{\mu_2}) = \int N(v|\Xi_{\mu_1}) \cdot N(v - u|\Xi_{\mu_2})dv \tag{3.24}$$

Where c_{μ_2} is the center of $N(u|\Xi_{\mu_2})$

Proof of this theorem is found in Diamond, Raphel and Williams [18]. In particular, if u is an integer then inner product of $\int N(v|\Xi_{\mu_1}) \cdot N(v-u|\Xi_{\mu_2})dv$ is simply the value of the box spline $N(\cdot|\Xi_{\mu_1} \bigcup \Xi_{\mu_2})$ at an integer. These values can be easily calculated. See for instance, Chui and Diamond [7].

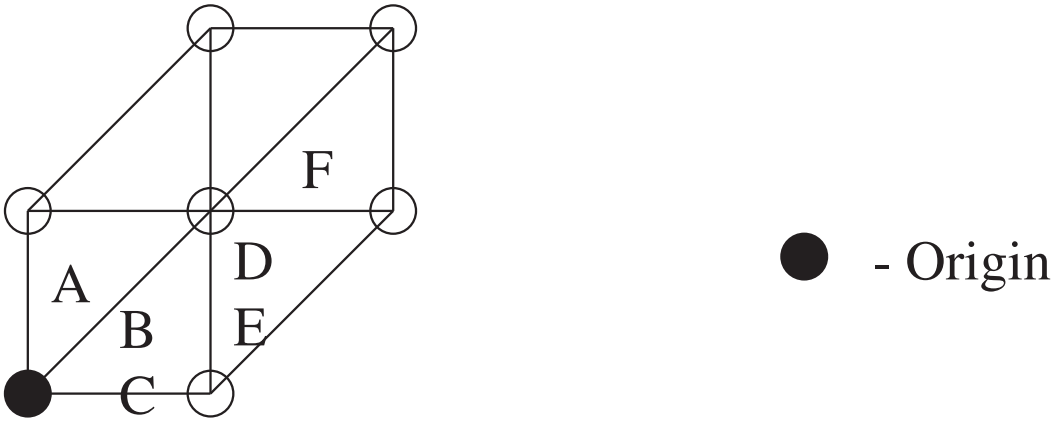


Figure 3.1: The support of the Courant finite element

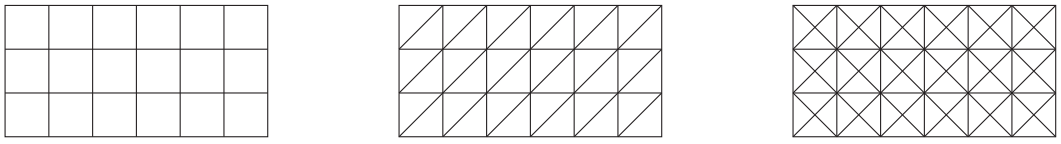


Figure 3.2: Two, three and four directional grids

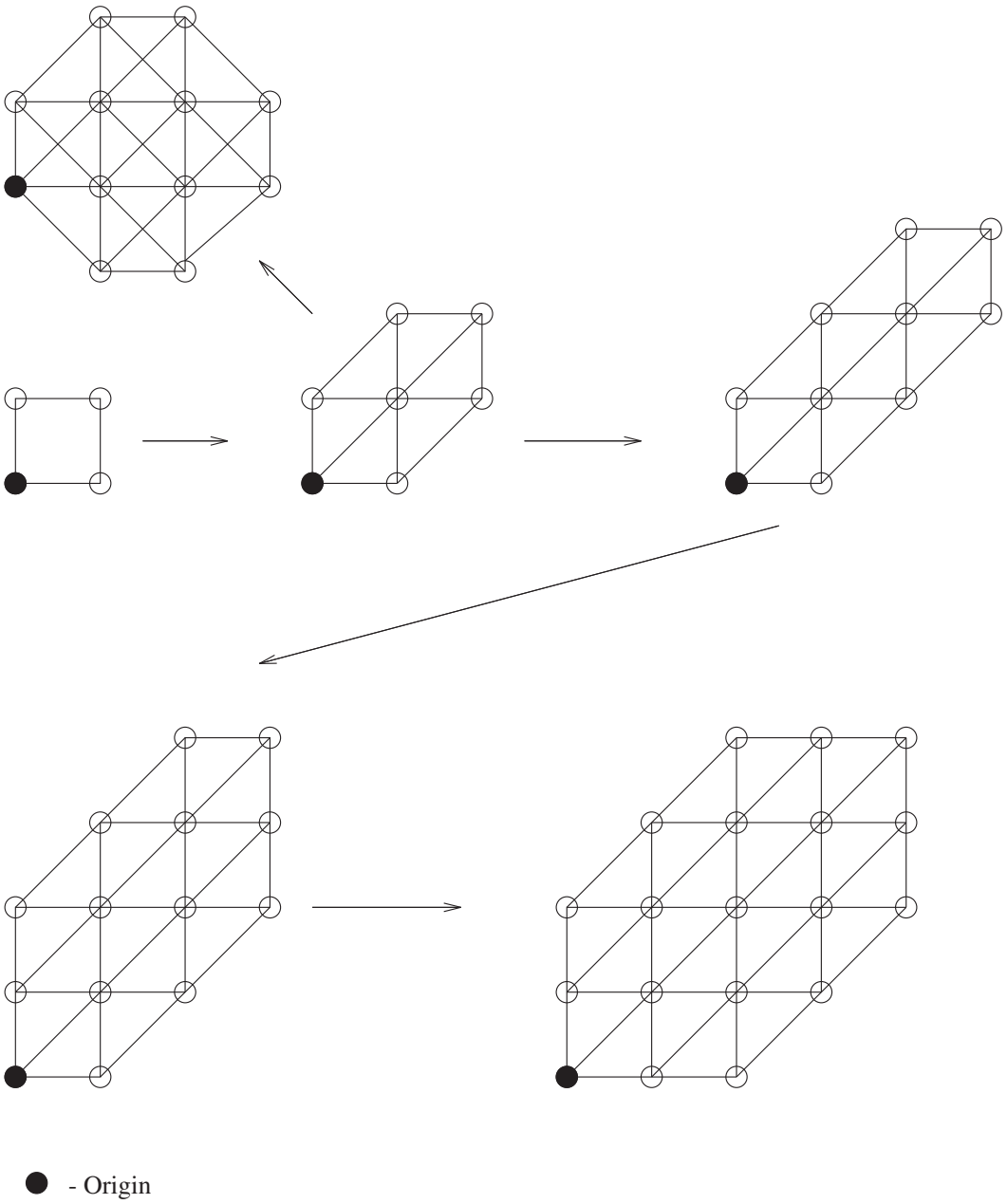
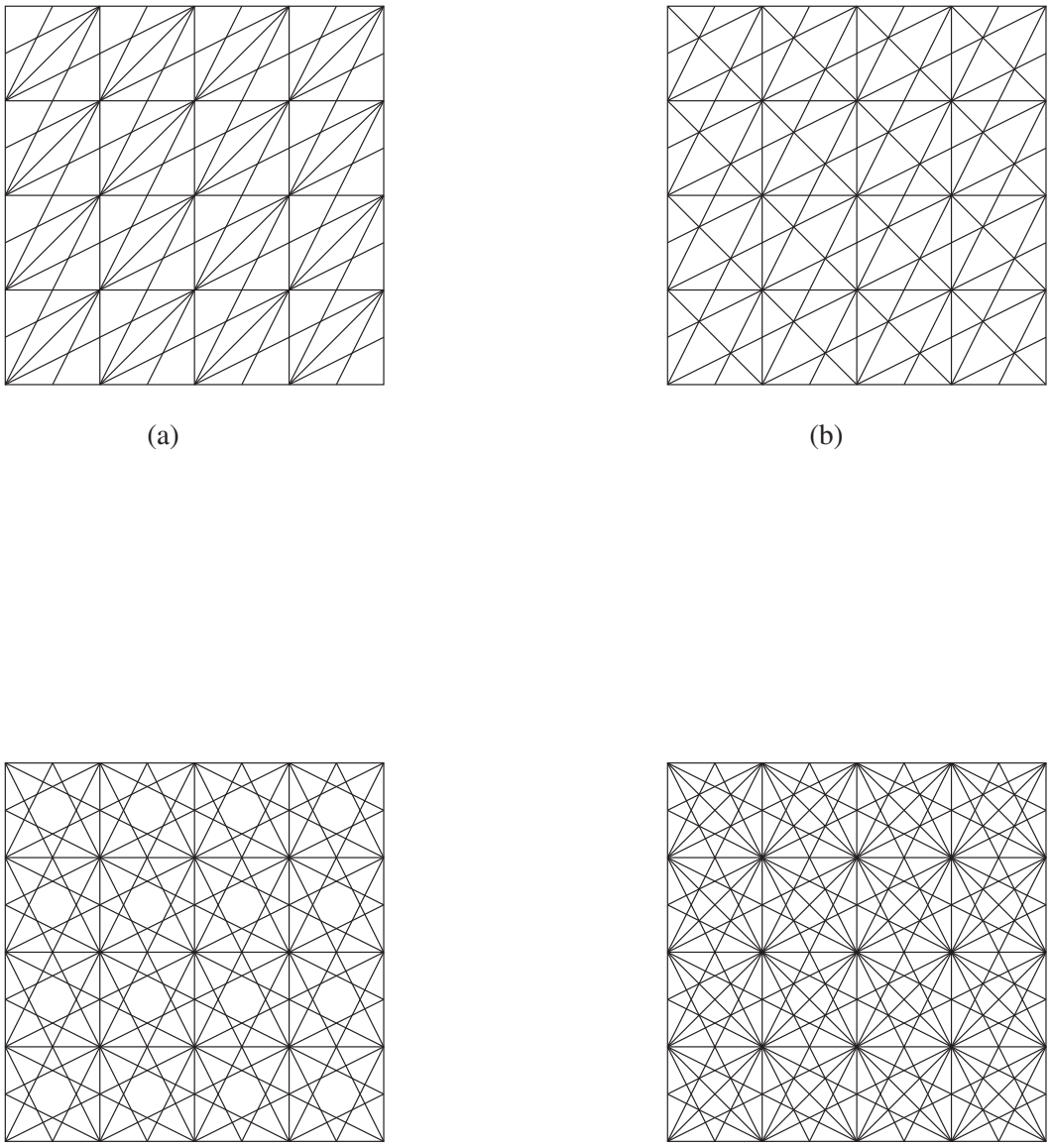


Figure 3.3: The support of some box splines



(a) (b)

(c) (d)

Figure 3.4: Various box spline grids

Chapter 4

Compactly Supported Box Spline Wavelets

The procedure for constructing multivariate non-tensor-product wavelets that generate a decomposition of $\mathcal{L}^2(\mathcal{R}^s)$, s ≥ 1 is described in this chapter and applied to yield explicit formulas for compactly supported spline wavelets based on the multiresolution analysis (MRA) of $\mathcal{L}^2(\mathcal{R}^s)$, $1 \leq s \leq 2$, generated by any box spline whose direction set constitutes a unimodular matrix.

4.1 Box Splines and Multiresolution Analysis

It is well known that if ϕ generates a MRA of $\mathcal{L}^2(\mathcal{R})$ in the sense of Mallat [26] and Meyer [27], and if ψ is a corresponding wavelet, then there are wavelets in $\mathcal{L}^2(\mathcal{R}^s)$ $s \geq 1$, given by $\Psi(x_1, ..., x_s) = \psi_1(x_1)....\psi_s(x_s)$, where all possible choices of $\psi_1...\psi_s$ from ϕ, ψ are included with exception of $\psi_1 = ... = \psi_s = \phi$. The wavelets Ψ so constructed are called wavelets of tensor-product type. For example, for $s = 2$ the wavelets of tensor-product type corresponding to ϕ, ψ are $\phi(x_1)\psi(x_2)$, $\phi(x_2)\psi(x_1)$, and $\psi(x_1)\psi(x_2)$. This chapter gives a procedure for constructing non-tensor-product wavelets in $\mathcal{L}^2(\mathcal{R}^s)$, s ≥ 1. To demonstrate this procedure, box splines with unimodular direction set $\mathcal{R}^s$, s= 1 and 2 are considered, yielding compactly supported non-tensor-product wavelets. In particular, since box splines are cardinal B-splines on $\mathcal{R}^1$, then minimally supported spline wavelets of Chui and Wang [11] are recovered for $s = 1$. Further the wavelets produced will be symmetric or antisymmetric.

Recall that in filtering theory, generalized linear phase filtering is achieved if and only if the filter function is symmetric or antisymmetric. In data compression if the filter does not have linear phase, then reconstructed images can be distorted . That is why symmetric (or antisymmetric) wavelets are essential for applications to data compression.That is why symmetric (or antisymmetric) wavelets are essential for applications to data

To facilitate the presentation, we must agree on certain notation. The symbol $\hat{f}$ will represent the Fourier transform of $f \in \mathcal{L}^2(\mathcal{R}^s)$ generated by

$$\hat{f}(\omega) = \int_{\mathcal{R}^s} f(x) e^{-i\omega x}\, dx \quad \omega \in \mathcal{R}^s$$

For an integer matrix $\Xi = (\xi^1, ..., \xi^n)$ whose columns $\xi^1, ..., \xi^n$ span $\mathcal{R}^s$, we denote by

$$\phi_\Xi(x) = N(x \mid \Xi)$$

the associated box spline on $\mathcal{R}^s$, whose Fourier transform is given by

$$\hat{\phi}_\Xi(\omega) = \prod_{k=1}^{n} (1 - e^{i \cdot \xi^k \cdot \omega})/(i \cdot \xi^k \cdot \omega). \tag{4.1}$$

It is well known that ϕ_Ξ has compact support [Ξ], is a piecewise polynomial of degree $n - s$, and lies in $C^{\rho-1}$ $(\mathcal{R}^s)$ with

$$\rho = \max\{r \ \mid \text{rank}(\Xi \backslash Y) = s \text{ for all } Y = (y_1, .., y_n) \subseteq \Xi\}$$

Dilations and translates of ϕ_Ξ generate the spaces

$$V_m(\Xi) := \text{clos}_{\mathcal{L}^2} \text{ span}\{\phi_\Xi(2^m \cdot -j) : j \in \mathcal{Z}^s\}, \text{m} \in \mathcal{Z}. \tag{4.2}$$

There is a crucial condition on matrix Ξ that is both necessary and sufficient for these spaces to form a MRA is in the sense of Mallat [26] and Meyer [27]. This property is defined by the requirement that any selection of s columns from Ξ that span $\mathcal{R}^s$ builds a unimodular matrix; i.e.,

$$\mid \det \text{Y} \mid = 1 \ \text{ for any basis } \ Y \subset \Xi \ \text{ of } \ \mathcal{R}^s \tag{4.3}$$

If condition 4.3 is satisfied, then the spaces $V_m(\Xi)$ enjoy all the properties of a MRA, namely

(i) $V_m \subset V_{m+1}$, $m \in \mathcal{Z}$;
(ii) $\bigcap_{m\in Z} V_m = \{0\}$, $\overline{\bigcup_{m\in Z} V_m} = \mathcal{L}^2(\mathcal{R}^s)$;
(iii) $f \in V_m \Longleftrightarrow f(2\cdot) \in V_{m+1}$;
(iv) for some $\phi \in V_0$, say $\phi := \phi_\Xi$, $V_m(\Xi) := \text{clos}_{\mathcal{L}^2}$ span $\{ \phi_\Xi(2^m \cdot -j)\}$;
(v) There exist constants A,B > 0 , such that for ϕ in (iv) and any sequence

$\{c_j\}_{j\in\mathcal{Z}^s} \in l^2$

$$A\|\{c_j\}\|_{l^2} \leq \left\|\sum c_j\phi(\cdot - j)\right\|_{\mathcal{L}^2} \leq B\|\{c_j\}\|_{l^2} \tag{4.4}$$

Definition 4.1 *Let $\psi \in \mathcal{L}^2(\mathcal{R}^s)$ and $\psi_{m,j}(x) := 2^{m/2}\psi(2^m x - j)$, $x \in \mathcal{R}^s$, $j \in \mathcal{Z}^s$, $m \in \mathcal{Z}$. Then the spaces generated by ψ are denoted by*

$$W_m := clos_{\mathcal{L}^2} span\{\psi(2^m \cdot -j) : j \in \mathcal{Z}^s\},\ m \in \mathcal{Z} \tag{4.5}$$

This family of subspaces of $\mathcal{L}^2(\mathcal{R}^s)$ gives a direct-sum decomposition of $\mathcal{L}^2(\mathcal{R}^s)$:

$$\mathcal{L}^2(\mathcal{R}^s) := \sum_{m\in\mathcal{Z}} W_m \tag{4.6}$$

if $\{\psi(\cdot - j) : j \in \mathcal{Z}^s\}$ is an unconditional basis of W_0 in the sense of Equation 4.4.

In general, the decomposition in Equation 4.6 means that for every $f \in \mathcal{L}^2(\mathcal{R}^s)$, we know that $f = \sum_{m\in\mathcal{Z}} f_m$ and $f_m \in W_m$. Properties (i) to (v) were proved for the spaces $V_m(\Xi)$ by Riemenschneider and Shen [31] and by Jia and Micchelli [22]. Here, we further explain properties (iii) and (v). We refer the reader to Chui, Stockler and Ward [8] for a detailed explanation. From Equation 4.1 we immediately have

$$\hat{\phi}_\Xi(\omega) = 2^{-s} P_\Xi(z)\hat{\phi}_\Xi(\omega/2)$$

with

$$P_\Xi(z) = 2^{-n+s}\prod_{k=1}^{n}(1 + z^{\xi^k}) \tag{4.7}$$

and

$$z = (z_1, ..., z_s) = (e^{-i\omega_1/2}, ..., e^{-i\omega_s/2}) \in T^s, \tag{4.8}$$

where T^s denotes the $s-$ torus in $\mathcal{C}^s$. We will assume that the relation between $z \in \mathcal{C}^s$ and $\omega \in \mathcal{R}^s$ is the one given by Equation 4.8. Equivalent to the above Fourier transform formulation is the two-scale relation

$$\phi_{\Xi}(x) = \sum_{j \in \mathcal{Z}^s} p_j \phi_{\Xi}(2x - j) \tag{4.9}$$

with $\{p_j\}$ being the coefficient sequence in the finite Taylor expansion of P_{Ξ} in Equation 4.7, namely

$$P_{\Xi}(z) = \sum_{j \in \mathcal{Z}^s} p_j z^j.$$

Property(iii) above is a consequence of Equation 4.9. Moreover, by condition 4.3 which we impose on Ξ, it follows that

$$\sum_{j \in \mathcal{Z}^s} \left| \hat{\phi}_{\Xi}(\omega/2 + 2\pi j) \right|^2 > 0 \quad \text{for all} \quad z \in T^s. \tag{4.10}$$

This is equivalent to property(v) for $\phi = \phi_{\Xi}$.

4.2 Construction of Wavelet Decompositions

In this section, a general method for constructing non-tensor-product wavelet decompositions relative to a certain class of functions ϕ in $\mathcal{L}^2(\mathcal{R}^s)$ is given. We assume the existence of a MRA of the spaces V_m generated by such functions ϕ as discussed in Section 4.2. We also have a nested sequence of subspaces $V_m,\ m \in \mathcal{Z}$, of $\mathcal{L}^2(\mathcal{R}^s)$ defined by

$$V_m = \cdots + W_{m-2} + W_{m-1}. \tag{4.11}$$

This leads to $V_{m+1} = V_m + W_m,\ m \in \mathcal{Z}$.

