9-30-2009

The Multilevel Structures of NURBs and NURBlets on Intervals

Weiwei Zhu
University of Missouri-St. Louis, weiwei.zhu@gmail.com

Recommended Citation

Zhu, Weiwei, "The Multilevel Structures of NURBs and NURBlets on Intervals" (2009). *Dissertations*. 509.
https://irl.umsl.edu/dissertation/509

The Multilevel Structures of NURBs and NURBlets on Intervals

by

Weiwei Zhu

2009

Major Professor

Chairman, Examining Committee

Committee Member

Committee Member

Committee Member

Dean of The Graduate School

Acknowledgments

I want to thank my advisor Dr. Wenjie He, and Professor Charles K. Chui for the big help in my dissertation research.

Abstract

This dissertation is concerned with the problem of constructing biorthogonal wavelets based on non-uniform rational cubic B-Splines on intervals. We call non-uniform rational B-Splines "NURBs", and such biorthogonal wavelets "NURBlets". Constructing NURBlets is useful in designing and representing an arbitrary shape of an object in the industry, especially when exactness of the shape is critical such as the shape of an aircraft. As we know presently most popular wavelet models in the industry are approximated at boundaries. In this dissertation a new model is presented that is well suited for generating arbitrary shapes in the industry with mathematical exactness throughout intervals; it fulfills interpolation at boundaries as well.

Table of Contents

List of Figures

Chapter 1

Introduction

B-Splines play an important role in the sophisticated geometric modeling, computer graphics, computer-aided design and manufacturing(CAD and CAM), and many other areas in the industry [6, 9, 26, 27, 34, 38, 53]. After a few decades of tremendous achievement, B-Splines have become the industry standards. However, B-Splines cannot represent many important shapes precisely, for instance, widely used conic shapes. In areas such as the aircraft industry accurate shape modeling is critical, otherwise the airplane would be crushed by air pressure. NURBs (Non-uniform Rational B-Splines) successfully solve the problem by representing both analytic and free-form surfaces with mathematical exactness and resolution independence. They are ubiquitous for CAD, CAM and computed-aided engineering (CAE), and therefore are industry standards with wide applications [33, 52, 53]. The role that NURBs play in CAD/CAM/CAE is like "that of the English language in science and business" [53].

On the other hand, multiresolution analysis (MRA) which stemmed from wavelet analysis has been applied in mass data processing in almost every field, from data mining, image processing, computer graphics, medical research, stock market, to the Internet. It provides a hierarchy structure from coarse levels to fine levels in the amount of detail. With this technique, B-Splines could have exerted their strengths in fast computation and local smoothness-controlling. Unfortunately, NURBs have not succeeded in "interfacing with" MRA because of their complicated rational structures. This limits the power of NURBs greatly in their industrial applications.

With MRA technique and based on B-Splines, wavelets have been extensively applied in both theoretical and applied areas [14, 15, 22, 62]. However, one major challenge remains: the models are built on the whole number axis while in real life, they should have boundaries. For example, this disadvantage makes JPEG 2000 have blurred edges in highly compressed images. Building innovative wavelet tools on intervals therefore becomes very crucial.

In this dissertation biorthogonal wavelets are constructed based on non-uniform rational cubic B-Splines on intervals, with natural cubic B-Splines at the boundaries. We call such wavelets “NURBlets”.

1.1 Overview

1.1.1 Parametric Functions

To represent curves and surfaces in mathematical functions, explicit functions are not good enough, since for each variable value, it allows only one function value correspondingly. Therefore it won’t allow curves to loop back. The solution is to represent each coordinate of a point on the curve by an explicit function of an independent parameter such as

$$C(t) = (x(t), y(t)) \quad \text{for} \quad t \in [a, b].$$

For example, if we define

$$x(t) = sin(t), \quad y(t) = cos(t) \quad \text{for} \quad t \in [0, 2\pi],$$

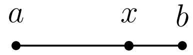
then function $C(t)$ represents a circle centered at original point with radius 1.

1.1.2 Smoothness

A function for curve $C(t)$ is said to have r-derivative continuity at a point if it has k-derivative *continuity* for $k \leq r$ at that point. Equivalently, we say it has C^r smoothness at that point. If a curve has C^r smoothness at each point on an interval, then it has C^r *smoothness* on that interval.