In what follows, we will describe a general construction process based on the knowledge of an initial decomposition of V_1, which need not be orthogonal. Finding an initial decomposition seems to be a central part of any wavelet construction. The construction of $\psi(x)$ will be uniquely determined by the construction of certain sequences $\{q_j\}_{j \in \mathcal{Z}^s} \in l^1$ such that

$$\psi(x) := \sum_{j \in \mathcal{Z}^s} q_j\ \phi(2x - j). \tag{4.12}$$

This function ψ also generates a closed subspace W_0 ($\subset V_1$) in the same manner that ϕ generates V_0. Of course the relation between the two subspaces V_0 and W_0 of V_1 must depend on the relation between two sequences $\{q_j\}_{j\in\mathcal{Z}^s} \in l^1$ and $\{p_j\}_{j\in\mathcal{Z}^s} \in l^1$. Some conditions on these sequences have to be imposed to assure that V_0 and W_0 are direct-sum decompositions of V_1 (see Chui [6], Chui, Stockler and Ward [8]). These sequences $\{p\}$, $\{q\}$ are called *reconstruction sequences.* Also central to our discussion are the *decomposition sequences* $\{g_j\}_{j\in\mathcal{Z}^s}$, $\{h_j\}_{j\in\mathcal{Z}^s} \in l^1$ where

$$\phi(2x-l) = 2^{-s} \sum_{j\in\mathcal{Z}^s} \left(g_{2j-l}\phi(x-j) + h_{2j-l}\psi(x-j)\right),\ l \in \mathcal{Z}^s. \qquad (4.13)$$

Equation 4.13 holds for all $x \in \mathcal{R}^s$ and describes the decomposition of V_1. Equation 4.13 is called as "decomposition relation". This relation can be easily proved for $\mathcal{L}^2(\mathcal{R}^s)$ in a manner similar to the proof for $\mathcal{L}^2(\mathcal{R})$ of Chui [6].

Our approach for determining reconstruction and decomposition sequences involves use of the inner product, a matrix formulation, an associated homogeneous system of equations and the determination of a null space. This requires only knowledge of linear algebra, in contrast to other functional analysis approaches. The basic steps in the approach are as follows:

- Select a consistent complementary subspace $W_j \perp V_j$ for all $j \in \mathcal{Z}^s$ and take an inner product of $\psi(x)$ and $\phi(x-l)$. Since $W_0 \perp V_0$, we know that $<\psi(x), \phi(x-l)> = 0$.

- Derive the governing equation for a matrix formulation:
  Substitute for $\psi(x)$ and $\phi(x-l)$ in $<\psi(x), \phi(x-l)> = 0$ with Equations 4.12 and 4.9 respectively. This yields the equations

  $$\sum_{k\in\mathcal{Z}^s} \sum_{r\in\mathcal{Z}^s} q_k\, p_{r-2l} < \phi(2x-k), \phi(2x-r) > = 0,\ x \in \mathcal{R}^s,\ l \in \mathcal{Z}^s. \quad (4.14)$$

  Note that index j in Equations 4.12 and 4.9 are replaced by indices r and k respectively to distinguish them.

- Fix the summation's indices for a particular l

- Find the number of nonzero q_k by analyzing a geometric representation of Equation 4.14 when indices vary for a fixed l.

- Produce the homogeneous system of equations using Equation 4.14 for those l for which $\phi(x-l)$ has support intersecting that of $\psi(x)$.
- Solve the homogeneous system of equations to find q_k.
  We can find a parametric solution for a homogeneous system of equations using the symbolic mathematical package MAPLE.
- Take the inner product of $\phi(x)$ and decomposition relation to find g_k and h_k

Since the symmetry (or antisymmetry) of the wavelet is an important property for practical applications, we can supply additional equations to the original homogeneous system of equations by imposing symmetry (or antisymmetry) conditions on the q_k. Our approach does not give orthonormal wavelets. In the next section, we illustrate this approach by considering box splines on $\mathcal{R}^1$ (Cardinal splines) and using a *general scale relation* instead of the two scale relation.

4.3 Construction of Wavelets from Cardinal Splines

In this section, we construct a wavelet decomposition based on the mth order cardinal splines $\phi_\Xi = N(\cdot|\Xi)$, where the direction matrix $\Xi = (\xi^1, \cdots, \xi^m)$, with $\xi^j = 1$ for all $1 \leq j \leq m$ satisfies condition 4.3. The center of the support of ϕ_Ξ is

$$C_\Xi = 1/2 \sum_{k=1}^{m} \xi^k. \tag{4.15}$$

Some equations given in previous sections in this chapter can be extended to *general scale* λ without loss of generality. Extending

$$V_j(\Xi) := \text{clos}_{\mathcal{L}^2} \text{ span}\{\phi_\Xi(\lambda^j \cdot -k) : k \in \mathcal{Z}^1\}, \text{ for } j, \lambda \in \mathcal{Z}, \ \lambda \geq 2,$$

$$W_j := \text{clos}_{\mathcal{L}^2} \text{ span}\{\psi(\lambda^j \cdot -r) : r \in \mathcal{Z}^1\}, \text{ for } j, \lambda \in \mathcal{Z} \ \lambda \geq 2,$$

Then the general scale relation becomes,

$$\phi_\Xi(x) = \sum_{r \in \mathcal{Z}^1} p_r \phi_\Xi(\lambda x - r) \tag{4.16}$$

where $\sum_{r\in\mathcal{Z}} p_r z^r = \lambda^{1-m}(1+z+z^2+\cdots+z^{\lambda-1})^m$.
The corresponding reconstruction sequences are given by

$$\psi_\Xi(x) = \sum_{k\in\mathcal{Z}^1} q_k \phi_\Xi(\lambda x - k). \tag{4.17}$$

The mathematical derivations of our approach can be broken down into the following steps.
Step 1: Select $W_0 \perp V_0$ and take an inner product of $\psi(x)$ and $\phi(x-l)$. Since $W_0 \perp V_0$, we know $<\psi(x), \phi_\Xi(x-l)> = 0$.
Step 2: To derive the governing equation, substitute for $\psi(x)$ and $\phi_\Xi(x-l)$ in $<\psi(x), \phi_\Xi(x-l)> = 0$ from Equations 4.16 and 4.17. Then replace r with $r-\lambda l$ without loss of generality. This leads to equation

$$\sum_{k\in\mathcal{Z}^1}\sum_{r\in\mathcal{Z}^1} q_k\ p_{r-\lambda l} < \phi_\Xi(\lambda x-k), \phi_\Xi(\lambda x - r) > = 0,\ x\in\mathcal{R}^1,\ l\in\mathcal{Z}^1. \tag{4.18}$$

By the definition of inner prodct,

$$< \phi_\Xi(\lambda x-k), \phi_\Xi(\lambda x-r) > = \lambda^{-1}\int N(y|\Xi)\ N(y-r+k|\Xi)\ dy,$$

where $y = \lambda x - r$.
Recall the inner product formula 3.24 for box splines, namely

$$N(x+2C_V|U\cup V) = \int N(y|U)\ N(y-x|V)\ dy,$$

where C_V is center of $N(\cdot|V)$.
Applying this formula to compute $<\phi_\Xi(\lambda x-k), \phi_\Xi(\lambda x-r)>$ gives,

$$\int N(y|\Xi)\ N(y-r+k|\Xi)\ dy\ = N(r-k+2C_\Xi|\Xi\cup\Xi),$$

where the center of $N(\cdot|\Xi\cup\Xi) = C_{\Xi\cup\Xi} = 2C_\Xi = m$.
Then Equation 4.18 becomes,

$$\sum_{k\in\mathcal{Z}^1}\sum_{r\in\mathcal{Z}^1} q_k\ p_{r-\lambda l} N(r-k+m|\Xi\cup\Xi)\ = 0,\ l\in\mathcal{Z}^1. \tag{4.19}$$

Step 3: Let us find the summation's indices in Equation 4.19 for a fixed l
(i) Since $\sum p_r z^r = \lambda^{1-m}(1+z+z^2+\cdots+z^{\lambda-1})^m$, $p_{r-\lambda l}$'s index-range is

$0 \longrightarrow (\lambda-1)m$. Then r varies over the range $\lambda l \longrightarrow \lambda l + (\lambda-1)m$.
(ii) It follows from the local support of the cardinal spline that $N(u|\Xi \cup \Xi) \equiv 0,\ \text{for}\ u \notin [0, 2m),\ u \in \mathcal{R}^1$, and that $N(r-k+m|\Xi \cup \Xi) > 0$ for $1 \leq r-k+m \leq 2m-1,\ r,\ k,\ m \in \mathcal{Z}^1$. Therefore $r-m+1 \leq k \leq r+m-1$ for fixed r.
(iii) From (i),(ii) we conclude that index k varies over the range $\lambda l - m + 1 \longrightarrow \lambda l + \lambda m - 1$.
By substituting (i),(ii) and (iii) into Equation 4.19, we get

$$\sum_{k=\lambda l-m+1}^{\lambda l+\lambda m-1} \sum_{r=\lambda l}^{\lambda l+(\lambda-1)m} q_k\ p_{r-\lambda l} N(r-k+m|\Xi \cup \Xi)\ = 0,\ l \in \mathcal{Z}^1. \tag{4.20}$$

Step 4: Here we will determine the number of nonzero q_k.
Equation 4.20 is easily transformed into the following homogeneous matrix form.

$$[\tilde{Q}]_{1\times K}\ [A]_{K\times R}\ [P]_{R\times 1}\ =\ 0 \tag{4.21}$$

where
$K = (\lambda+1)m - 1,\ R = (\lambda-1)m + 1,$
$[\tilde{Q}] = [q_{\lambda l-m+1}, \cdots, q_{\lambda l+\lambda m-1}],$
$a_{i,j,} = N(r-k+m|\Xi \cup \Xi) > 0,\ a_{i,j} \in [A],\ i = k - \lambda l + m,\ j = r - \lambda l + 1,$
and $[P] = [p_0, \cdots, p_{(\lambda-1)m}]^T$.
Let us observe some interesting features of Equation 4.21, such as invariance under l, by analyzing Equation 4.20:

- Consider what happens when indices k, r vary within their bounds in Equation 4.20

$$\begin{aligned} k &= \lambda l - m + 1 + \theta_k,\ \theta_k : 0 \longrightarrow (\lambda+1)m - 2,\ \theta_k \in \mathcal{Z}, \\ r &= \lambda l + \theta_r,\ \theta_r : 0 \longrightarrow (\lambda-1)m,\ \theta_r \in \mathcal{Z}. \end{aligned}$$

  Therefore $r-k+m = \theta_r - \theta_k + 2m - 1,\ r - \lambda l = \theta_r$ which implies $N(r-k-m|\Xi \cup \Xi)$ and $p_{r-\lambda l}$ do not depend on l, and thus $[A]$ and $[P]$ are independent of l.

- From index k's bounds in Equation 4.20, it is obvious that size of vector $[\tilde{Q}]$ is independent of l.

- Let $[A]$ $[P] = [B]_{K\times 1} = [b_0, \cdots, b_{\theta_k}, \cdots, b_{(\lambda+1)m-2}]$. Then from Equation 4.21, we have

$$b_0 q_{\lambda l-m+1} + b_1 q_{\lambda l-m+2} + \cdots + b_{\theta_k} q_{\lambda l-m+1+\theta_k} + \cdots + b_{(\lambda+1)m-2} q_{\lambda l+\lambda m-1} = 0 \tag{4.22}$$

  From the above observations, we deduce that b_{θ_k} remains unchanged and q_k associated with b_{θ_k} will change when we vary l to produce linear equations.

Further we assume that
(i) $q_k = 0$ if $0 \leq k \leq (\lambda + 1)m - 2$
(ii) $q_k = 0$ if $b_{\theta_k} = 0$ and $\theta_k = k$
(iii) Otherwise $q_k \neq 0$.

We have to find out how many q_k satisfy assumption (ii) in order to compute the actual number of nonzero q_k. Now our main task is to determine whether b_{θ_k} is nonzero or zero. Since [B] = [A][P], this task is accomplished by analyzing the geometric representation of [A][P]. We analyze the geometric representation shown in Figure 4.1.

- Figure 4.1(a): The range of θ_r such that $p_{\theta_r} > 0$, $\theta_r \in \mathcal{Z}$. $(\theta_r - \text{Range})$
- Figure 4.1(b): The range of u such that $N(u|\Xi \cup \Xi) > 0$, $u \in \mathcal{Z}$. (N-support)
- Figure 4.1(c): A P-Window is a $(\lambda-1)m$ length line segment. Its lower and upper ends are denoted by W_l and W_u.
- Figure 4.1(d): To compute the b_{θ_k},
  - (i) Place the P-Window over the one dimensional integer axis (where N-support lies) such that $W_l \equiv -\theta_k + 2m - 1$,
  - (ii) Find $S \equiv$ Overlapping region between P-Window and N-Support,
  - (iii) Compute $b_{\theta_k} = \sum N(j|\Xi \cup \Xi)p_i$ for all $i, j \in S \bigcap \mathcal{Z}$, where point i in P-window lies exactly over point j in N-Support within S.

- Figure 4.1(e): Compute $[B] = [b_0, \cdots, b_{(\lambda+1)m-2}]$ by placing P-Window over N-Support at integer points between point $X = 2m-1$ and point $Y = -(\lambda-1)m+1$.

It is important to note that we do not actually compute the value of b_{θ_k} by analyzing the geometric representation, we just determine whether b_{θ_k} is nonzero or zero. It follows from Figure 4.1(e) that there is at least one overlapping point at any instant of computation of [B]. Then there is no $b_{\theta_k} = 0$ which implies that no q_k satisfies assumption (ii). Therefore the number of nonzero q_k is given by $(\lambda+1)m-1$ and these q_k are denoted as $[Q]^T_{K\times 1} = [q_0, \cdots, q_{(\lambda+1)m-2}]$.

Step 5: Now we analyze how many equations can be produced to determine $[q_0, \cdots, q_{(\lambda+1)m-2}]$.
The homogeneous system of equations can be produced by computing Equation 4.22 for different values of l. Some conditions are imposed on l to assure that these equations consist of at least some of the elements of $\{q_0, \cdots, q_{(\lambda+1)m-2}\}$, namely

$$\lambda l + \lambda m - 1 \geq 0 \text{ and } \lambda l - m + 1 \leq (\lambda+1)m - 2$$

which can be rewritten as

$$1 - \lambda m \leq \lambda l \leq (\lambda+2)m - 3.$$

Since $l \in \mathcal{Z}$, we have $1-m \leq l \leq m+div\{(2m-3)/\lambda\}$. It is worth noting that when $\lambda = 2$, we can produce $3m-2$ equations to determine $\{q_0, \cdots, q_{3m-2}\}$.

Step 6: It follows from **Step 5** that we have a homogeneous system of equations in the form of $[H]_{L\times K}[Q]_{K\times 1} = 0$ where $K = (\lambda+1)m-1$ and $L = 2m+div\{(2m-3)/\lambda\}$. We can automatically find a parametric solution to this system using MAPLE.

The mathematical derivations given in above steps and the symmetry or antisymmetry condition of the wavelets lead to the following theorem.

Theorem 4.1 *The recontruction sequences $\{q\}$ that are constructed from m^{th} order cardinal splines, are determined by solving homogeneous system of*

equations with $\tilde{L}$ number of equations and M number of unknowns, namely $[\tilde{H}]_{\tilde{L}\times M}[Q]_{M\times 1} = 0$. The equations are generated by using Equation 4.22 and imposing the symmetry or antisymmetry condition. The rank of matrix $[\tilde{H}]$ is less than M. M and $\tilde{L}$ are given by following equations.

$$\begin{aligned} M &= (\lambda+1)m - 1 \\ \textit{and} & \\ \tilde{L} &= 2m + div\{(2m-3)/\lambda\} + div\{M/2\}. \end{aligned}$$

Where $\lambda \geq 2,\ m \geq 1$ and $\lambda, m \in \mathcal{Z}$.

We will look at some numerical examples now. At end of this chapter, the q_k for these examples are tabulated.

EXAMPLE 1 :- Consider the 4th order cardinal spline under the two scale relation ($\lambda = 2$).
See Appendix A for MAPLE codes, matrices, condition for *wavelet decompositions* and parameters in this example. We have a homogeneous system of equations with $[H]_{10\times 11}$ and $[Q]_{11\times 1}$, i.e., ten equations and eleven unknowns. The q_k in parametric form are symmetric, depend on parameter $t1$ and they satisfy the necessary condition for *wavelet decompositions*, namely

$$\sum_{k\in\mathcal{Z}} q_k \hat{\phi}_{\Xi}(0) = 0 \implies \sum_{k=0}^{10} q_k = 0$$

regardless of the value of parameter $t1$. This implies that we can produce an infinite variety of wavelets by varying the value of parameter $t1$. In particular, when $t1 = 1/40320$ we get the minimally supported wavelets of Chui and Wang [11], i.e.,

$$q_k = (-1)^k \sum_{l=0}^{m} p_l N(k+1-l|\Xi \cup \Xi) \tag{4.23}$$

where $k = 0, \cdots, 3m-2$, $\sum p_r z^r = 2^{1-m}(1+z)^m$, $N(\cdot|\Xi \cup \Xi) = 2m^{th}$ order cardinal spline and $m = 4$.

EXAMPLE 2 :- Consider the 4th order cardinal spline under the four scale relation ($\lambda = 4$).
See Appendix B for MAPLE codes, matrices, condition for *wavelet decompositions* and parameters in this example. We have a homogeneous system

of equations with $[H]_{9\times 19}$ and $[Q]_{19\times 1}$, i.e., nine equations and nineteen unknowns. The q_k in parametric form satisfy the necessary condition for *wavelet decompositions*, namely

$$\sum_{k\in\mathcal{Z}} q_k \hat{\phi}_{\Xi}(0) = 0 \implies \sum_{k=0}^{19} q_k = 0$$

regardless of the values of parameters. We need more equations in order to have a desirable solution with few parameters. As mentioned earlier, the symmetry (or antisymmetry) of the wavelet is an important property for practical applications such as data compression. We can get nine additional equations by imposing a symmetry condition on the q_k. Now we get a new homogeneous system of equations with $[H1]_{18\times 19}$ and $[Q1]_{19\times 1}$, i.e., eighteen equations and nineteen unknowns. Although there are eighteen equations in $[H1]$, the rank of $[H1]$ is sixteen. By substituting t1=25196v, t2=1681v, t3=v we will get a new solution [Q2]. Surprisingly the new solution $[Q2]$ (when v=1/322560) is generated by the equation

$$q_k = (-1)^k \sum_{l=0}^{(\lambda-1)m} p_l N(k+1-l|\Xi\cup\Xi) \tag{4.24}$$

where $k = 0, \cdots, (\lambda+1)m-2$, $\sum p_r z^r = \lambda^{1-m}(1+z+z^2+\cdots+z^{\lambda-1})^m$, $N(\cdot|\Xi\cup\Xi) = 2m^{th}$order cardinal spline, $\lambda = 4$ and $m = 4$. The Equation 4.24 is a generalized version of Equation 4.23 (see Chui [11]).

EXAMPLE 3 :- Consider the 4th order cardinal spline under the eight scale relation ($\lambda = 8$).
Here we will create $[Q]_{35\times 1}$ using Equation 4.24 with $\lambda = 8$ and test whether $[Q]$ will satisfy $[H]_{8\times 35}[Q]_{35\times 1} = [0]$. Further it is shown that $[Q]$ satisfies the necessary condition for *wavelet decompositions*, namely $\sum_{k=0}^{35} q_k = 0$ and $[H][Q] = [0]$. See Appendix C for MAPLE codes.

Now we return to the steps of our algorithm.