1.1.3 Barycentric Coordinates

Consider $x \in [a, b]$ $(a < b)$ such that



If we let

$$u = \frac{x - a}{b - a}$$

then the Barycentric coordinate for x are defined as

$$x = (1 - u)a + ub. \tag{1.1.1}$$

1.1.4 Inner Products

Suppose two functions $f(x)$, $g(x) \in L^2(\mathbb{R})$, then the inner product is defined by

$$\langle f, g \rangle = \int f(x)\bar{g}(x)dx, \tag{1.1.2}$$

and the inner product with weight ω is defined by

$$\langle f, g \rangle_\omega = \int_{-\infty}^{\infty} \omega^2 f(x)\bar{g}(x)dx. \tag{1.1.3}$$

1.2 Dissertation Outline

The remainder of this dissertation expounds the idea we outline above. The next chapter gives detailed background on B-Splines, NURBs, multiresolution analysis and wavelets.

Chapter 3 constructs natural cubic B-Splines at boundaries. The matrices for refinement relation in scaling functions are given.

We discuss the characters of MRA of NURBs in chapter 4. Weighted MRA NURBs at boundaries are specified, with examples and explanations.

Finally in chapter 5, we explain how to construct biorthogonal wavelets based on NURBs on intervals — such wavelets are called NURBlets. Lifting scheme is first discussed since it is the technique we use for the construction. Biorthogonal NURBlet construction in single-knot and two-knot insertion is explored, followed by elaboration on constructing the key matrix F.

We also investigate one vanishing moment in one knot insertion case. The chapter ends by three examples presenting our algorithms for the biorthogonal NURBlet construction.

Chapter 6 concludes the dissertation. A summary of the work is given, followed by an outline of future research directions.

Chapter 2

Introduction to B-Splines, NURBs, MRA and Wavelets

The model presented here arises from B-Splines and wavelets. Before providing the details of the model, a review of B-Splines, NURBs, multiresolution analysis and wavelet techniques is presented in this chapter.

2.1 B-Splines

Spline techniques have been used long before the computer age. A draftsman used a long flexible strip tied with lead weights in places to get a smooth curve. The curve varies in curvature depending on the positions of the lead weights. Such a strip is called a *spline*, and the positions where lead weights are tied at are called *control points* since they control the shape of the spline.

A spline has a few essential properties. First, it has enough smoothness. In other words, it has at least C^1 or higher in continuity. Secondly, if we move a control point to a new position, only the shape of the spline curve nearby the point is changed, while overall shape will not be affected. This is known as *local control.*

To find a mathematical representation for a spline, we define a curve $C(x)$ blending control points p_i by some "good" functions $f_i(x)$ as weights such that $C(x) = \sum_i p_i f_i(x)$. We wish our weight functions to be simple, differentiable, mathematically well understood, and able to represent precisely all free-form shapes. Intuitively, polynomials are our first candidates because they are simple, computationally efficient, and easy to work with. However, polynomials usually do not convey clear geometric insights. For this reason we choose a special type of polynomials, Bernstein polynomials, as our weight functions. The curves weighted by Bernstein polynomials are called *Bézier curves.*

2.1.1 Bézier Curves

In late 1950s and early 1960s, P. Bézier and P. de Casteljau independently found the same curves, Bézier curves. Given a set of control points, a Bézier curve interpolates two endpoints and approximates the rest points smoothly. If we connect consecutive control points, we get a polygon called *control polygon* since the Bézier curve is in its convex hull. Figure 2.1 shows a cubic Bézier curve with four control points p_k $(k = 0, \dots, 3)$.

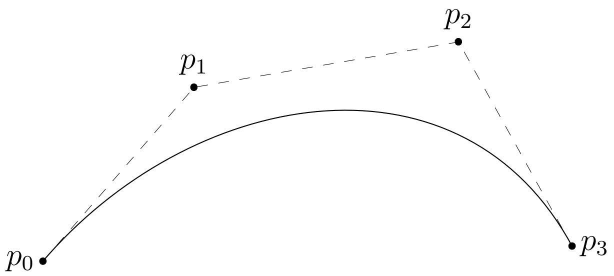


Figure 2.1: Cubic Bézier curve.