Step 7: Here we will give an approach for determining the decomposition sequences .
The decomposition relation 4.13 (related to the two scale relation) for cardinal splines

$$\phi_{\Xi}(2x-l) = 2^{-1} \sum_{j\in\mathcal{Z}^1} g_{2j-l}\phi_{\Xi}(x-j) + h_{2j-l}\psi_{\Xi}(x-j),\ l\in\mathcal{Z}^1, \tag{4.25}$$

holds for all $x \in \mathcal{R}^1$.
Taking the inner product of the $\phi_\Xi(x)$ and Equation 4.25 yields

$$< \phi_\Xi(2x-l), \phi_\Xi(x) > = 1/2 \sum_{j \in \mathcal{Z}^1} g_{2j-l} < \phi_\Xi(x-j), \phi_\Xi(x) > \qquad (4.26)$$

because $V_0 \perp W_0$. Further applying the inner product formula 3.24 and Equation 4.9 to Equation 4.26 gives

$$\sum_{r=0}^{m} p_r N(r-l+m|\Xi \cup \Xi) = \sum_{j \in \mathcal{Z}^1} g_{2j-l} N(j+m|\Xi \cup \Xi). \qquad (4.27)$$

This produces a linear system of equations to determine g_k by varying l in Equation 4.27. The number of equations and the number of unknowns depend on truncation of the infinite decomposition sequences. The decomposition sequences must be truncated in order to apply the Pyramid algorithm. When the finite decomposition sequences are used instead of the original infinite decomposition sequences, there will be some discrepancy. We refer the reader to the text by C. Chui [6] for more details.

Similarly, taking the inner product of $\psi_\Xi(x)$ and Equation 4.25 and then applying the inner product formula 3.24 and (4.12) gives

$$2 \sum_{k=0}^{3m-2} q_k N(k-l+m|\Xi \cup \Xi) = \sum_{j \in \mathcal{Z}^1} \sum_{k=0}^{3m-2} \sum_{r=0}^{3m-2} q_k q_r h_{2j-l} N(j+r-k|\Xi \cup \Xi). \qquad (4.28)$$

Again, we can produce a linear system of equations from Equation 4.28 to find h_k.

We have presented all the steps of our approach. If $\lambda = 2$ and $s = 1$, we always produce a number of equations equal to one less than number of q_k and get the solution in one parameter. By assigning an appropriate value for this parameter, we recover the minimally supported wavelets of Chui and Wang [11]. If $\lambda > 2$, we produce fewer equations than the number of q_k . In this case we can supply some additional equations by imposing symmetry (or antisymmetry) conditions on the q_k. However, for all cases of λ, the q_k can be generated by

$$q_k = (-1)^k \sum_{l=0}^{(\lambda-1)m} p_l N(k+1-l|\Xi \cup \Xi), \ \lambda \geq 2 \text{ and } \lambda \in \mathcal{Z}$$

where $k = 0, \cdots, (\lambda+1)m-2$, $\sum p_r z^r = \lambda^{1-m}(1+z+z^2+\cdots+z^{\lambda-1)})^m$, $N(\cdot|\Xi \cup \Xi) = 2m^{th}$order cardinal spline.

We conclude that the construction of cardinal splines wavelets would be implemented by performing following steps:

- Choose $W_0 \perp V_0$.
- Select scale λ and order of the cardnial spline m.
- Create matrix $[\tilde{H}]$ from Equation 4.22 by increasing l from $1-m$ to $m+$ $div\{(2m-3)/\lambda\}$ and from the symmetry or antisymmetry conditions.
- Solve the homogeneous system of equations $[\tilde{H}][Q] = 0$ to determine the reconstruction sequences $\{q_k\}$.

In the next chapter we will apply our approach to bivariate three direction box splines.

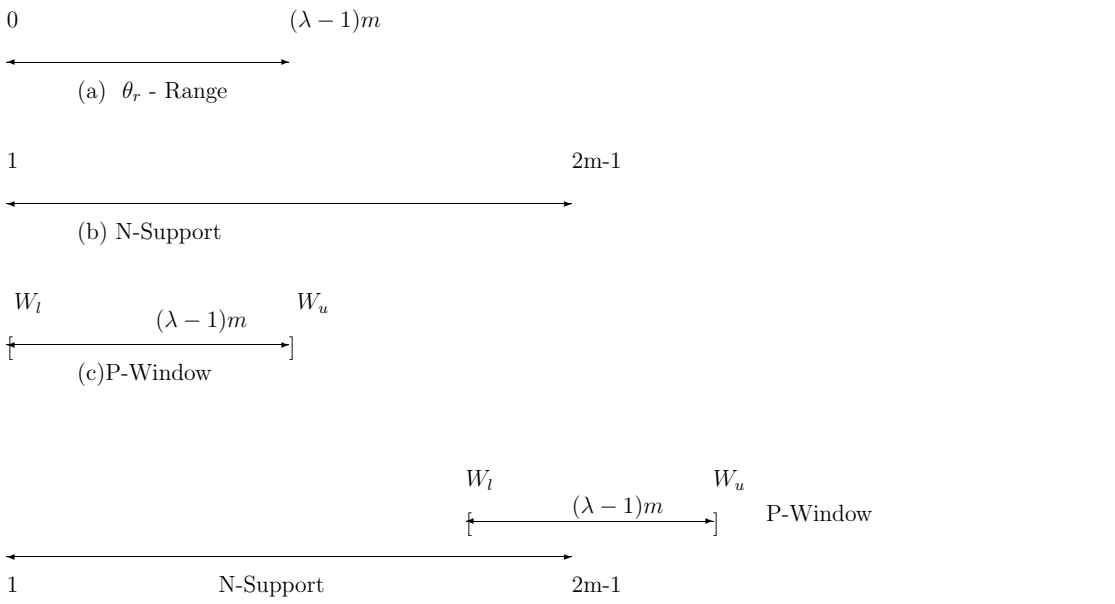


Figure 4.1: The geometric representation for computation of [B]

Cardinal Spline Wavelets For Two Scale & Four Scale Relations

The fourth order cardinal spline is considered.

$$Q = \begin{pmatrix} q_0 &= & 1 \\ q_1 &= & -124 \\ q_2 &= & 1677 \\ q_3 &= & -7904 \\ q_4 &= & 18482 \\ q_5 &= & -24264 \\ q_6 &= & -18482 \\ q_7 &= & -7904 \\ q_8 &= & 1677 \\ q_9 &= & -124 \\ q_{10} &= & 1 \end{pmatrix} \qquad Q = \begin{pmatrix} q_0 &= & 1 \\ q_1 &= & -124 \\ q_2 &= & 1681 \\ q_3 &= & -8400 \\ q_4 &= & 25196 \\ q_5 &= & -56624 \\ q_6 &= & 102476 \\ q_7 &= & -152880 \\ q_8 &= & 193206 \\ q_9 &= & -209064 \\ q_{10} &= & 193206 \\ q_{11} &= & -152880 \\ q_{12} &= & 102476 \\ q_{13} &= & -56624 \\ q_{14} &= & 25196 \\ q_{15} &= & -8400 \\ q_{16} &= & 1681 \\ q_{17} &= & -124 \\ q_{18} &= & 1 \end{pmatrix}$$

Two Scale q's Four Scale q's

Table 4.1: Results of Examples 1 and 2

Cardinal Spline Wavelets For Eight Scale Relation

The fourth order cardinal spline is considered.

$$Q = \begin{pmatrix} q_0 &= & 1 \\ q_1 &= & -124 \\ q_2 &= & 1681 \\ q_3 &= & -8400 \\ q_4 &= & 25200 \\ q_5 &= & -57120 \\ q_6 &= & 109200 \\ q_7 &= & -186480 \\ q_8 &= & 293996 \\ q_9 &= & -436304 \\ q_{10} &= & 613196 \\ q_{11} &= & -814800 \\ q_{12} &= & 1026480 \\ q_{13} &= & -1233120 \\ q_{14} &= & 1419600 \\ q_{15} &= & -1570800 \\ q_{16} &= & 1671606 \\ q_{17} &= & -1707624 \\ q_{18} &= & 1671606 \\ q_{19} &= & -1570800 \\ q_{20} &= & 1419600 \\ q_{21} &= & -1233120 \\ q_{22} &= & 1026480 \\ q_{23} &= & -814800 \\ q_{24} &= & 613196 \\ q_{25} &= & -436304 \\ q_{26} &= & 293996 \\ q_{27} &= & -186480 \\ q_{28} &= & 109200 \\ q_{29} &= & -57120 \\ q_{30} &= & 25200 \\ q_{31} &= & -8400 \\ q_{32} &= & 1681 \\ q_{33} &= & -124 \\ q_{34} &= & 1 \end{pmatrix}$$

Table 4.2: Result of Example 3

Chapter 5

Construction of Wavelets from Box Splines

Box spline wavelets can be constructed by applying our approach as presented in Chapter 4. Each step of our extended multivariate approach is explained in detail. However, we will report some of the results without a detailed explanation, because these results are just an extension of the work done in Section 4.3.

5.1 Bivariate Three Directional Box splines Wavelets

In this section, we wish to apply our approach to the bivariate three direction box splines. More specifically, let $\Xi = (\xi^1 : m_1, \xi^2 : m_2, \xi^3 : m_3)$ be the direction matrix with m_1, m_2, m_3 denoting multiplicities of the direction vectors $\xi^1 = (1,0)^T, \xi^2 = (0,1)^T$ and $\xi^3 = (1,1)^T$. Thus, $\phi_\Xi = N(\cdot|\Xi)$ is a bivariate function satisfying condition 4.3, and its support is a hexagonal subset of the rectangle $[0, m_1 + m_3] \times [0, m_2 + m_3]$. The center of the support of ϕ_Ξ is

$$C_\Xi = 1/2 \sum_{k=1}^{m_1+m_2+m_3} \xi^k \ = (m_1 + m_3, m_2 + m_3)^T/2. \tag{5.1}$$

Some equations given in construction of univariate cardinal spline wavelet (Section 4.3) can be extended these to bivariate three direction box splines

without loss of generality:

$$V_j(\Xi) := \text{clos}_{L^2}\ \text{span}\{\phi_\Xi(\lambda^j \cdot -k) : k \in \mathcal{Z}^2\},\ \text{for } j, \lambda \in \mathcal{Z},\ \lambda \geq 2.$$

$$W_j := \text{clos}_{L^2}\ \text{span}\{\psi(\lambda^j \cdot -r) : r \in \mathcal{Z}^2\},\ \text{for } j, \lambda \in \mathcal{Z}\ \lambda \geq 2.$$

The general scale relation is,

$$\phi_\Xi(x_1, x_2) = \sum_{r_1, r_2 \in \mathcal{Z}^1} p_{r_1,r_2} \phi_\Xi(\lambda x_1 - r_1, \lambda x_2 - r_2), \tag{5.2}$$

where

$$\sum_{r_1,r_2 \in \mathcal{Z}} p_{r_1,r_2} w^{r_1} z^{r_2} =$$
$$\lambda^{2-m}(1 + w + \cdots + w^{\lambda-1})^{m_1}(1 + z + \cdots + z^{\lambda-1})^{m_2}(1 + wz + \cdots + w^{\lambda-1}z^{\lambda-1})^{m_3}$$

and $m = m_1 + m_2 + m_3$.
Reconstruction sequences are given by

$$\psi_\Xi(x_1, x_2) = \sum_{k_1,k_2 \in \mathcal{Z}^1} q_{k_1,k_2} \phi_\Xi(\lambda x_1 - k_1, \lambda x_2 - k_2). \tag{5.3}$$

Now we apply our approach.
Step 1: Select $W_0 \perp V_0$. Since $W_0 \perp V_0$, $< \psi(x_1, x_2), \phi_\Xi(x_1 - l_1, x_2 - l_2) > = 0$ for $l_1, l_2 \in \mathcal{Z}^1$.
Step 2: The governing equation is just an extension of the governing equation of a cardinal spline wavelet Equation 4.19:

$$\sum_{k_1,k_2 \in \mathcal{Z}^1} \sum_{r_1,r_2 \in \mathcal{Z}^1} q_{k_1,k_2}\ p_{r_1 - \lambda l_1, r_2 - \lambda l_2} N(u, v | \Xi \cup \Xi)\ = 0, \tag{5.4}$$

where $u = r_1 - k_1 + m_1 + m_3$ and $v = r_2 - k_2 + m_2 + m_3$.
Step 3: Let us find the summation's indices in Equation 5.4 for a fixed l

- (i) Since

  $$\sum p_{r_1,r_2} w^{r_1} z^{r_2} =$$
  $$\lambda^{2-m}(1 + w + \cdots + w^{\lambda-1})^{m_1}(1 + z + \cdots + z^{\lambda-1})^{m_2}(1 + wz + \cdots + w^{\lambda-1}z^{\lambda-1})^{m_3}$$

  $p_{r_1 - \lambda l_1, r_2 - \lambda l_2}$'s index-range is $0 \longrightarrow (\lambda - 1)(m_1 + m_3)$, $0 \longrightarrow (\lambda - 1)(m_2 + m_3)$. Then r_1 varies over the range $\lambda l_1 \longrightarrow \lambda l_1 + (\lambda - 1)(m_1 + m_3)$ and r_2 varies over the range $\lambda l_2 \longrightarrow \lambda l_2 + (\lambda - 1)(m_2 + m_3)$.

- (ii) It follows from the local support of the box spline that

$$N(U_1, U_2|\Xi \cup \Xi) \equiv 0,$$

for $U_1 \notin [0, 2(m_1 + m_3),\ U_2 \notin [0, 2(m_2 + m_3),\ U_1, U_2 \in \mathcal{R}^1$.
It also follows that

$$N(r_1 - k_1 + m_1 + m_3, r_2 - k_2 + m_2 + m_3|\Xi \cup \Xi) > 0,$$

for $1 \leq r_1 - k_1 + m_1 + m_3 \leq 2(m_1 + m_3) - 1$,
$1 \leq r_2 - k_2 + m_2 + m_3 \leq 2(m_2 + m_3) - 1$,
$r_1, r_2,\ k_1, k_2,\ m_1, m_2, m_3 \in \mathcal{Z}^1$.
Therefore

$$r_1 - m_1 - m_3 + 1 \leq k_1 \leq r_1 + m_1 + m_3 - 1$$
$$r_2 - m_2 - m_3 + 1 \leq k_2 \leq r_2 + m_2 + m_3 - 1$$

for fixed r_1, r_2.

- (iii) From (i) and (ii) we conclude that index k_1 varies over the range

$$\lambda l_1 - m_1 - m_3 + 1 \longrightarrow \lambda l_1 + \lambda(m_1 + m_3) - 1$$

and index $k_2$varies over the range

$$\lambda l_2 - m_2 - m_3 + 1 \longrightarrow \lambda l_2 + \lambda(m_2 + m_3) - 1.$$

By substituting (i),(ii),(iii) into Equation 5.4, we get

$$\sum_{k_1=L_{k1}}^{U_{k1}} \sum_{k_2=L_{k2}}^{U_{k2}} \sum_{r_1=L_{r1}}^{U_{r1}} \sum_{r_2=L_{r2}}^{U_{r2}} q_{k_1,k_2}\ p_{r_1-\lambda l_1, r_2-\lambda l_2} N(u, v|\Xi \cup \Xi)\ = 0 \qquad (5.5)$$

where $u = r_1 - k_1 + m_1 + m_3,\ \ v = r_2 - k_2 + m_2 + m_3$,
$L_{k1} = \lambda l_1 - m_1 - m_3 + 1,\ \ U_{k1} = \lambda l_1 + \lambda(m_1 + m_3) - 1$,
$L_{k2} = \lambda l_2 - m_2 - m_3 + 1,\ \ U_{k2} = \lambda l_2 + \lambda(m_2 + m_3) - 1$,
$L_{r1} = \lambda l_1,\ \ U_{r1} = \lambda l_1 + (\lambda - 1)(m_1 + m_3)$,
$L_{r2} = \lambda l_2,\ \ U_{r2} = \lambda l_2 + (\lambda - 1)(m_2 + m_3)$.

Step 4: Here we will determine the number of nonzero q_k.
Equation 5.5 can be transformed into following matrix form.

$$[\tilde{Q}]_{1\times K}\ [A]_{K\times R}\ [P]_{R\times 1}\ =\ 0 \tag{5.6}$$

where $K = (U_{k1}-L_{k1}+1)\times(U_{k2}-L_{k2}+1)$, $R = (U_{r1}-L_{r1}+1)\times(U_{r2}-L_{r2}+1)$,
$[\tilde{Q}] = [q_{L_{k1},L_{k2}}, \cdots, q_{U_{k1},U_{k2}}]$,
$a_{i,j} = N(r_1 - k_1 + m_1 + m_3, r_2 - k_2 + m_2 + m_3 | \Xi \cup \Xi)$, $a_{i,j} \in [A]$,
$i = (k_1 - L_{k1})K + (k_2 - L_{k2} + 1)$, $j = (r_1 - L_{r1})R + (r_2 - L_{r2} + 1)$,
and $[P] = [p_{0,0}, \cdots, p_{(U_{r1}-L_{r1}),(U_{r2}-L_{r2})}]^T$.

As we analyzed Equation 4.21, it can be shown that matrices $[A], [P]$ and the size of vector $[\tilde{Q}]$ are independent of l_1, l_2. Let

$$[A]\ [P] = [B]_{K\times 1} = [b_{0,0}, \cdots, b_{\theta_{k_1},k_2}, \cdots, b_{(U_{k1}-L_{k1}),(U_{k2}-L_{k2})}].$$

Then from Equation 5.6 we have that

$$b_{0,0}q_{L_{k1},L_{k2}} + \cdots + b_{\theta_{k_1},\theta_{k_2}}q_{L_{k1}+\theta_{k1},L_{k2}+\theta_{k_2}} + \cdots + b_{(U_{k1}-L_{k1}),(U_{k2}-L_{k2})}q_{U_{k1},U_{k2}} = 0 \tag{5.7}$$

where $\theta_{k_1} : 0 \longrightarrow U_{k1} - L_{k1}$, $\theta_{k_2} : 0 \longrightarrow U_{k2} - L_{k2}$
Further we assume that
(i) $q_{k_1,k_2} = 0$ if $0 \not\leq k_1 \not\leq U_{k1} - L_{k1}$, $0 \not\leq k_2 \not\leq U_{k2} - L_{k2}$
(ii) $q_{k_1,k_2} = 0$ if $b_{\theta_{k_1},\theta_{k_2}} = 0$ and $\theta_{k_1} = k_1$, $\theta_{k_2} = k_2$
(iii) Otherwise $q_{k_1,k_2} \neq 0$.

We have to find out how many q_{k_1,k_2} satisfy assumption (ii) in order to compute the actual number of nonzero q_{k_1,k_2}. The q_{k_1,k_2} satisfying assumption (ii) are determined by analyzing the geometric representation of $[A][P]$. Here we will give the results and defer the analysis until the next section.

- Result 1: Index r_1, r_2 of p_{r_1,r_2} is a hexagonal subset of the integer lattice (Figure 5.1(a))
- Result 2: Index $\theta_{k_1}, \theta_{k_2}$ of $b_{\theta_{k_1},\theta_{k_2}}$ is a hexagonal subset of the integer lattice (Figure 5.1(b))
- Result 3: Since we know the indices of $[B]$ and $[P]$, we can reformulate the Equation 5.6 by introducing permutation matrices $[X]$ and $[Y]$ (see subsection 5.21).

$$[\tilde{Q}]_{1\times K}\ [X][X]^T\ [A]_{K\times R}\ [Y][Y]^T\ [P]_{R\times 1}\ =\ 0 \tag{5.8}$$

where $[Y] = [I_{1,f(1)}][[I_{2,f(2)}][I_{3,f(3)}]\cdots[I_{R,f(R)}]$, $[X] = [I_{1,\hat{f}(1)}][[I_{2,\hat{f}(2)}]\cdots[I_{K,\hat{f}(K)}]$,
$[I_{i,f(i)}]$ = columns i and f_i are interchanged in identity matrix $[I]_{R\times R}$,
$[I_{j,\hat{f}(j)}]$ = columns j and $\hat{f}_j$ are interchanged in identity matrix $[I]_{K\times K}$,
$f(i) = i + 1/2N_1(N_1 - 2(\lambda-1)m_3 - 1) - 1/2N_2(N_2+1)$,
$\hat{f}(j) = j + 1/2\hat{N}_1(\hat{N}_1 - 2(\lambda+1)m_3 + 1) - 1/2\hat{N}_2(\hat{N}_2+1)$,
$N_1 = min(l, (\lambda-1)m_3 - 1)$, $\hat{N}_1 = min(\hat{l}, (\lambda+1)m_3 - 2)$,
$N_2 = max(l - (\lambda-1)m_1, 0)$, $\hat{N}_2 = max(\hat{l} - (\lambda+1)m_1 + 1, 0)$,
$l = div\{(i-1)/((\lambda-1)(m_1+m_3)+1)\} + 1$,
$\hat{l} = div\{(j-1)/((\lambda+1)(m_1+m_3)-1)\} + 1$.