Let $C(x)$ be such a Bézier curve function with n degree on interval [0, 1], then it can be expressed as

$$C(x) = \sum_{i=0}^{n} p_i B_{i,n}(x), \qquad x \in [0,1]. \tag{2.1.1}$$

Coefficients p_i are control points, and $B_{i,n}(x)$ are called *Bernstein polynomial bases* with degree n such that

$$B_{i,n}(x) = \binom{n}{i}(1-x)^{n-i}x^i, \qquad x \in [0,1]. \tag{2.1.2}$$

A few important properties of Bernstein polynomials on [0,1] are [53]:

1. *Nonnegativity*: $B_{i,n}(x) \geq 0$ for all i, n and $x \in [0,1]$;

2. $B_{0,n}(0) = B_{n,n}(1) = 1$;

3. *Partition of unity*: $\sum_{i=0}^{i=n} B_{i,n}(x) = 1$ for $x \in [0,1]$;

4. *Recursive definition*: $B_{i,n}(x) = (1-x)B_{i,n-1}(x) + xB_{i-1,n-1}(x)$;

5. *Symmetry by* $x = \frac{1}{2}$: $B_{i,n}(x) = B_{i,n}(1-x)$;

6. *Derivative*: $B_{i,n}(x)' = n(B_{i-1,n-1}(x) - B_{i,n-1}(x))$.

These properties yield the following geometric properties of Bézier curves:

1. *Interpolating at endpoints;*

2. *The curves are contained in the convex hulls defined by their control points*;

3. *Invariant under affine transformation*;

4. *Derivative*: $C(x)' = n\sum_{i=0}^{n-1} B_{i,n-1}(p_{i+1} - p_i)$.

The last property indicates that the derivative of a Bézier curve is a linear combination of a few neighboring control points. At the endpoints, the first and second derivatives are linear combinations of only first or last few control points such that

$$\begin{array}{cc} C(0)' = n(p_1 - p_0) & C(0)'' = n(n-1)(p_2 - 2p_1 + p_0) \\ C(1)' = n(p_n - p_{n-1}) & C(1)'' = n(n-1)(p_n - 2p_{n-1} + p_{n-2}). \end{array} \tag{2.1.3}$$

In Barycentric coordinates, a Bézier curve on interval $[a,b]$ becomes

$$C(x) = \sum_{i=0}^{n} p_i B_{i,n}(x) \tag{2.1.4}$$

$$= \sum_{i=0}^{n} p_i \binom{n}{i} (\frac{b-x}{b-a})^{n-i} (\frac{x-a}{b-a})^i \tag{2.1.5}$$

$$= \sum_{i=0}^{n} p_i \binom{n}{i} (1-u)^{n-i} u^i \tag{2.1.6}$$

where

$$u = \frac{x-a}{b-a} \quad \text{for } x \in [a,b].$$

Its first and second derivatives at the endpoints are

$$\begin{array}{ll} C'(a) = \frac{n(p_1-p_0)}{b-a} & C'(b) = \frac{n(p_n-p_{n-1})}{b-a} \\ C''(a) = \frac{n(n-1)(p_0-2p_1+p_2)}{(b-a)^2} & C''(b) = \frac{n(n-1)(p_{n-2}-2p_{n-1}+p_n)}{(b-a)^2}. \end{array} \tag{2.1.7}$$

2.1.2 de Casteljau's Algorithm

On interval $[a,b]$, given a Bézier curve, how to find a specific point on it? Or how do we compute the function value $C(x)$ in the equation (2.1.1)? P. de Casteljau proposed a recursive algorithm. The algorithm has following advantages:

1. It is simple for computer programming;

2. It has clear geometry meaning (see Fig 2.2);

3. It is fast enough to draw curves at interactive rates.

The algorithm is illustrated in the following. On $[a,b]$, let a Bézier curve with a set of control points $\{p_0,\ldots,p_n\}$ be denoted by $C(x)$, then for any $x \in [a,b]$, the corresponding point $P(x,y)$ on the curve cuts the curve into two curve segments $C_l(x)$ and $C_r(x)$, which are defined on control points $\{p_0,\cdots,p_{n-1}\}$ and $\{p_1,\cdots,p_n\}$ respectively. The recursive definition of Bézier basis functions leads to

$$C(x) = (1-x)C_l(x) + xC_r(x). \tag{2.1.8}$$

The two curve segments can be subdivided again. We then have the following recursive algorithm if we let

$$p_i^k = (1-x)p_i^{k-1}(x) + xp_{i+1}^{k-1}(x), \qquad k = 1,\cdots,n, \;\; i = 0,\cdots,n-k, \tag{2.1.9}$$

where $p_i^k(x)$ is the i^{th} control point for Bézier curve of k degree, then $C(x) = p_0^n$. Figure 2.2 shows an example of computing $C(x)$ when $x = \frac{2}{3}$.

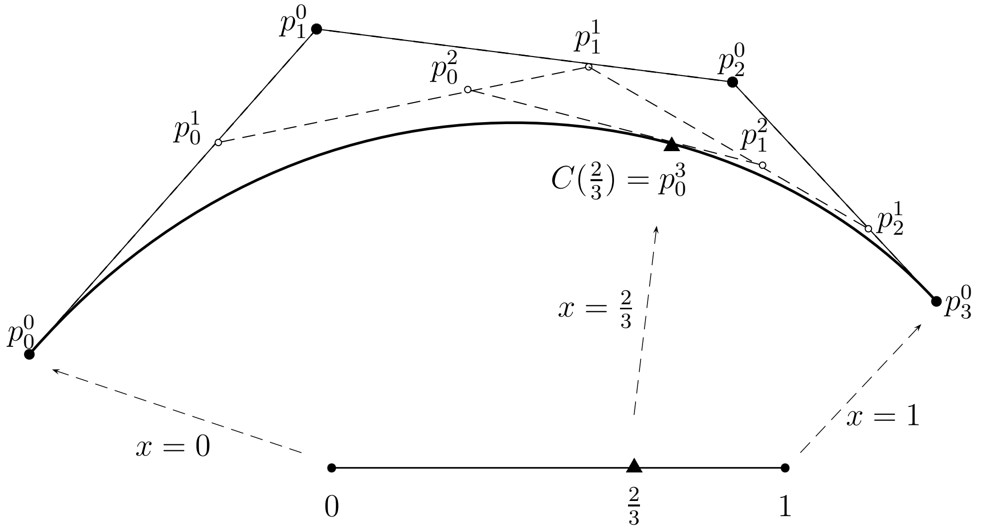


Figure 2.2: de Casteljau's algorithm of computing $C(\frac{2}{3})$.

Pseudo code:

```
deCasteljauAlgorithm($n, \tau, P$)

/* de Casteljau's algorithm: computing a point value on curve $f(x)$
Input: $n$ — degree of Bézier basis functions
       $\tau$ — $x$-coordinate of the point and $\tau \in [0, 1]$
       P— array of control points in which $P[i] = p_i$ for $i = 0, \ldots, n$
Output: $f(\tau)$ */
       for(i = 0; i <= n; i + +)
           Q[i] = P[i];
       for(i = 1; i <= n; i + +)
           for(j = 0; j <= n − i; j + +)
               Q[j] = (1 − u)Q[j] + uQ[j + 1];
       return Q[0];
```

We often use a triangle scheme to illustrate the de Casteljau's algorithm. See Figure 2.3.

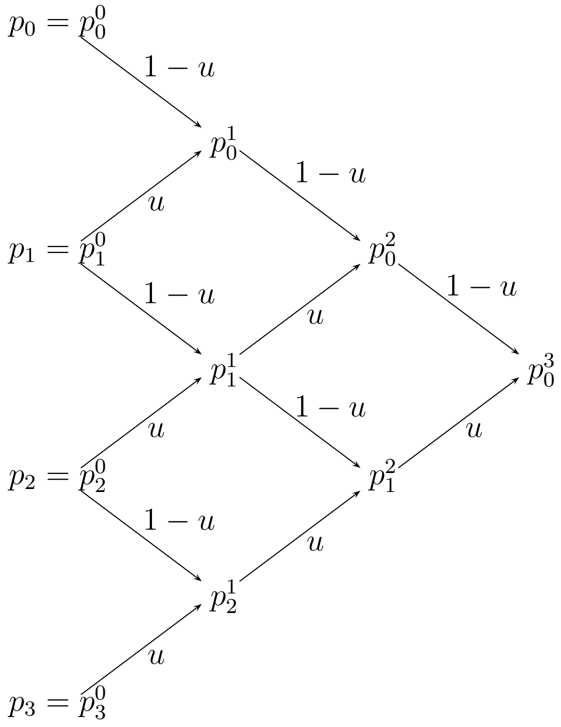


Figure 2.3: de Casteljau's algorithm.

2.1.3 B-Spline Basis Functions

Though Bézier curves beautifully render curves, they have a few drawbacks:

- They are globally supported on the interval;
- They cannot interpolate control points except endpoints;
- To approximate a curve with $n + 1$ control points, a Bézier curve representation requires polynomials of degree n which may be high and hence impractical.

We need to look for "better" basis functions.