- Result 4: From assumption (ii) and Result 2, index k_1, k_2 of q_{k_1,k_2} is the same as index $\theta_{k_1}, \theta_{k_2}$ of $b_{\theta_{k_1},\theta_{k_2}}$ (Figure 5.1(a)). After some calculation we find that number of nonzero $q_{k_1,k_2} = M$ is given by

  $$M = (\lambda+1)^2(m_1m_2 + m_2m_3 + m_3m_1) - (\lambda+1)(m_1+m_2+m_3) + 1.$$

  These nonzero q_{k_1,k_2} are denoted as $[Q]^T_{M\times 1} = [q_{0,0}, \cdots, q_{U_{k1}-L_{k1},U_{k2}-L_{k2}}]$.

Now we return to the next step of our approach.

Step 5: We analyze how many equations can be produced to determine $[Q]$. The homogeneous system of equations can be produced by computing Equation 5.7 for different values of l_1, l_2. Some conditions are imposed on l_1, l_2 so as to assure that all the equations consisting of some elements of $[Q]$ will be produced by varying l_1, l_2, namely
$U_{k1} \geq 0$ and $L_{k1} \leq U_{k1} - L_{k1} \implies 1 - \lambda(m_1+m_3) \leq \lambda l_1 \leq (\lambda+2)(m_1+m_3) - 3$
$U_{k2} \geq 0$ and $L_{k2} \leq U_{k2} - L_{k2} \implies 1 - \lambda(m_2+m_3) \leq \lambda l_2 \leq (\lambda+2)(m_2+m_3) - 3$
Since $l_1, l_2 \in \mathcal{Z}$, we know that
$l_1^{min} = 1 - m_1 + m_3 \leq l_1 \leq m_1 + m_3 + div\{(2(m_1+m_3) - 3)/\lambda\} = l_1^{max}$,
$l_2^{min} = 1 - m_2 + m_3 \leq l_2 \leq m_2 + m_3 + div\{(2(m_2+m_3) - 3)/\lambda\} = l_2^{max}$.

Step 6: It follows from **Step 5** that we have a homogeneous system of equations in the form of $[H]_{L\times M}[Q]_{M\times 1} = 0$ where $L = (l_1^{max} - l_1^{min} + 1)(l_2^{max} - l_2^{min} + 1)$ and $M = (\lambda+1)^2(m_1m_2 + m_2m_3 + m_3m_1) - (\lambda+1)(m_1+m_2+m_3) + 1$. We can find a parametric solution to this system using MAPLE.

The mathematical derivations given in above steps and the symmetry or antisymmetry condition of the wavelets lead to the following theorem.

Theorem 5.1 *The recontruction sequences $\{q\}$ that are constructed from a bivariate three direction box spline box with direction vectors $\xi^1 = (1,0)^T$, $\xi^2 = (0,1)^T$, $\xi^3 = (1,1)^T$ having multiplicities m_1, m_2 and m_3, are determined by solving homogeneous system of equations with $\tilde{L}$ number of equations and M number of unknowns, namely $[\tilde{H}]_{\tilde{L}\times M}[Q]_{M\times 1} = 0$. The equations are generated by using Equation 5.7 and imposing the symmetry or antisymmetry condition. The rank of matrix $[\tilde{H}]$ is less than M. M and $\tilde{L}$ are given by following equations.*

$$M = (\lambda+1)^2(m_1m_2 + m_2m_3 + m_3m_1) - (\lambda+1)(m_1+m_2+m_3) + 1$$

and

$$\tilde{L} = (l_1^{max} - l_1^{min} + 1)(l_2^{max} - l_2^{min} + 1) + div\{M/2\}.$$

Where
$l_1^{min} = 1 - m_1 + m_3$, $l_1^{max} = m_1 + m_3 + div\{(2(m_1+m_3) - 3)/\lambda\}$,
$l_2^{min} = 1 - m_2 + m_3$, $l_2^{max} = m_2 + m_3 + div\{(2(m_2+m_3) - 3)/\lambda\}$,
$\lambda \geq 2,\ m_1 \geq 1,\ m_2 \geq 1,\ m_3 \geq 1$ *and* $\lambda, m_1, m_2, m_3 \in \mathcal{Z}$.

We look at some numerical examples now. At end of this chapter, the q_{k_1,k_2} for these examples are tabulated.

EXAMPLE 1 :- Consider the two scale relation ($\lambda = 2$) and the box spline with direction vectors $\xi^1 = (1,0)^T$, $\xi^2 = (0,1)^T$, $\xi^3 = (1,1)^T$. We set the multiplicities of direction vectors to be 1, 1 and 2 respectively. See Appendix D for MAPLE codes, matrices, condition for *wavelet decompositions* and parameters in this example. We will have a homogeneous system of equations with $[H]_{29\times 34}$and $[Q]_{34\times 1}$ after eliminating trivial equations from the original forty nine ($L = 49$) equations. The q_{k_1,k_2} in parametric form satisfy the necessary condition for *wavelet decompositions*, namely

$$\sum_{k_1,k_2\in\mathcal{Z}} q_{k_1,k_2}\hat{\phi}_{\Xi}(0,0) = 0 \implies \sum_{k_1=0}^{7}\sum_{k_2=0}^{7} q_{k_1,k_2} = 0,$$

regardless of the values of parameters. We need more equations in order to have a desirable solution with few parameters. As mentioned earlier the symmetry (or antisymmetry) of the wavelet is an important property for practical applications such as data compression. We can get seventeen additional equations by imposing a condition of antisymmetry on q_{k_1,k_2}. The

q_{k_1,k_2} are antisymmetric about the center of their hexagonal support. Now we get a new homogeneous system of equations with $[H1]_{46\times 34}$ and $[Q1]_{34\times 1}$. Although there are forty six equations in $[H1]$, the rank of $[H1]$ is thirty one. By substituting v1= -2v, v2= -3v, v3= -v we will get a new solution [Q2]. Surprisingly the new solution $[Q2]$ (when v=1/5760) is generated by the equation

$$q_{k_1,k_2} = (-1)^{k_1} \sum_{l_1=0}^{(\lambda-1)(m_1+m_3)} \sum_{l_2=0}^{(\lambda-1)(m_2+m_3)} p_{l_1,l_2} N(k_1+1-l_1, k_2+1-l_2|\Xi\cup\Xi) \tag{5.9}$$

where $k_1 = 0,\cdots,(\lambda+1)(m_1+m_3)-2,\ \ k_2 = 0,\cdots,(\lambda+1)(m_2+m_3)-2,$ $m_1 = 1,\ m_2 = 1,\ m_3 = 2,\ \lambda = 2,\ \ p_{l_1,l_2}$ is as given in Equation 5.2,
and $N(\cdot|\Xi\cup\Xi)$ is a box spline with direction vectors $\xi^1 = (1,0)^T$, $\xi^2 = (0,1)^T$, $\xi^3 = (1,1)^T$ with multiplicities $2m_1, 2m_2$ and $2m_3$.
We recognize Equation 5.9 as an extended version of Equation 4.24 given in Chapter 4.

EXAMPLE 2 :- Consider the same box spline as in Example 1 and the four scale relation ($\lambda = 4$). Here we will create $[Q]_{106\times 1}$ using Equation 5.9 with $\lambda = 4$ and test whether $[Q]$ will satisfy $[H]_{8\times 106}[Q]_{106\times 1} = [0]$. It is shown in Appendix E that $[Q]$ satisfies $[H][Q] = [0]$ and the necessary condition for *wavelet decompositions*, namely $\sum_{k_1=0}^{13}\sum_{k_2=0}^{13} q_{k_1,k_2} = 0$. See Appendix E for MAPLE codes.

Now we return to the steps of our approach.

Step 7: Here we will give an approach for determining the decomposition sequences. As we did in the **Step 7** for the cardinal spline, we take the inner product of $\phi_\Xi(x_1,x_2)$ and the decomposition relation 4.13 (related to two scale relation) and then apply the inner product formula 3.24 and Equation 5.2. This gives

$$\sum_{r_1=0}^{n1}\sum_{r_2=0}^{n2} p_{r_1,r_2} N(r_1-l_1+n1, r_1-l_2+n2|\Xi\cup\Xi) = \sum_{j_1,j_2\in\mathcal{Z}^1} g_{2j_1-l_1,2j_2-l_2} N(j_1+n1, j_2+n2|\Xi\cup\Xi), \tag{5.10}$$

where $n1 = m_1+m_3,\ n2 = m_2+m_3$.
Similarly taking the inner product of $\psi_\Xi(x_1,x_2)$ and Equation 4.13 and then

applying the inner product formula 3.24 and Equation 5.3 gives

$$2\sum_{k_1,k_2=0,0}^{s1,s2} q_{k_1,k_2} N(u_1,v_1|\Xi\cup\Xi) = \sum_{j_1,j_2\in\mathcal{Z}^1}\sum_{k_1,k_2=0,0}^{s1,s2}\sum_{r_1,r_2=0}^{s1,s2} q_{k_1,k_2}q_{r_1,r_2}h_{2j_1-l_1,2j_2-l_2}N(u_2,v_2|\Xi\cup\Xi), \tag{5.11}$$

where
$s1 = 3(m_1+m_3)-2,\ s2 = 3(m_2+m_3)-2,$
$u_1 = k_1 - l_1 + m_1 + m_3,\ v_1 = k_2 - l_2 + m_2 + m_3,$
$u_2 = j_1 + r_1 - k_1,\ v_2 = j_2 + r_2 - k_2.$
Equations 5.10 and 5.11 produce linear equations to determine g_{k_1,k_2} and h_{k_1,k_2} by varying l_1, l_2. The number of equations and the number of unknowns depend on truncation of the infinite decomposition sequences. As mentioned in Section 4.3, the truncation will cause some discrepency. For more details, refer the text by C. Chui [6].

We have presented all the steps of our approach for the bivariate three direction box splines. We observed that we always produce fewer equations than the number of q_{k_1,k_2} for all cases of λ. We can supply some additional equations by imposing a condition of symmetry (or antisymmetry) on q_{k_1,k_2} and then solve the associated homogeneous system of equations $[H][Q]$ by assigning appropriate values for parameters. The support of q_{k_1,k_2} is a hexagonal subset of the rectangular grid and q_{k_1,k_2} is symmetric about the center of its support. However for all cases of λ, the q_{k_1,k_2} are generated by

$$q_{k_1,k_2} = (-1)^{k_1}\sum_{l_1=0}^{(\lambda-1)(m_1+m_3)}\ \sum_{l_2=0}^{(\lambda-1)(m_2+m_3)} p_{l_1,l_2}N(k_1+1-l_1,k_2+1-l_2|\Xi\cup\Xi), \tag{5.12}$$

where $k_1 = 0,\cdots,(\lambda+1)(m_1+m_3)-2,\ \ k_2 = 0,\cdots,(\lambda+1)(m_2+m_3)-2)$, the values of p_{l_1,l_2} are as given in Equation 5.2 and $N(\cdot|\Xi\cup\Xi)$ is a box spline with direction vectors $\xi^1=(1,0)^T, \xi^2=(0,1)^T, \xi^3=(1,1)^T$ with multiplicities $2m_1, 2m_2$ and $2m_3$.

We conclude that the construction of the bivariate three direction box splines wavelets would be implemented by performing following steps:

- Choose $W_0 \perp V_0$.
- Select scale λ and multiplicites m_1, m_2, m_3 direction vectors $\xi^1 = (1,0)^T$, $\xi^2 = (0,1)^T$, $\xi^3 = (1,1)^T$.
- Create matrix $[\tilde{H}]$ from Equation 5.7 by increasing l_1 from l_1^{min} to l_1^{max} and increasing l_2 from l_2^{min} to l_2^{max} and from the symmetry or antisymmetry conditions. l_1^{min}, l_1^{max}, l_2^{min}, l_2^{max} are given by Theorem 5.1.
- Solve the homogeneous system of equations $[\tilde{H}][Q] = 0$ to determine the reconstruction sequences $\{q_k\}$.

5.2 Analysis of The Geometric Representation of Matrix $[B]$

We analyze the geometric representation shown in Figure 5.2 and Figure 5.3.

- Figure 5.2(a): The range of $\theta_{r_1}, \theta_{r_2}$ for $p_{\theta_{r_1},\theta_{r_2}} > 0$, $\theta_{r_1}, \theta_{r_2} \in \mathcal{Z}$. ($\theta_{r_1}, \theta_{r_2}$ − Range).
  It was shown by Daehlen and Lyche [14] that the range of $\theta_{r_1}, \theta_{r_2}$ is a hexagonal subset of rectangular integer grid $[0,(\lambda-1)(m_1+m_3] \times [0,(\lambda-1)(m_2+m_3]$. This gives Result 1 in Section 5.1.
- Figure 5.2(b): The range of i, j such that $N(i,j|\Xi \cup \Xi) > 0$, $i,j \in \mathcal{Z}$. (N-Support)
  This is a hexagonal subset of the rectangular integer grid $[0, 2(m_1 + m_3)] \times [0, 2(m_2+m_3)]$. $N(i,j|\Xi \cup \Xi)$ can only be nonzero if (i,j) belongs to the interior of the hexagonal support. Therefore $N(i,j|\Xi \cup \Xi)$ is zero along the boundaries of the hexagon $OJKLMN$ and nonzero within the hexagon $QRSTUV$ (including the boundaries).
- Figure 5.3(a): A P-Window is a just a hexagon with same dimension as the range of $\theta_{r_1}, \theta_{r_2}$. Its lowest and highest vertices are denoted by $W1$ and $W4$.
  We have drawn a line $W1W7$ in the direction $(1,1)^T$ to facilitate our analysis.

- Figure 5.3(b): To compute the $b_{\theta_{k_1},\theta_{k_2}}$,
  1. Place the P-Window over a rectangular integer grid such that $W1 \equiv\ i = -\theta_{k_1} + 2(m_1 + m_3) - 1, j = -\theta_{k_2} + 2(m_2 + m_3) - 1)$.
  2. Find $S\ \equiv$ Overlapping region between P-Window and N-Support.
  3. Compute $b_{\theta_{k_1},\theta_{k_2}} = \sum N(i_1, j_1|\Xi \cup \Xi)p_{i_2,j_2}$ for all $i_1, i_2, j_1, j_2 \in S \bigcap \mathcal{Z}$. where the point (i_2, j_2) in the P-window lies exactly over the point (i_1, j_1) in N-Support within S.
  4. Compute $[B] = [b_{0,0}, \cdots, b_{(\lambda+1)(m_1+m_3)-2,(\lambda+1)(m_2+m_3)-2}]$ by placing the P-Window at integer points within rectangular integer grid $WXYZ \equiv [-(\lambda-1)(m_1+m_3)+1, -(\lambda-1)(m_2+m_3)+1] \times [2(m_1+m_3)-1, 2(m_2+m_3)-1]$. For example $b_{0,0}$ and $b_{(\lambda+1)(m_1+m_3)-2,(\lambda+1)(m_2+m_3)-2}$ are computed by placing the P-Window such that $W1 \equiv Z$ and $W1 \equiv X$ respectively.

If there is no overlap between the P-Window and the N-Support when we place the P-Window at integer points within rectangular integer grid $WXYZ$ then $b_{\theta_{k_1},\theta_{k_2}} = 0$. The P-Window moves down ($\downarrow$) while θ_{k_2} increases from zero and θ_{k_1} remains constant and to left ($\leftarrow$) while θ_{k_1} increases from zero and θ_{k_2} remains constant. This type of movement is inconvenient for finding the overlap. We can easily determine the overlap by moving the P-Window in the direction $(-1, -1)^T$ ($\swarrow$) within $WXYZ$ while θ_{k_1} and θ_{k_2} increase.

Let us analyze the Figure 5.4 .

- $WXYZ$ is split into two regions by drawing the line ZB in the direction $(1, 1)^T$. The region R_I is below the line ZB and the region R_{II} is above the line ZB.

- Consider the movement of the P-Window along the line ZB. Place the P-Window at position P1 ($Z \equiv W1$), then drag the P-Window such that $W1W7$ moves along the line ZB and stop dragging when $W1$ reaches $B(W1 \equiv B)$. When $W1$ reaches B, the edge $W3W4$ of the P-Window lies over the edge QV. Therefore, if we drag the P-Window one point further along ZB, the edge $W3W4$ lies over the edge ON. This overlapping situation does not interest us because $N(i, j|\Xi \cup \Xi)$ is zero along the boundaries of the hexagon $OJKLMN$. We are only interested in the overlap between the P-Window and the interior hexagon $QRSTUV$ of N-Support.

- In the region R_I, the P-Window moves along the lines that start from points in the line ZY, end at points in the line WX and are parallel to the line ZB until the P-Window comes to the position P2 since $W4W5$ lies over QR. After passing the line XE (corresponding to P2), P-Window moves along the lines that start from points in the line ZY, end at points in the line XY and are parallel to the line ZB. It is clear from the position P3 ($W6W5$ lies over RS) that there is no overlap below the line CD.

- In the region R_{II}, the P-Window moves along the lines that start from points in the line ZW, end at points in the line WX and are parallel to the line ZB. It is clear from the position P4 ($W2W3$ lies over UV) that there is no overlap above the line AF.

From the analysis of Figure 5.4, we conclude that S ($\equiv$ Overlapping region between the P-Window and the N-Support) is a hexagon $XCDZFA$. A simple calculation gives the dimension of the $XCDZFA$
$AX = ZD = ZS + SD = 2m_1 - 1 + (\lambda - 1)m_1 = (\lambda + 1)m_1 - 1.$
$XC = ZF = ZU + UF = 2m_2 - 1 + (\lambda - 1)m_2 = (\lambda + 1)m_2 - 1.$
$WF = WA = CY = YD = ZY - ZD = (\lambda+1)(m_2+m_3)-2-((\lambda+1)m_2-1) = (\lambda + 1)m_3 - 1$
Since $i = -\theta_{k_1} + 2(m_1 + m_3) - 1, j = -\theta_{k_2} + 2(m_2 + m_3) - 1)$, the index $(\theta_{k_1}, \theta_{k_2})$ is just a integer translation of the hexagon $XCDZFA$ (see Figure 5.1(b)). This gives Result 2 and Results 4 in Section 5.1. A similar analysis can be done for Figure 5.5. Both analyses give the same results.

It is worth noting that since the support of $[B]$ is a geometric convolutions of support of the sequence $\{p\}$ ($\equiv [P]$) and the support of the box spline $N(\cdot|\Xi \bigcup \Xi)$ ($\equiv [A]$) the same results can be derived directly by using *Minkowski Sum*. The algorithm for computing this sum is given by A.E. Middleditch [30].

5.2.1 Derivation of Permutation Matrices $[X]$ and $[Y]$

If we find a transformation T such that it transforms the points from a hexagonal subset of the rectangular integer grid to linearly ordered points in the integer line, then we can determine the permutation matrices $[X]$, $[Y]$ in Equation 5.8. Consider the Figure 5.6 and following steps for finding the T.

- First assume that point H is a grid point of the rectangular integer grid $OQSU$, and transformation F transforms H to an integer point h in the integer line. h= $F(x, y)$. If F transforms

$$(0,0),(0,1),(0,2)\cdots \longrightarrow 0,1,2\cdots$$

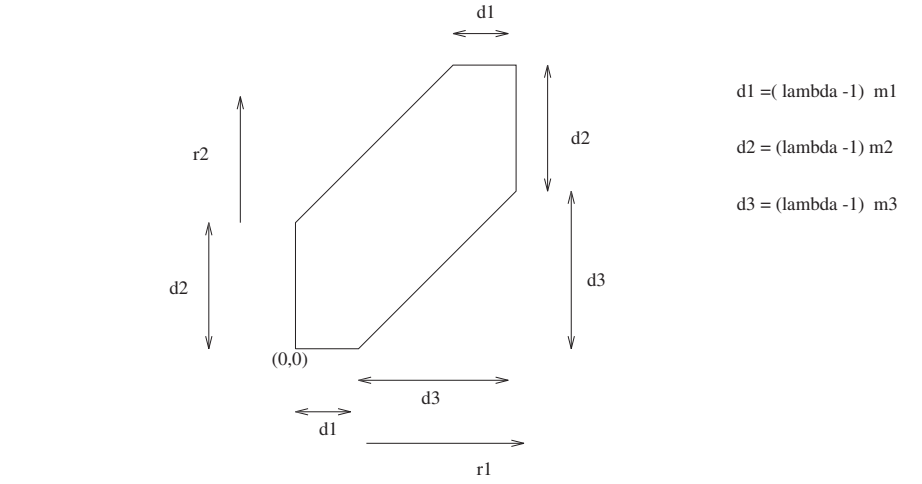
  then $F(x, y) = x(s1 + s3 + 1) + y$.

- Remove N_p (= number of the points in the region $APQB$) from F. Since the number of points in the line AB is in arithmetic series when x increases, N_p is calculated easily. N_p increases as x increases from 0 to $s3 - 1$ and remains constant when $x > s3 - 1$.

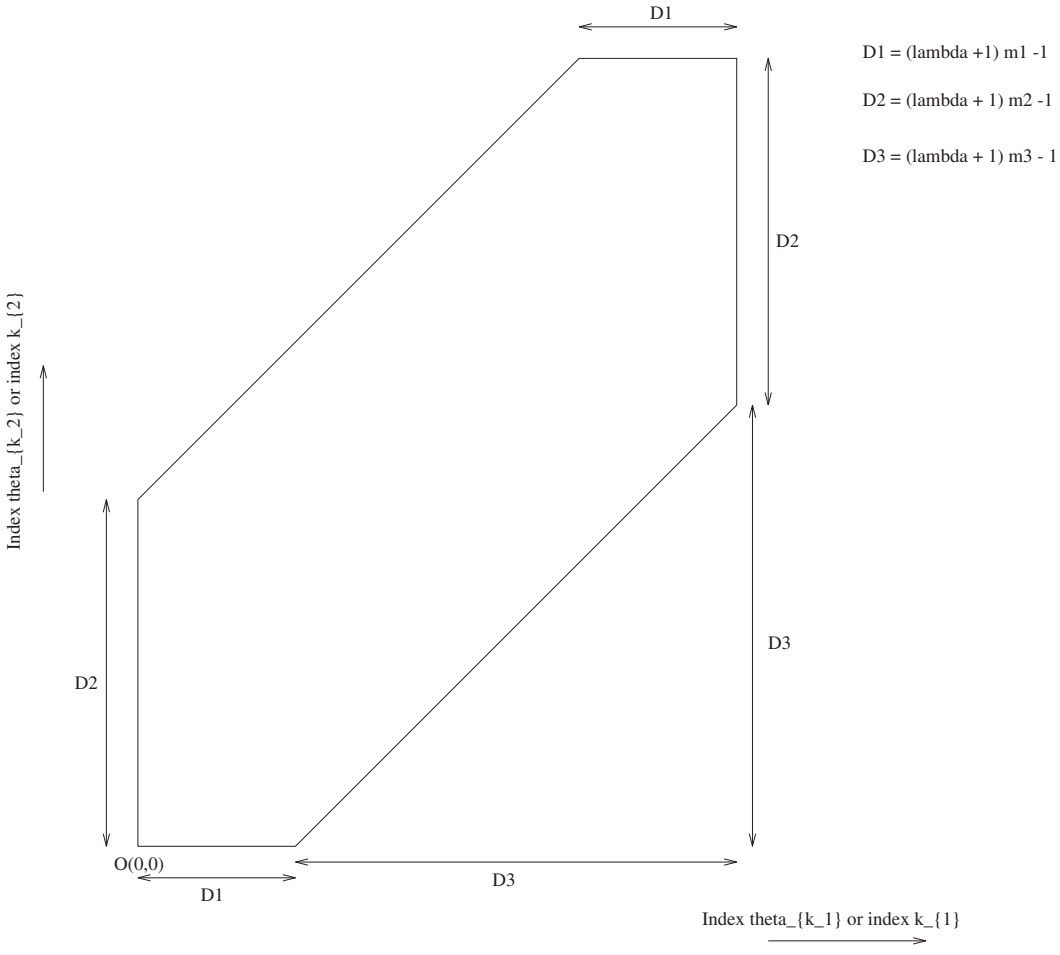
- Remove also M_p (= number of the points in the region CVD) from F. Since the number of points in the line CD is in arithmetic series when x increases, M_p is calculated easily. M_p increases as x increases from $s3 + 1$ to $s3 + s2$ and remains constant when $x < s3 + 1$.

- From the above observations, we conclude that $T = F(x, y) - N_p - M_p$, where $N_p = 1/2N1(2s3 - N1 + 1)$, $M_p = 1/2N2(N2 + 1)$, $N1 = min(x, s3 - 1)$, $N2 = max(x - s3, 0)$.

- From this transformation T, we can easily deduce Result 3 in the Section 5.1

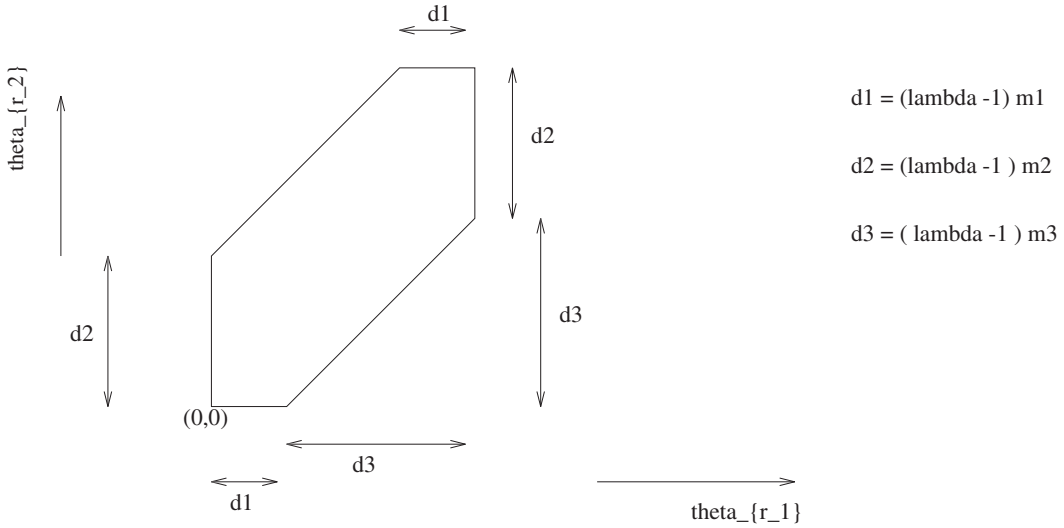


(a) Index r_{1}, r_{2}

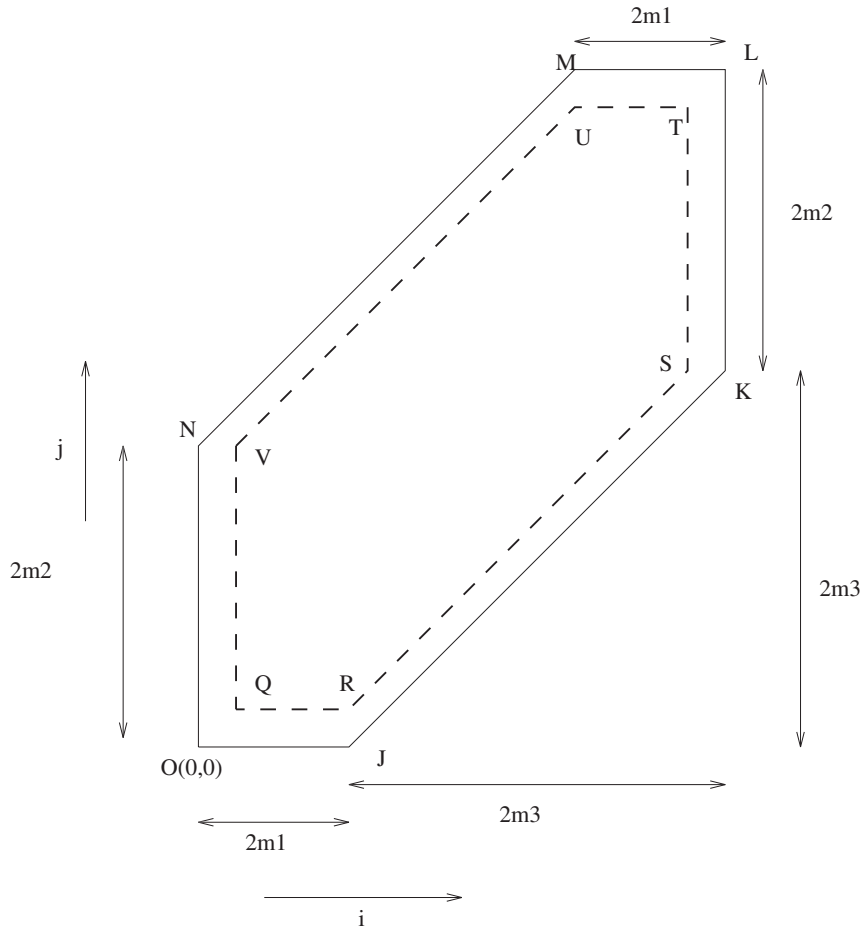


(b) Index theta_{k_1}, theta_{k_2} or index k_{1}, k_{2}

Figure 5.1: Indices of p_{r_1,r_2}, q_{k_1,k_2} and $b_{\theta_{k_1},\theta_{k_2}}$



(a) theta_{r_1}, theta_{r_2} - Range



(b) N - Support

Figure 5.2: The supports of $p_{\theta_{r_1},\theta_{r_2}}$ and $N(i,j \mid \Xi \bigcup \Xi)$

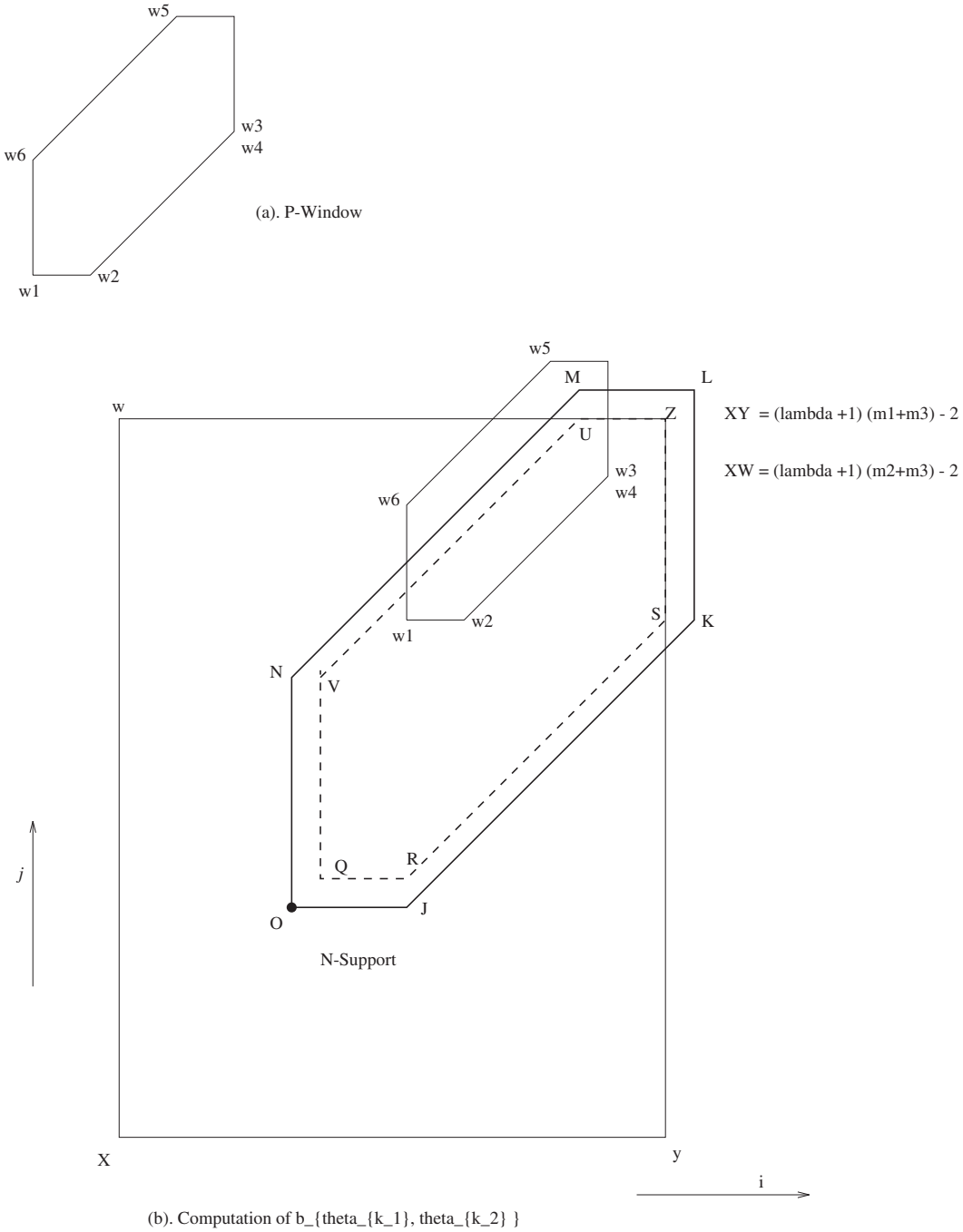


Figure 5.3: The geometric representation for computation of $b_{\theta_{k_1},\theta_{k_2}}$

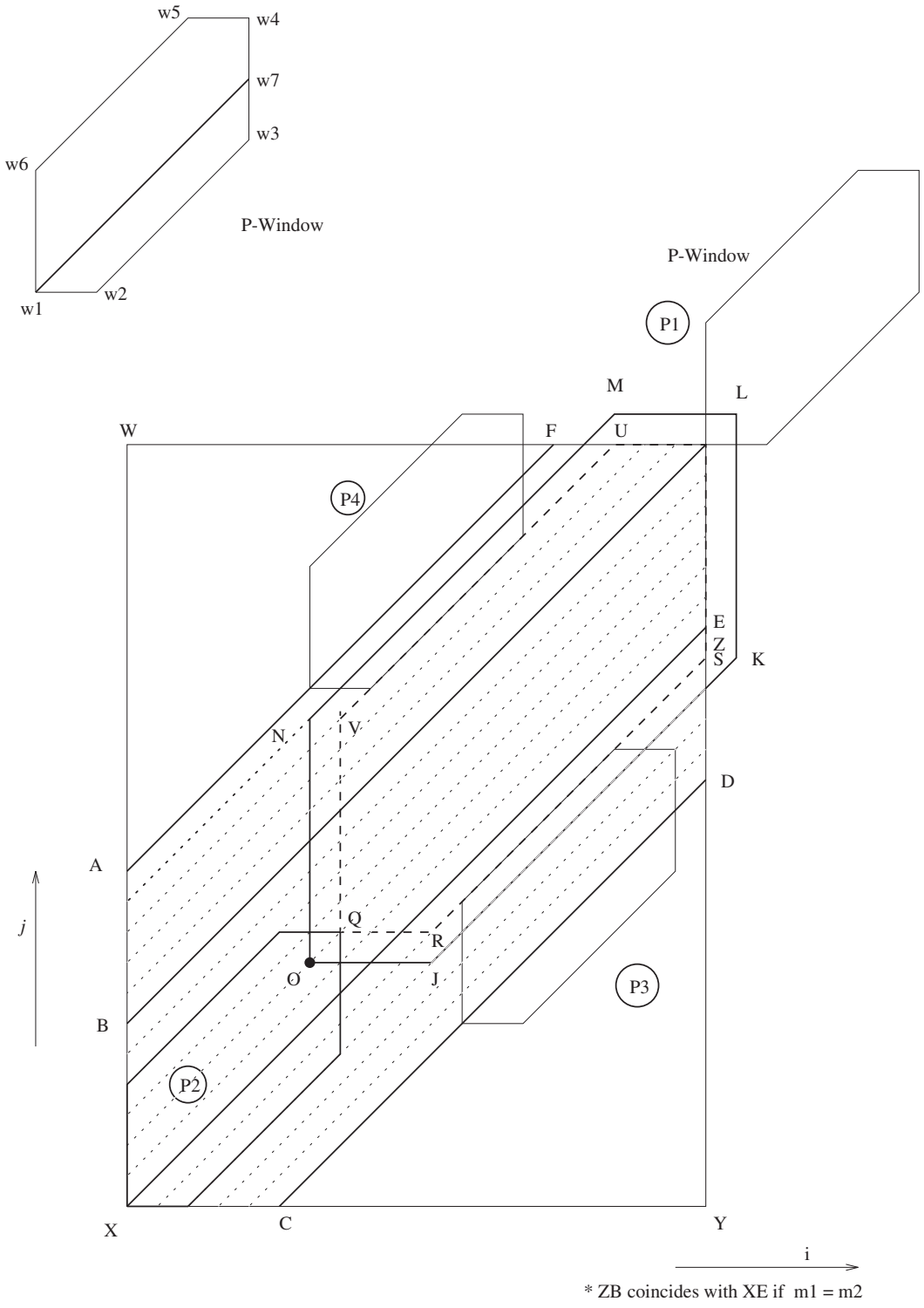


Figure 5.4: The geometric representation for computation of [B]: case $m_1 < m_2$

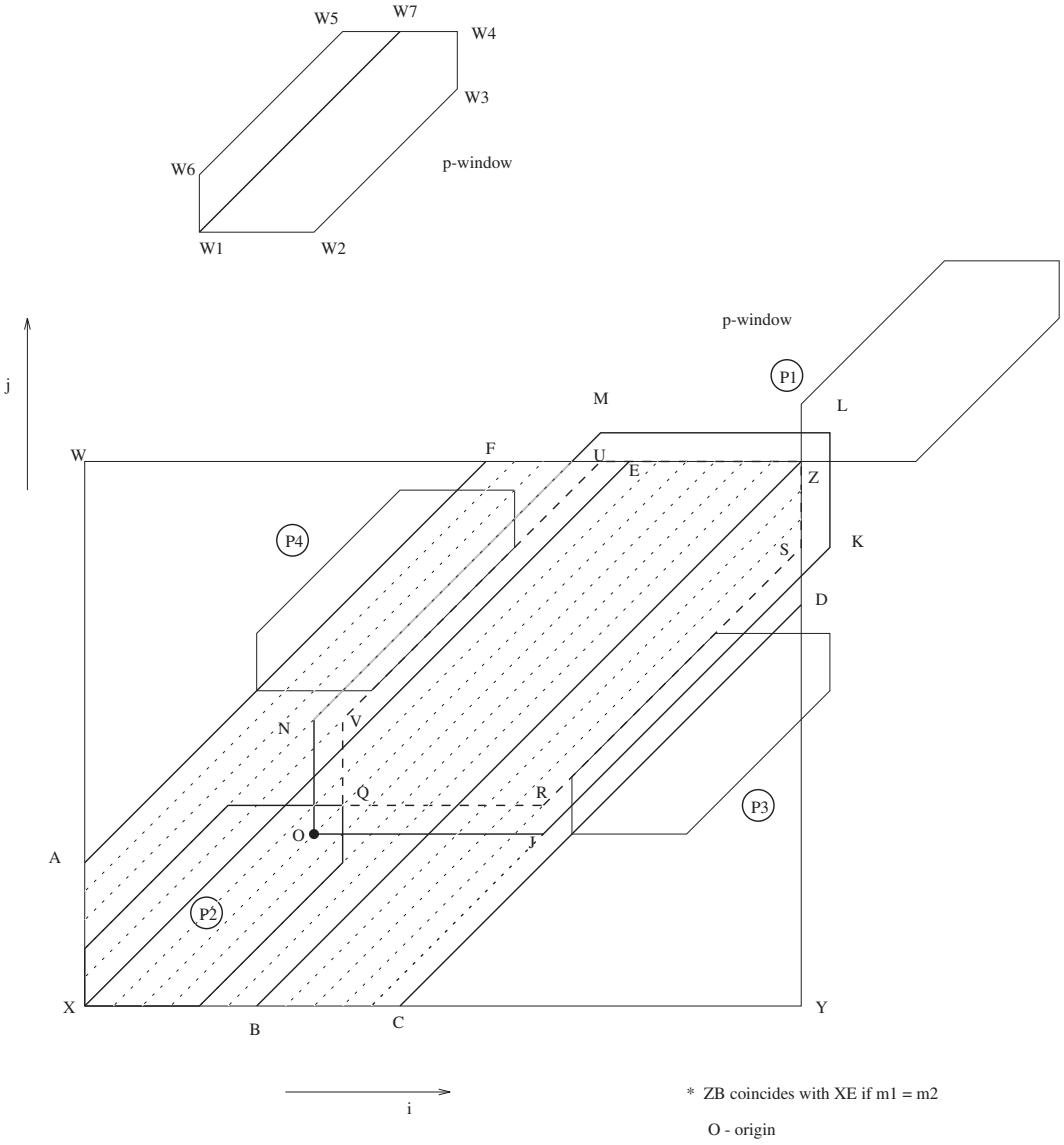


Figure 5.5: The geometric representation for computation of [B]: case $m_1 > m_2$

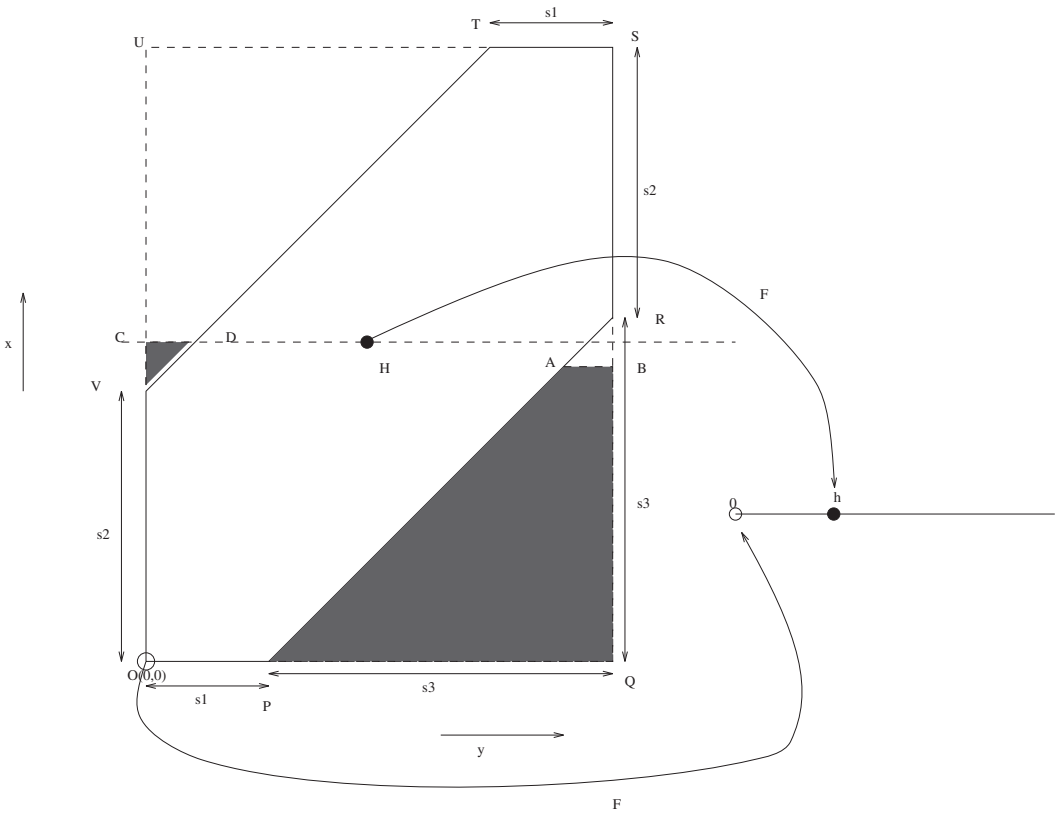


Figure 5.6: A useful transformation for determining the permutation matrices [X] and [Y]

Box Spline Wavelets For Two Scale Relation

Direction vectors $\xi^1 = (1,0)^T, \xi^2 = (0,1)^T, \xi^3 = (1,1)^T$ with multiplicities 1, 1, 2 are considered.

$$Q = \begin{pmatrix} q_{0,0} = 1 \\ q_{0,1} = 3 \\ q_{0,2} = 2 \\ q_{1,0} = -3 \\ q_{1,1} = -55 \\ q_{1,2} = -84 \\ q_{1,3} = -32 \\ q_{2,0} = 2 \\ q_{2,1} = 84 \\ q_{2,2} = 353 \\ q_{2,3} = 357 \\ q_{2,4} = 86 \\ q_{3,1} = -32 \\ q_{3,2} = -357 \\ q_{3,3} = -791 \\ q_{3,4} = -552 \\ q_{3,5} = -86 \\ q_{4,2} = 86 \\ q_{4,3} = 552 \\ q_{4,4} = 791 \\ q_{4,5} = 357 \\ q_{4,6} = 32 \\ q_{5,3} = -86 \\ q_{5,4} = -357 \\ q_{5,5} = -353 \\ q_{5,6} = -84 \\ q_{5,7} = -2 \\ q_{6,4} = 32 \\ q_{6,5} = 84 \\ q_{6,6} = 55 \\ q_{6,7} = 3 \\ q_{7,5} = -2 \\ q_{7,6} = -3 \\ q_{7,7} = -1 \end{pmatrix}$$

All other $q_{i,j} = 0$ Where $0 \leq i, j \leq 7$

Table 5.1: Result of Example 1

Box Spline Wavelets For Four Scale Relation

Direction vectors $\xi^1 = (1,0)^T, \xi^2 = (0,1)^T, \xi^3 = (1,1)^T$ with multiplicities 1, 1, 2 are considered.

$$
Q = \left[\begin{array}{lcl}
q_{0,0} &=& 1 \\ q_{0,1} &=& 3 \\ q_{0,2} &=& 3 \\ q_{0,3} &=& 3 \\ q_{0,4} &=& 2 \\
q_{1,0} &=& -3 \\ q_{1,1} &=& -5 \\ q_{1,2} &=& -87 \\ q_{1,3} &=& -87 \\ q_{1,4} &=& -84 \\ q_{1,5} &=& -32 \\
q_{2,0} &=& 3 \\ q_{2,1} &=& 87 \\ q_{2,2} &=& 360 \\ q_{2,3} &=& 450 \\ q_{2,4} &=& 447 \\ q_{2,5} &=& 363 \\ q_{2,6} &=& 90 \\
q_{3,0} &=& -3 \\ q_{3,1} &=& -87 \\ q_{3,2} &=& -450 \\ q_{3,3} &=& -1020 \\ q_{3,4} &=& -1167 \\ q_{3,5} &=& -1083 \\ q_{3,6} &=& -720 \\ q_{3,7} &=& -150 \\
q_{4,0} &=& 2 \\ q_{4,1} &=& 84 \\ q_{4,2} &=& 447 \\ q_{4,3} &=& 1167 \\ q_{4,4} &=& 2033 \\ q_{4,5} &=& 2157 \\ q_{4,6} &=& 1794 \\ q_{4,7} &=& 1074 \\ q_{4,8} &=& 206 \\
 & \vdots &
\end{array}\right]
\left|\begin{array}{lcl}
 & \vdots & \\
q_{5,1} &=& -32 \\ q_{5,2} &=& -363 \\ q_{5,3} &=& -1083 \\ q_{5,4} &=& -2157 \\ q_{5,5} &=& -3191 \\ q_{5,6} &=& -3066 \\ q_{5,7} &=& -2346 \\ q_{5,8} &=& -1272 \\ q_{5,9} &=& -206 \\
q_{6,2} &=& 90 \\ q_{6,3} &=& 720 \\ q_{6,4} &=& 1794 \\ q_{6,5} &=& 3066 \\ q_{6,6} &=& 3900 \\ q_{6,7} &=& 3420 \\ q_{6,8} &=& 2346 \\ q_{6,9} &=& 1074 \\ q_{6,10} &=& 150 \\
q_{7,3} &=& -150 \\ q_{7,4} &=& -1074 \\ q_{7,5} &=& -2346 \\ q_{7,6} &=& -3420 \\ q_{7,7} &=& -3900 \\ q_{7,8} &=& -3066 \\ q_{7,9} &=& -1794 \\ q_{7,10} &=& -720 \\ q_{7,11} &=& -90 \\
q_{8,4} &=& 206 \\ q_{8,5} &=& 1272 \\ q_{8,6} &=& 2346 \\ q_{8,7} &=& 3066 \\ q_{8,8} &=& 3191 \\ q_{8,9} &=& 2157 \\ q_{8,10} &=& 1083 \\ q_{8,11} &=& 363 \\ q_{8,12} &=& 32 \\
 & \vdots &
\end{array}\right|
\left[\begin{array}{lcl}
 & \vdots & \\
q_{9,5} &=& -206 \\ q_{9,6} &=& -1074 \\ q_{9,7} &=& -1794 \\ q_{9,8} &=& -2157 \\ q_{9,9} &=& -2033 \\ q_{9,10} &=& -1167 \\ q_{9,11} &=& -447 \\ q_{9,12} &=& -84 \\ q_{9,13} &=& -2 \\
q_{10,6} &=& 150 \\ q_{10,7} &=& 720 \\ q_{10,8} &=& 1083 \\ q_{10,9} &=& 1167 \\ q_{10,10} &=& 1020 \\ q_{10,11} &=& 450 \\ q_{10,12} &=& 87 \\ q_{10,13} &=& 3 \\
q_{11,7} &=& -90 \\ q_{11,8} &=& -363 \\ q_{11,9} &=& -447 \\ q_{11,10} &=& -450 \\ q_{11,11} &=& -360 \\ q_{11,12} &=& -87 \\ q_{11,13} &=& -3 \\
q_{12,8} &=& 32 \\ q_{12,9} &=& 84 \\ q_{12,10} &=& 87 \\ q_{12,11} &=& 87 \\ q_{12,12} &=& 5 \\ q_{12,13} &=& 3 \\
q_{13,9} &=& -2 \\ q_{13,10} &=& -3 \\ q_{13,11} &=& -3 \\ q_{13,12} &=& -3 \\ q_{13,13} &=& -1
\end{array}\right]
$$

All other $q_{i,j} = 0$ Where $0 \leq i, j \leq 13$

Table 5.2: Result of Example 2

Chapter 6

Conclusion

Our Matrix-Nullspace method for constructing multivariate non-tensor-product wavelets that generate an orthogonal decomposition of $\mathcal{L}^2(\mathcal{R}^s),\ s \geq 1$, involves the use of an inner product, a matrix formulation, an associated homogeneous system of equations and the determination of a null space, and can be applied to yield explicit formulas for compactly supported spline-wavelets based on the multiresolution analysis of $\mathcal{L}^2(\mathcal{R}^s),\ 1 \leq s \leq 2$, generated by any box spline whose direction set constitutes a unimodular matrix. We have demonstrated the validity of our method by applying our approach to the univariate cardinal splines in Chapter 4 and to the bivariate three directional box splines in Chapter 5. In particular, when univariate cardinal B-splines and the two scale relation are considered, the minimally supported cardinal spline wavlets of Chui and Wang [11] are recovered.

Since the most important property of the wavelet, at least in many applications to data compression, is the property of generalized linear phase, a condition on the symmetry or antisymmetry of the wavelet can be imposed to yield symmetric or antisymmetric box spline wavelets. The reconstruction sequence $\{q_{k_1,k_2}\}$ is symmetric or antisymmetric about the center of its support. Further, we have observed that supports of the reconstruction sequences $\{q_{k_1,k_2}\}$ and $\{p_{k_1,k_2}\}$ have similar geometrical shapes. For example, for the univariate cardinal spline $\{q_k\}$ and $\{p_k\}$ are segments of the integer line and for bivariate box spline, $\{q_{k_1,k_2}\}$ and $\{p_{k_1,k_2}\}$ are hexagonal subsets of rectangular integer grids.

We tested our approach with two, four and eight scale relations. For all tested scales, the reconstruction sequences, $\{q\}$, are generated by Equation

4.24 for the cardinal spline and by Equation 5.12 for box splines. We strongly believe that Equations 4.24 and 5.12 can generate reconstruction sequences for all different scales.

At present, there are no special provisions for constructing the wavelets from the box spline with unimodular direction sets in $\mathcal{R}^3$, the orthonormal wavelets and polyhedral wavelets. We hope our approach can be extended to construction of box spline wavelets in $\mathcal{R}^3$. It may also be possible to construct orthonormal wavelets and polyhedral wavelets by modifying the notion of the multiresolution analysis.

Appendix A

Example 1 in Chapter 4

This appendix consists MAPLE files for driver, routine of creating vector $[B]$ and routine of creating matrix $[H]$ and ouputs for matricies $[H]$, $[Q]$ and necessary condition for wavlet decompositions. Here we consider fourth order cardinal spline under a two scale relation.

MAPLE code for driver file:

```
with(linalg);
# Read the routine for creating the vector [B]
read(‘create_vecB‘);

# Read the routine for creating the matrix [H] by using Equation 4.22.
read(‘create_mtxH‘);

# Read necessary input parameters for the routine create_vecB
# Input parameters are scale (lambda=2), order of the cardinal spline (m=4),
# support of the 2m order cardinal spline (Bindex),
# values of 2m order cardinal spline within its support (Bvalue)
# values of sequence p_{r}, given by Equation 4.16 (p).
read(‘allvarC1‘);

# Call the routine create_vecB
B:=create_vecB(lambda,m,Bindex,p,Bvalue);
```

```
# Call the routine create_mtxH
H:= create_mtxH(B,lambda,m);

# Display the matrix [H]
print(H);

# Solve the homogeneous system of equations [H][Q]=0
t:=rowdim(H);
b:=array(1..t,1..1,sparse);
Q:=linsolve(H,b,'r',v);

# Display the vector [Q] (reconstruction sequence)
print(Q);

# NECESSARY CONDITION FOR WAVELET DECOMPOSITIONS
sum:=0;
for i from 1 to rowdim(Q) do
sum:= sum + Q[i,1];
od;
```

MAPLE code for routine create_vecB:

```
create_vecB:=proc(lambda,m,Bindex,p, Bvalue)
# lambda = Scale
# m = Order of the cardinal spline

# Bindex = array(1..2m+1,1..1,sparse). We marked this array such that
#          Bindex[i+1,1]:=C if the value of 2m order cardinal spline (N_{2m})
#          is nonzero at an integer point i within its support, otherwise
#          Bindex[i+1,1]:=0, where i is the C^th nonzero integer point
#          within support of N_{2m}.

# Bvalue = The nonzero values of N_{2m} at integer points within its support,
#           i.e, Bvalue[C,1]= value of N_{2m} at the intger point j.

# p = array(1..(lambda-1)m+1,1..1,sparse). We assigned  such that
#     p[r+1,1]:= the value of sequence p_{r}. The sequence p_{r} is given by
#     Equation 4.16.
```

```
local i,al,c1,t1,t2,ak,ar,col_ele,count_ele,x,y,index,N,Lk,Lr,Uk,Ur;

N:=(lambda+1)*m-1  # Size of vector [B]
c1:=array(1..1,1..N,sparse);
t1:=rowdim(Bindex);
t2:=coldim(Bindex);
col_ele:=0;
count_ele:=0;

# It follows from Step 4 in Sectiin 4.3 that vector [B] is independant of l.
# Therefore select an arbitary value for l.
al:=m-1; # Value of l

# Indices k and r in Equation 4.20.
Lk:=lambda*al-m+1;                # Lower bound of index k of q_{k}
Uk:=lambda*al+(lambda*m)-1;       # Upper bound of index k of q_{k}
Lr:=lambda*al;                    # Lower bound of index r of p_{r}
Ur:=lambda*al+(lambda-1)*m        # Upper bound of index r of p_{r}

# Create vector [B] by using Equations 4.20 and 4.21.
for ak from Lk by 1 to Uk do
for ar from Lr by 1 to Ur do
x:= ar-ak+m+1; y:=br-bk+m+1;
if((x>=1 and x<=t1) and (y>=1 and y<=t2)) then
index:=Bindex[x,y];
if(index>0) then
col_ele:=col_ele+ Bvalue[index,1]*p[ar-lambda*al+1,1];
fi;
fi;
od;

count_ele:=count_ele+1;
c1[1,count_ele]:=col_ele;
col_ele:=0;
od;
```

```
c1;
end;
```

MAPLE code for routine create_mtxH:

```
create_mtxH:=proc(B,lambda,m)
# B = Vector [B] in Equation 4.22
# lambda = Scale
# m = Order of the cardinal spline
local i,al,mtx1,,Q,ak,col_ele,count_ele,row,index1,t1,t,r,v,alL,alU;

r:=lambda+1;
t1:=r*m-2; # Largest index of q_{k}
t:=t1+1;   # Size of vector [B]
alL:=trunc((1-lambda*m)/lambda);      # Lower bound of l
alU:= trunc(((lambda+2)*m-3)/lambda); # Upper bound of l
v:=alU-alL+1;  # Number of equations produced from Equation 4.22 by varying l

col_ele:=0;
count_ele:=0;
row:=0;
mtx1:=array(1..v,1..t,sparse);
Q:=array(0..t1,1..1,sparse);

# Mark the nonzero q_{k}. It follows from Step 4 in Section 4.3 that there
# is no q_{k}=0 when 0 =< k =< (lambda+1)m-1. Therefore mark the nonzero q_{k}
by its count within range of k .
for i from 0 to t1 do
count_ele:= count_ele +1;
Q[i,1]:=count_ele;
od;

# Produce equations from Equation 4.22 by varying l within its bounds.
for al from alL to alU do
row:=row+1;
for ak from (lambda*al-m+1) by 1 to (lambda*m+(lambda*al)-1) do
col_ele:=col_ele+1;
if((ak>=0 and ak<=t1) then
```

```
index1:=Q[ak,bk];
mtx1[row,index1]:= B[1,col_ele];
fi;
od;
od;
mtx1;
end;
```

OUTPUTS:

```
                                    [H] :=

                    31
                 [-----, 1/80640, 0, 0, 0, 0, 0, 0, 0, 0, 0]
                  20160

                 247   559     31
               [----, -----, -----, 1/80640, 0, 0, 0, 0, 0, 0, 0]
                2520  26880  20160

             337   9241   247   559     31
           [----, -----, ----, -----, -----, 1/80640, 0, 0, 0, 0, 0]
            1120  40320  2520  26880  20160

          247   9241   337   9241   247   559     31
        [----, -----, ----, -----, ----, -----, -----, 1/80640, 0, 0, 0]
         2520  40320  1120  40320  2520  26880  20160

     31    559    247   9241   337   9241   247   559     31
  [-----, -----, ----, -----, ----, -----, ----, -----, -----, 1/80640, 0]
   20160  26880  2520  40320  1120  40320  2520  26880  20160

                  31    559    247   9241   337   9241   247   559     31
  [0, 1/80640, -----, -----, ----, -----, ----, -----, ----, -----, -----]
                20160  26880  2520  40320  1120  40320  2520  26880  20160

                            31    559    247   9241   337   9241   247
```

```
[0, 0, 0, 1/80640, -----, -----, ----, -----, ----, -----, ----]
                   20160  26880  2520  40320  1120  40320  2520

                              31    559    247   9241   337
    [0, 0, 0, 0, 0, 1/80640, -----, -----, ----, -----, ----]
                             20160  26880  2520  40320  1120

                                       31    559    247
        [0, 0, 0, 0, 0, 0, 0, 1/80640, -----, -----, ----]
                                      20160  26880  2520

                                             31
           [0, 0, 0, 0, 0, 0, 0, 0, 0, 1/80640, -----]
                                            20160

                      [     _t[1]     ]
                      [               ]
                      [  - 124 _t[1]  ]
                      [               ]
                      [   1677 _t[1]  ]
                      [               ]
                      [  - 7904 _t[1] ]
                      [               ]
                      [  18482 _t[1]  ]
                      [               ]
               [Q] := [ - 24264 _t[1] ]
                      [               ]
                      [  18482 _t[1]  ]
                      [               ]
                      [  - 7904 _t[1] ]
                      [               ]
                      [   1677 _t[1]  ]
                      [               ]
                      [  - 124 _t[1]  ]
                      [               ]
                      [     _t[1]     ]
```

```
NECESSARY CONDITION FOR WAVELET DECOMPOSITION:

sum := 0

sum := _t[1]

sum := - 123 _t[1]

sum := 1554 _t[1]

sum := - 6350 _t[1]

sum := 12132 _t[1]

sum := - 12132 _t[1]

sum := 6350 _t[1]

sum := - 1554 _t[1]

sum := 123 _t[1]

sum := - _t[1]

sum := 0
```

Appendix B

Example 2 in Chapter 4

This appendix consists MAPLE files for driver, routine of creating matrix $[H1]$ and ouputs only for matricies $[Q1]$, $[Q2]$ and rank of matrix $[H1]$ $(=r)$. Here we consider fourth order cardinal spline under a four scale relation.

MAPLE code for driver file:

```
with(linalg);
# Read the routine for creating the vector [B]
read(‘create_vecB‘);

# Read the routine for creating the matrix [H] by using Equation 4.22.
read(‘create_mtxH‘);

# Read the routine for creating the matrix [H1] by imposing a symmetry
# or antisymmetry condition.
read(‘Symmetry_create_mtxH1‘);

# Read necessary input parameters for the routine create_vecB
# Input parameters are scale (lambda=4), order of the cardinal spline (m=4),
# support of the 2m order cardinal spline (Bindex),
# values of 2m order cardinal spline within its support (Bvalue)
# values of sequence p_{r}, given by Equation 4.16 (p).
read(‘allvarC2‘);
```

```
# Call the routine create_vecB
B:=create_vecB(lambda,m,Bindex,p,Bvalue);

# Call the routine create_mtxH
H:= create_mtxH(B,lambda,m);

# Solve the homogeneous system of equations [H][Q]=0
t:=rowdim(H);
b:=array(1..t,1..1,sparse);
Q:=linsolve(H,b,'r',v);

# NECESSARY CONDITION FOR WAVELET DECOMPOSITIONS
sum:=0;
for i from 1 to rowdim(Q) do
sum:= sum + Q[i,1];
od;

# Call the routine Symmetry_create_mtxH1. There is no need for calling this
# if there are enough equations.
H1:=Symmetry_create_mtxH1(H,lambda,m);

# Display the matrix [H1]
print(H1);

# Check the rank of matrix [H1] (= r)
r:=rank(H1);

# Solve the homogeneous system of equations [H1][Q1]=0
t1:=rowdim(H1);
b1:=array(1..t1,1..1,sparse);
Q1:=linsolve(H1,b1,'r',v);

# Display the vector [Q1] (reconstruction sequence)
print(Q1);
```

MAPLE code for routine create_mtxH1:

```
Symmetry_create_mtxH1:=proc(H,lambda,m)
```

```
# H = matrix [H], created by usinf the routine create_mtxH
# lambda = Scale
# m = order of the cardinal spline
local i,j,col_ele,D,d,t,v,;

D:=1;
if (lambda=2) then
D=0;
fi;
v:=rowdim(H);
t:=coldim(H);

# Additional equations from symmetry or antisymmetry condition. No need for
# additional equations if lambda=2.
d:=D*trunc(((lambda+1)*m -1)/2) # Number of additional equations
mtx1:=array(1..v+d,1..t,sparse);

# Copy the rows of matrix [H] to matrix [mtx1]
for i from 1 to v do
for j from 1 to t do
mtx1[i,j]:=H[i,j];
od;
od;

# Get additional equations by imposing a condition of symmetry or antisymmetry
# on q_{k}. No need for additional equations if lambda =2.
col_ele := 0;
if ( lambda > 2 ) then
for i from v+1 to v+d do
col_ele:=col_ele+1;
mtx1[i,col_ele]:=1;
mtx1[i,t-col_ele+1]:= (-1)^(t);
od;
fi;
mtx1;
end;
```

MAPLE codes for routines create_vecB and create_mtxH are same codes as in Appendix A.

OUTPUTS:

```
                                r := 16

                [                        _t[3]                        ]
                [                                                     ]
                [                    - 124 _t[3]                      ]
                [                                                     ]
                [                        _t[2]                        ]
                [                                                     ]
                [              200044 _t[3] - 124 _t[2]               ]
                [                                                     ]
                [                        _t[1]                        ]
                [                                                     ]
                [    - 124 _t[1] - 333206284 _t[3] + 200044 _t[2]     ]
                [                                                     ]
                [    1677 _t[1] + 4654303910 _t[3] - 2793846 _t[2]    ]
                [                                                     ]
                [  - 22041675916 _t[3] + 13230620 _t[2] - 7904 _t[1] ]
                [                                                     ]
                [  51602075289 _t[3] - 30974155 _t[2] + 18482 _t[1]  ]
                [                                                     ]
        [ Q1] := [ - 67763393840 _t[3] + 40674920 _t[2] - 24264 _t[1] ]
                [                                                     ]
                [  51602075289 _t[3] - 30974155 _t[2] + 18482 _t[1]  ]
                [                                                     ]
                [  - 22041675916 _t[3] + 13230620 _t[2] - 7904 _t[1] ]
                [                                                     ]
                [    1677 _t[1] + 4654303910 _t[3] - 2793846 _t[2]    ]
                [                                                     ]
```

```
[    - 124 _t[1] - 333206284 _t[3] + 200044 _t[2]    ]
[                                                    ]
[                        _t[1]                       ]
[                                                    ]
[             200044 _t[3] - 124 _t[2]               ]
[                                                    ]
[                        _t[2]                       ]
[                                                    ]
[                     - 124 _t[3]                    ]
[                                                    ]
[                        _t[3]                       ]
```

```
> Q2:=subs( _t[1]=25196*v, _t[2]=1681*v, _t[3]=v, op(Q1));
```

```
                                  [      v     ]
                                  [            ]
                                  [   - 124 v  ]
                                  [            ]
                                  [   1681 v   ]
                                  [            ]
                                  [  - 8400 v  ]
                                  [            ]
                                  [   25196 v  ]
                                  [            ]
                                  [  - 56624 v ]
                                  [            ]
                                  [  102476 v  ]
                                  [            ]
                                  [ - 152880 v ]
                                  [            ]
                                  [  193206 v  ]
                                  [            ]
                            Q2 := [ - 209064 v ]
                                  [            ]
                                  [  193206 v  ]
```

```
[           ]
[ - 152880 v ]
[           ]
[  102476 v  ]
[           ]
[  - 56624 v ]
[           ]
[   25196 v  ]
[           ]
[  - 8400 v  ]
[           ]
[   1681 v   ]
[           ]
[   - 124 v  ]
[           ]
[      v     ]
```

Appendix C

Example 3 in Chapter 4

This appendix consists MAPLE files for driver and routine of creating vector $[Q]$. Here we consider fourth order cardinal spline under a eight scale relation.

MAPLE code for driver file:

```
with(linalg);
# Read the routine for creating the vector [B]
read(‘create_vecB‘);

# Read the routine for creating the matrix [H]
read(‘create_mtxH‘);

# Read the routine for creating the vector [Q]
read(‘create_vecQ‘);

# Read necessary input parameters for the routine create_vecB
# Input parameters are scale (lambda=8), order of the cardinal spline (m=4),
# support of the 2m order cardinal spline (Bindex),
# values of 2m order cardinal spline within its support (Bvalue)
# values of sequence p_{r}, given by equation 4.16 (p).
read(‘allvarC3);

# Call the the routine create_vecB
B:=create_vecB(lambda,m,Bindex,p,Bvalue);
```

```
# Call the the routine create_mtxH
H:= create_mtxH(B,lambda,m);

# Call the the routine create_vecQ
Q:=create_vecQ(lambda,m,Bindex,p,Bvalue);

# NECESSARY CONDITION FOR WAVELET DECOMPOSITIONS
sum:=0;
for i from 1 to rowdim(Q) do
sum1:= sum1 + Q[i,1];
od;

# TEST [H][Q]=0
z:=multiply(H,Q);
```

MAPLE code for routine create_vecQ

```
create_Q:= proc(lambda,m,Bindex,p,Bvalue)
# lambda = Scale
# m = Order of the cardinal spline

# Bindex = array(1..2m+1,1..1,sparse). We marked this array such that
#          Bindex[i+1,1]:=C if the value of 2m order cardinal spline (N_{2m})
#          is nonzero at an integer point i within its support, otherwise
#          Bindex[i+1,1]:=0, where i is the C^th nonzero integer point
#          within support of N_{2m}.

# Bvalue = The nonzero values of N_{2m} at integer points within its support,
#           i.e, Bvalue[C,1]= value of N_{2m} at the intger point j.

# p = array(1..(lambda-1)m+1,1..1,sparse). We assigned  such that
#     p[r+1,1]:= the value of sequence p_{r}. The sequence p_{r} is given by
#     Equation 4.16.

local i,j,al,c1,t1,t2,t,ak,col_ele,count_ele,x,index1,index2,row,r,Nu,Q;
Nu:=(lambda+1)*m-2;     # Largest index k of q_{k}
# The number of nonzero q_{k}
```

```
t:=Nu+1;

c1 := array(1 .. t,1 .. 1,sparse);
Q:=array(0..Nu,1..1,sparse);
t1 := rowdim(Bindex);
t2 := coldim(Bindex);
count_ele:=0;
col_ele:= 0;
row:= 0;

# Mark the nonzero q_{k}. It follows from Step 4 in Section 4.3 that there
# is no q_{k}=0 when 0 =< k =< (lambda+1)m-1. Therefore mark the nonzero q_{k}
by its count within range of k .
for i from 0 to Nu do
count_ele:= count_ele +1;
Q[i,j]:=count_ele;
od;

# Create q_{k} by using Equation 4.24
for ak from 0 to Nu do
row:=row+1;
for al from 0 to (lambda-1)*m do
x := ak-al+2;
if (1 <= x) and (x <= t1)  then
index2 := Bindex[x,1];
if (0 < index2)  then
col_ele := (col_ele+Bvalue[index2,1]*p[al+1,1]);
fi;
fi;
od;
c1[row,1] := col_ele*((-1)^ak);
col_ele := 0;
od;
c1;
end;
```

MAPLE code for routine create_mtxH is same codes as in Appendix A.

Appendix D

Example 1 in Chapter 5

This appendix consists MAPLE files for driver, routines of creating matrix $[H]$, $[H1]$ and vector $[B]$ and ouputs only for matricies $[Q1]$, $[Q2]$ and rank of matrix $[H1]$ $(=r)$. Here we consider the box spline with direction vectors $\xi^1 = (1,0)^T$, $\xi^2 = (0,1)^T$, $\xi^3 = (1,1)^T$ under the two scale relation. The multiplicities of direction vectors are 1, 1 and 2.

MAPLE code for driver file:

```
with(linalg);
# Read the routine for creating the vector [B]
read(‘create_vecB‘);

# Read the routine for creating the matrix [H] by using Equation 4.22.
read(‘create_mtxH‘);

# Read the routine for creating the matrix [H1] by imposing a symmetry
# or antisymmetry condition.
read(‘Symmetry_create_mtxH1‘);

# Read necessary input parameters for the routine create_vecB
# Input parameters are scale (lambda=2),
# multiplicities of direction vectors (m1=1, m2=1 ,m3=2)
# support of box spline (N_{2m1, 2m2, 2m3}) whose direction vectors have
# multiplicites 2m1, 2m2, 2m3 (Bindex),
```

```
# values of N_{2m1, 2m2, 2m3} within its support (Bvalue)
# values of sequence p_{r1,r2}, given by Equation 5.2 (p).
read(‘allvarB1‘);

# Call the routine create_vecB
B:=create_vecB(lambda,m1,m2,m3,Bindex,p,Bvalue);

# Call the routine create_mtxH
H:= create_mtxH(B,lambda,m1,m2,m3);

# Solve the homogeneous system of equations [H][Q]=0
t:=rowdim(H);
b:=array(1..t,1..1,sparse);
Q:=linsolve(H,b,’r’,v);

# NECESSARY CONDITION FOR WAVELET DECOMPOSITIONS
sum:=0;
for i from 1 to rowdim(Q) do
sum:= sum + Q[i,1];
od;

# Call the routine Symmetry_create_mtxH1. There is no need for calling this
# if there are enough equations.
H1:=Symmetry_create_mtxH1(H,lambda,m1,m2,m3);

# Check the rank of matrix [H1] (= r)
r:=rank(H1);

# Solve the homogeneous system of equations [H1][Q1]=0
t1:=rowdim(H1);
b1:=array(1..t1,1..1,sparse);
Q1:=linsolve(H1,b1,’r’,v);

# Display the vector [Q1] (reconstruction sequence)
print(Q1);
```

MAPLE code for routine create_vecB:

```
create_vecB1:=proc(lambda,m1,m2,m3,Bindex,p,Bvalue)
# lambda = Scale
# m1,m2,m3 = Multiplicities of box spline's direction vectors.
#             The corresponding box spline is denoted as N_{m1,m2,m3}

# Bindex = array(1..2m1+2m3+1,1..2m2+2m3+1,sparse). We marked this array such
#          that Bindex[i+1,j+1]=C if the value of box spline N_{2m1,2m2,2m3}
#          is nonzero at an integer point (i,j) witin its support, otherwise
#          Bindex[i+1,j+1]=0, where point (i,j) is the C^th nonzero integer
#          point within the support of N_{2m1,2m2,2m3}.

# Bvalue = The nonzero values of N_{2m1,2m2,2m3} at integer points  within its
#          support, i.e, Bvalue[C,1]= value of N_{2m1,2m2,2m3} at the intger
#          point (i,j).

# p = array(1..(lambda-1)(m1+m3)+1,1..(lambda-1)(m2+m3)+1,sparse). We assigned
#     such that p[r1+1,r2+1]:= the value of sequence p_{r1,r2}. The sequence
#     p_{r1,r2} is given by Equation 5.2.


local i,j,al,bl,c1,t1,t2,ak,bk,ar,br,col_ele,count_ele,x,y,index;
N:=((lambda+1)*(m1+m3)-1)*((lambda+1)*(m2+m3)-1); # Size of vector [B]
c1:=array(1..1,1..N,sparse);
t1:=rowdim(Bindex);
t2:=coldim(Bindex);
col_ele:=0;
count_ele:=0;

# It follows from Step 4 in Sectiin 5.1 that vector [B] is independant of
#  l_{1} and l_{2}. Therefore select arbitary values for l_{1} and l_{2}.
al:=m1+m3-1;    # Value of l_{1}
bl:=m2+m3-1;    # Value of l_{2}

# Indices k_{1}, k_{2}, r_{1} and r_{2} in Equation 5.5.
Lk1:=lambda*al-(m1+m3)+1;               # Lower bound of index k1 of q_{k1,k2}
Uk1:=lambda*al+(lambda*(m1+m3))-1;      # Upper bound of index k1 of q_{k1,k2}
Lk2:=lambda*bl-(m2+m3)+1;               # Lower bound of index k2 of q_{k1,k2}
```

```
Uk2:=lambda*bl+(lambda*(m2+m3))-1;      # Upper bound of index k2 of q_{k1,k2}
Lr1:=lambda*al;                         # Lower bound of index r1 of p_{r1,r2}
Ur1:=lambda*al+(lambda-1)*(m1+m3)       # Upper bound of index r1 of p_{r1,r2}
Lr2:=lambda*bl;                         # Lower bound of index r2 of p_{r1,r2}
Ur2:=lambda*bl+(lambda-1)*(m2+m3)       # Upper bound of index r2 of p_{r1,r2}

# Create vector [B] by using Equations 5.5 and 5.6.
for ak from Lk1 by 1 to Uk1 do
for bk from Lk2 by 1 to Uk2 do
for ar from Lr1 by 1 to Ur1 do
for br from Lr2 by 1 to Ur2 do
x:= ar-ak+m1+m3+1; y:=br-bk+m2+m3+1;
if((x>=1 and x<=t1) and (y>=1 and y<=t2)) then
index:=Bindex[x,y];
if( index>0) then
col_ele:=col_ele+ Bvalue[index,1]*p[ar-lambda*al+1,br-lambda*bl+1];
fi;
fi;
od;
od;
count_ele:=count_ele+1;
c1[1,count_ele]:=col_ele;
col_ele:=0;
od;
od;
c1;
end;
```

MAPLE code for routine create_mtxH:

```
create_mtxH:=proc(B,lambda,m1,m2,m3)
# B = Vector [B] in Equation 5.7
# lambda = Scale
# m1,m2,m3 = Multiplicites of direction vectors
local i,j,al,bl,mtx1,mtx2,Q,ak,bk,col_ele,count_ele,row,row2,D,index1,index2,
t1,t2,t,r,u,v1,v2,v,alL,alU,blL,blU,;

r:=lambda+1;
```

```
t1:=r*(m1+m3)-2;      # Largest index k_{1} of q_{k1,k2}
t2:=r*(m2+m3)-2;      # Largest index k_{2} of q_{k1,k2}
t:=(t1+1)*(t2+1);     # Size of vector [B]
u:=t-(r*m3)*(r*m3-1); # The number of nonzero q_{k1,k2}. This is a Result 4 of
                      # Step 4 in Section 5.1.
alL:=trunc((1-lambda*(m1+m3))/lambda);         # Lower bound of l_{1}
alU:= trunc(((lambda+2)*(m1+m3)-3)/lambda);    # Upper bound of l_{1}
blL:= trunc((1-lambda*(m2+m3))/lambda);        # Lower bound of l_{2}
blU:= trunc(((lambda+2)*(m2+m3)-3)/lambda);    # Upper bound of l_{2}
v1:=alU-alL+1;
v2:=blU-blL+1;
v:=v1*v2; # Number of equations produced from Equation 5.7 by varying
          #  l_{1} and l_{2}
col_ele:=0;
count_ele:=0;
row:=0; row2:=0;
ok:=0;
mtx1:=array(1..v,1..t,sparse);
mtx2:=array(1..v,1..u,sparse);
Q:=array(0..t1,0..t2,sparse);

# Mark the nonzero q_{k1,k2}. It follows from Results 2 and 4 of Step 4 in
# Section 5.1 that the support of q_{k1,k2} is a hexogonal subset of
# rectangular integer grid [0,0]x[(lambda+1)(m1+m3)-2,(lambda+1)(m2+m3)-2].
# Therefore mark Q[k1,k2]=0 if q_{k1,k2}=0, otherwise Q[k1,k2]= count of
# q_{k1,k2} within the rectangular integer grid.
for i from 0 to t1 do
for j from 0 to t2 do
count_ele:= count_ele +1;
Q[i,j]:=count_ele;
if (i>=0 and i<=r*m3-2) and (j >=i+r*m2 and j<= (r*(m2+m3)-2)) then
Q[i,j]:=0 ;
fi;
if (i>=r*m1 and i<=r*(m1+m3)-2 ) and (j >=0 and j<= i-r*m1 ) then
Q[i,j]:=0 ;
fi;
od;
```

```
od;

# Produce equations from Equation 5.7 by varying l_{1},l_{2} within the bounds.
for al from alL to alU do
for bl from blL  to  blU do
row:=row+1;
for ak from (lambda*al-m1-m3+1) by 1 to (lambda*(m1+m3)+lambda*al-1) do
for bk from (lambda*bl-m2-m3+1) by 1 to (lambda*(m2+m3)+lambda*bl-1) do
col_ele:=col_ele+1;
if((ak>=0 and ak<=t1) and (bk>=0 and bk<=t2)) then
index1:=Q[ak,bk];
if( index1>0) then
mtx1[row,index1]:= B[1,col_ele];
fi;
fi;
od;
od;

col_ele:=0;
count_ele:=0;
for i from 0 to t1 do
for j from 0 to t2 do
if (Q[i,j]>0) then
count_ele:= count_ele +1;
index2:=Q[i,j];
if ( mtx1[row,index2]>0) then
mtx2[row,count_ele]:= mtx1[row,index2];
fi;
fi;
od;
od;
od;
od;
mtx2;
end;
```

MAPLE code for routine create_mtxH1:

```
Symmetry_create_mtxH1:=proc(H,lambda,m1,m2,m3)
# H = matrix [H], created by usinf the routine create_mtxH
# lambda = Scale
# m1,m2,m3 = Multiplicities of direction vectors
local i,j,col_ele,d,t,v,D;

v:=rowdim(H);
t:=coldim(H);
D:=(lambda+1)*(m1+m3)-1;

# Additional equations from symmetry or antisymmetry condition.
# Number of additional equations is d
d:=trunc(((lambda+1)^(2)*(m1*m2+m2*m3+m3*m1)-(lambda+1)*(m1+m2+m3) +1)/2);
mtx1:=array(1..v+d,1..t,sparse);

# Copy the rows of matrix [H] to matrix [mtx1]
for i from 1 to v do
for j from 1 to t do
mtx1[i,j]:=H[i,j];
od;
od;

# Get additional equations by imposing a condition of symmetry or antisymmetry
# on q_{k1,k2}.
col_ele := 0;
for i from v+1 to v+d do
col_ele:=col_ele+1;
mtx1[i,col_ele]:=1;
mtx1[i,t-col_ele+1]:= (-1)^(D);
od;
mtx1;
end;
```

OUTPUTS:

```
                                        r:=31
```

```
[                - v[3]                ]
[                                      ]
[                - v[2]                ]
[                                      ]
[                - v[1]                ]
[                                      ]
[      221                442          ]
[      --- v[1] + v[2] - --- v[3]      ]
[       59                 59          ]
[                                      ]
[          663         4571            ]
[        - --- v[1] + ---- v[3]        ]
[           59          59             ]
[                                      ]
[     2418                    1209      ]
[   - ---- v[3] + 28 v[2] + ---- v[1]  ]
[      59                      59      ]
[                                      ]
[                16 v[1]               ]
[                                      ]
[                - v[1]                ]
[                                      ]
[     4979                    9958      ]
[   - ---- v[1] - 28 v[2] + ---- v[3]  ]
[      59                      59      ]
[                                      ]
[        8613         38053            ]
[        ---- v[1] - ----- v[3]        ]
[         59           59              ]
[                                      ]
[    9711                    19422     ]
[  - ---- v[1] - 119 v[2] + ----- v[3] ]
[     59                       59      ]
[                                      ]
[               - 43 v[1]              ]
[                                      ]
```

```
                 [                16 v[1]                ]
                 [                                       ]
                 [  16588                    33176       ]
                 [  ----- v[1] + 119 v[2] - ----- v[3]  ]
                 [    59                       59        ]
                 [                                       ]
                 [        91717        22524             ]
                 [        ----- v[3] - ----- v[1]        ]
                 [          59           59              ]
                 [                                       ]
                 [  20332                    40664       ]
                 [  ----- v[1] + 184 v[2] - ----- v[3]  ]
                 [    59                       59        ]
                 [                                       ]
                 [                43 v[1]                ]
          Q1 := [                                       ]
                 [               - 43 v[1]               ]
                 [                                       ]
                 [   20332                    40664      ]
                 [ - ----- v[1] - 184 v[2] + ----- v[3] ]
                 [     59                       59       ]
                 [                                       ]
                 [         91717        22524            ]
                 [       - ----- v[3] + ----- v[1]       ]
                 [           59           59             ]
                 [                                       ]
                 [   16588                    33176      ]
                 [ - ----- v[1] - 119 v[2] + ----- v[3] ]
                 [     59                       59       ]
                 [                                       ]
                 [               - 16 v[1]               ]
                 [                                       ]
                 [                43 v[1]                ]
                 [                                       ]
                 [   9711                    19422       ]
                 [   ---- v[1] + 119 v[2] - ----- v[3]  ]
                 [    59                       59        ]
```

```
[                                      ]
[         8613         38053           ]
[       - ---- v[1] + ----- v[3]       ]
[          59           59             ]
[                                      ]
[    4979                    9958      ]
[    ---- v[1] + 28 v[2] - ---- v[3]   ]
[     59                     59        ]
[                                      ]
[                v[1]                  ]
[                                      ]
[              - 16 v[1]               ]
[                                      ]
[    2418                    1209      ]
[    ---- v[3] - 28 v[2] - ---- v[1]   ]
[     59                     59        ]
[                                      ]
[         663         4571             ]
[         --- v[1] - ---- v[3]         ]
[          59          59              ]
[                                      ]
[       221                442         ]
[     - --- v[1] - v[2] + --- v[3]     ]
[        59                 59         ]
[                                      ]
[                v[1]                  ]
[                                      ]
[                v[2]                  ]
[                                      ]
[                v[3]                  ]
```

```
> Q2:=subs(v[3]=-v, v[2]=-3*v, v[1]=-2*v, op(Q1));
```

```
                                [    v    ]
```

```
                  [         ]
                  [   3 v   ]
                  [         ]
                  [   2 v   ]
                  [         ]
                  [  - 3 v  ]
                  [         ]
                  [  - 55 v ]
                  [         ]
                  [  - 84 v ]
                  [         ]
                  [  - 32 v ]
                  [         ]
                  [   2 v   ]
                  [         ]
                  [   84 v  ]
                  [         ]
                  [  353 v  ]
                  [         ]
                  [  357 v  ]
                  [         ]
                  [   86 v  ]
                  [         ]
                  [  - 32 v ]
                  [         ]
                  [ - 357 v ]
                  [         ]
                  [ - 791 v ]
                  [         ]
                  [ - 552 v ]
                  [         ]
                  [  - 86 v ]
[Q2] := [         ]
                  [   86 v  ]
                  [         ]
                  [  552 v  ]
                  [         ]
```

```
[  791 v  ]
[         ]
[  357 v  ]
[         ]
[   32 v  ]
[         ]
[  - 86 v ]
[         ]
[ - 357 v ]
[         ]
[ - 353 v ]
[         ]
[  - 84 v ]
[         ]
[  - 2 v  ]
[         ]
[   32 v  ]
[         ]
[   84 v  ]
[         ]
[   55 v  ]
[         ]
[   3 v   ]
[         ]
[  - 2 v  ]
[         ]
[  - 3 v  ]
[         ]
[   - v   ]
```

Appendix E

Example 2 in Chapter 5

This appendix consists MAPLE files for driver and routines of creating vector $[Q]$. Here we consider the box spline with direction vectors $\xi^1 = (1, 0)^T$, $\xi^2 = (0, 1)^T$, $\xi^3 = (1, 1)^T$ under the four scale relation. The multiplicities of direction vectors are 1, 1 and 2.

MAPLE code for driver file:

```
with(linalg);
# Read the routine for creating the vector [B]
read(‘create_vecB‘);

# Read the routine for creating the matrix [H]
read(‘create_mtxH‘);

# Read the routine for creating the vector [Q]
read(‘create_vecQ‘);

# Read necessary input parameters for the routine create_vecB
# Input parameters are scale (lambda=4),
# multiplicities of direction vectors (m1=1, m2=1 ,m3=2)
# support of box spline (N_{2m1, 2m2, 2m3}) whose direction vectors have
# multiplicites 2m1, 2m2, 2m3 (Bindex),
# values of N_{2m1, 2m2, 2m3} within its support (Bvalue)
# values of sequence p_{r1,r2}, given by equation 5.2 (p).
```

```
read(‘allvarB2‘);

# Call the the routine create_vecB
B:=create_vecB(lambda,m1,m2,m3,Bindex,p,Bvalue);

# Call the the routine create_mtxH
H:= create_mtxH(B,lambda,m1,m2,m3);

# Call the the routine create_vecQ
Q:=create_vecQ(lambda,m1,m2,m3,Bindex,p,Bvalue);

# NECESSARY CONDITION FOR WAVELET DECOMPOSITIONS
sum:=0;
for i from 1 to rowdim(Q) do
sum1:= sum1 + Q[i,1];
od;

# TEST [H][Q]=0
z:=multiply(H,Q);
```

MAPLE code for routine create_vecQ

```
create_Q:= proc(lambda,m1,m2,m3,Bindex,p,Bvalue)
# lambda = Scale
# m1,m2,m3 = Multiplicities of box spline's direction vectors.
#             The corresponding box spline is denoted as N_{m1,m2,m3}

# Bindex = array(1..2m1+2m3+1,1..2m2+2m3+1,sparse). We marked this array such
#          that Bindex[i+1,j+1]=C if the value of box spline N_{2m1,2m2,2m3}
#          is nonzero at an integer point (i,j) witin its support, otherwise
#          Bindex[i+1,j+1]=0, where point (i,j) is the Cˆth nonzero integer
#          point within the support of N_{2m1,2m2,2m3}.

# Bvalue = The nonzero values of N_{2m1,2m2,2m3} at integer points  within its
#          support, i.e, Bvalue[C,1]= value of N_{2m1,2m2,2m3} at the intger
#          point (i,j).

# p = array(1..(lambda-1)(m1+m3)+1,1..(lambda-1)(m2+m3)+1,sparse). We assigned
```

```
#     such that p[r1+1,r2+1]:= the value of sequence p_{r1,r2}. The sequence
#     p_{r1,r2} is given by Equation 5.2.

local i,j,al,bl,c1,c2,t1,t2,t,ak,bk,col_ele,count_ele,x,y,index1,index2,row,
r,Nu,Mu,Q,v ;
r:=lambda+1;
Nu:=(lambda+1)*(m1+m3)-2;     # Largest index k_{1} of q_{k1,k2}
Mu:=(lambda+1)*(m2+m3)-2;     # Largest index k_{2} of q_{k1,k2}
# The number of integer points within the ranges of k_{1} and k_{2}
v:=(Nu+1)*(Mu+1);
# The number of nonzero q_{k1,k2}
t:=r^(2}*(m1*m2+m2*m3+m3*m1)-r*(m1+m2+m3)+1;

c1 := array(1 .. v,1 .. 1,sparse);
c2:=array(1..t,1..1,sparse);
Q:=array(0..Nu,0..Mu,sparse);
t1 := rowdim(Bindex);
t2 := coldim(Bindex);
count_ele:=0;
col_ele:= 0;
row:= 0;

# Mark the nonzero q_{k1,k2}. It follows from Results 2 and 4 of Step 4 in
# Section 5.1 that the support of q_{k1,k2} is a hexogonal subset of
# rectangular integer grid [0,0]x[(lambda+1)(m1+m3)-2,(lambda+1)(m2+m3)-2].
# Therefore mark Q[k1,k2]=0 if q_{k1,k2}=0, otherwise Q[k1,k2]= count of
# q_{k1,k2} within the rectangular integer grid.
for i from 0 to Nu do
for j from 0 to Mu do
count_ele:= count_ele +1;
Q[i,j]:=count_ele;
if (i>=0 and i<=r*m3-2) and (j >=i+r*m2 and j<= (r*(m2+m3)-2)) then
Q[i,j]:= 0;
fi;
if (i>=r*m1 and i<=r*(m1+m3)-2 ) and (j >=0 and j<= i-r*m1 ) then
Q[i,j]:= 0;
fi;
```

```
od;
od;

# Create q_{k1,k2} by using Equation 5.12
for ak from 0 to Nu do
for bk from 0 to Mu do
row:=row+1;
for al from 0 to (lambda-1)*(m1+m3) do
for bl from 0 to (lambda-1)*(m2+m3) do
x := ak-al+2;
y := bk-bl+2;
if (1 <= x) and (x <= t1) and (1 <= y) and (y <= t2) then
index2 := Bindex[x,y];
if (0 < index2)  then
col_ele := (col_ele+Bvalue[index2,1]*p[al+1,bl+1]);
fi;
fi;
od;
od;

c1[row,1] := col_ele*((-1)^ak);
col_ele := 0;
od;
od;

# Select only nonzero q_{k1,k2}
row:=0;
count_ele:=0;
for i from 0 to Nu do
for j  from 0 to Mu do
count_ele:=count_ele+1;
index1:=Q[i,j];
if (0 < index1) then
row:=row+1;
c2[row,1]:=c1[count_ele,1];
fi;
od;
```

```
od;
c2;
end;
```

MAPLE code for routine create_mtxH is same codes as in Appendix D.

Bibliography

[1] R. H. Bartels, J. C. Beatty, and B. A. Barsky. *An Introduction to Splines for use in Computer Graphics and Geometric Modeling*. Morgan Kaufmann, 1987.

[2] G. Battle. A block spin construction of ondelettes part i: Lemarie functions. *Comm. Math. Phys.*, 110:601–615, 1987.

[3] A.P. Calderon. Intermediate spaces and interpolation, the complex method. *Studia Math.*, 24:113–190, 1964.

[4] A.S. Cavaretta and C. A. Micchelli. Subdivision algorithms. In *Mathematical Methods in Computer Aided Geometric design*, pages 115–153, N.Y, 1989. Academic Press.

[5] C. K. Chui. *Wavelets: with an emphasis on time-frequency analysis,*. Lecture Notes. Texas A&M University, 1990.

[6] C. K. Chui. *An Introduction to wavelets*. Academic Press, N.Y, 1992.

[7] C. K. Chui and H. Diamond. Characterization of multivariate quasi-interpolation formulas and their applications. *Numer. Math.*, 57:105–121, 1990.

[8] C. K. Chui, J. Stockler, and J. D. Ward. Compactly supported box spline wavelets. *Approx. Theory and Appl.*, 8:77–100, 1992.

[9] C. K. Chui and J. Z. Wang. *A cardinal spline approach to wavelets*. CAT Report # 211. Texas A&M University, 1990.

[10] C. K. Chui and J. Z. Wang. *A general frame work of compactly supported splines and wavelets*. CAT Report # 219. Texas A&M University, 1990.

[11] C. K. Chui and J. Z. Wang. On compactly supported spline wavelets and a duality principle. *Trans. Amer. Math. Soc.*, 330:903–916, 1992.

[12] C.K. Chui. *Multivariate Splines*. CBMS-NSF Series in Applied Math. #54. SIAM Publ., Philedelphia, 1988.

[13] A. Cohen. PhD thesis, Univ. Paris-Dauphine, 1990.

[14] M. Daehlen and T. Lyche. Box splines and applications. In *Geometric Modeling: Methods and Applications*, pages 35–93. Springer Series in Computer Graphics–Systems and Applications, Springer Verlag, 1991.

[15] I. Daubechies. Orthonormal bases of compactly supported wavelets. *Comm. Pure and Appl. Math.*, 41:909–996, 1988.

[16] I. Daubechies. *Wavelets*. CBMS-NSF Conf. Lecutre Notes. Lowell Univ., 1990.

[17] C. de Boor and K. Hollig. B-splines from parallelepipeds. *J. Analyse Math.*, 42:99–115, 1982/83.

[18] H. Diamond, L. A. Raphel, and D. Williams. Compressed representation of curves and images using a multiresolution box-spline framework. *Research partially supported by U.S Air Force Office of Scientific Research Contract No. 91NM062*, West Virginia University.

[19] D. Gabor. Theory of communication. *J.IEE(London).*, 93:429–457, 1946.

[20] A Grossmann, Kronland-Martinet, and J . Morlet. Reading and understanding continuous wavelet transform. In *Wavelets: Time frequency methods and phase space*, pages 2–10. Springer-Verlag, New York, 1989.

[21] A Grossmann and J . Morlet. Decomposition of hardy functions into square integrable wavelets of constant shape. *SIAM J. Math. Anal.*, 15:723–736, 1984.

[22] R. Q. Jia and C. A. Micchelli. Using the refinement equations for the construction of pre-wavelets ii. In *Curves and Surfaces*, pages 209–246. Academic Press, San Diego, 1991.

[23] P. G. Lemarie. Ondelettes a localisation exponentielle. *Journal de Math. Pures et Appl.*, 67:227–236, 1988.

[24] S. Mallet. PhD thesis, Univ. of Pennsylvania, Philedelphia, 1988.

[25] S. Mallet. Multifrequency channel decompositions of images and wavelet models. *IEEE Trans. ASSP*, 37:2091–2110, 1989.

[26] S. Mallet. Multiresolution representations and wavelet orthonormal bases of $\mathcal{L}^2(\mathcal{R})$. *Trans. Amer. Math. Soc.*, 315:69–87, 1989.

[27] Y. Meyer. La principe d'incertitude, bases hilbertiennes et algebres d'operateurs. *Seminaire Bourbaki*, No 662, 1985-1986.

[28] Y. Meyer. *Ondelettes et Operateurs, (two volumes).* Herman, Paris, 1990.

[29] Y. Meyer. *Ondelettes et functions splines, Seminaire Equations aux derivees partielles.* No 662. Herman, Paris, Dec. 1986.

[30] A. E. Middleditch. The representation and manipulation of convex polygons. Internal Technical Memo 84:6, Computer Aided Engineering Group, Polytechnic of Central London, February 1984.

[31] S. Riemenschneider and Z. Shen. *Box splines, cardinal series, and wavelets, in Approximation Theory and Functional Analysis.* Academic Press, New York, 1990.

[32] I. J. Schoenberg. *Cardinal Spline Interpolation.* CBMS-NSF Series in Applied Math. #12. SIAM Publ., Philedelphia, 1973.

[33] J. O. Stroemberg. A modified franklin system and higher-order spline systems of $\mathcal{R}^n$ as unconditional bases for hardy spaces. In *Conference in Harmonic Analysis in honor of Antoni Zygmund*, pages 475–493. VOL II, Wadsworth Math. Series, 1983.