Let's go back to our original spline curve, $C(x)$. Suppose $C(x)$ has $l + 1$ control points p_i $(j = 0, 1, \ldots, l)$ with corresponding basis functions $f_i(x)$

$$C(x) = \sum_{i=0}^{l} p_i f_i(x). \tag{2.1.10}$$

For each x, $C(x)$ is a linear combination of control points p_i weighted by $f_i(x)$. $f_i(x)$ are basis functions so we called them *B-Spline basis functions*. Ideal B-Spline basis functions should have desirable degree (so that the curve attains certain degree of smoothness), continuity (so that editing control points will not influence the continuity of the curve) and properties listed for Bézier basis functions (section 2.1.1). Figure 2.4 indicates uniform basis functions for a set of control points. They are identical yet on different intervals. That is, they influence their own control points with the same weight and same support length but on different time intervals. However, in general, these weights might be different both in degree and domain span of their influence, like Figure 2.5 has shown. The numbers in domain axis partition the domain into different intervals for the basis functions and consequently different basis functions are formed. We call these numbers in domain "*knots*", and the sequence of knots a "*knot vector*".

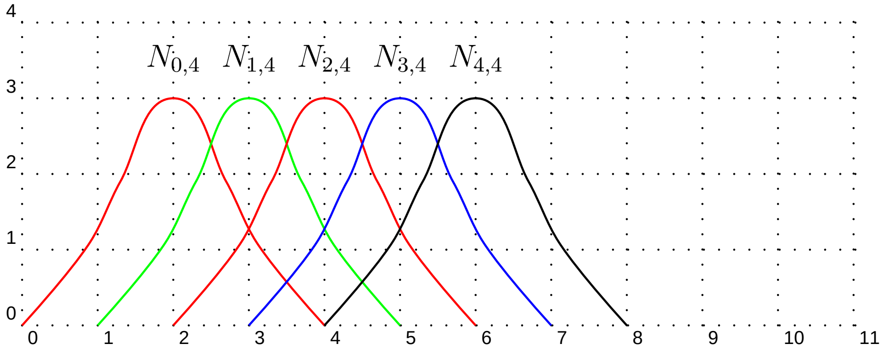


Figure 2.4: Uniform basis functions.

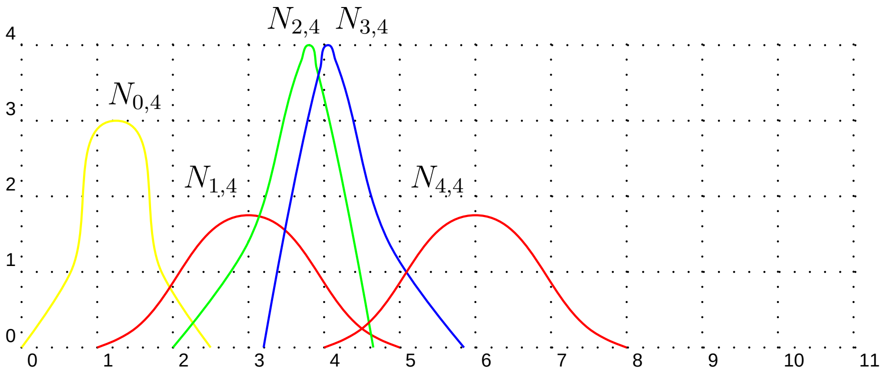


Figure 2.5: Non-uniform basis functions.

We then can define our basis B-Spline functions. Among a few different versions in definition, we adopt the following one [53].

Given a knot sequence $\mathbf{x} = \{x_0, x_1, \ldots, x_n\}$ and basis functions of order m, B-Spline basis functions are defined by:

$$N_{k,1}(x) = \begin{cases} 1, & \text{if } x \in [x_k, x_{k+1}) \\ 0, & \text{otherwise} \end{cases}$$

$$N_{k,m}(x) = \frac{x - x_k}{x_{k+m-1} - x_k} N_{k,m-1}(x) + \frac{x_{k+m} - x}{x_{k+m} - x_{k+1}} N_{k+1,m-1}(x). \tag{2.1.11}$$

For example, let $m = 4$ with uniform interval [0,4], then $N_4(x)$, the *cubic B-Spline,* is

$$N_4(x) = \begin{cases} \frac{1}{6}x^3, & \text{if } x \in [0,1]; \\ \frac{2}{3} - 2x + 2x^2 - \frac{1}{2}x^3, & \text{if } x \in [1,2]; \\ -\frac{22}{3} + 10x - 4x^2 + \frac{1}{2}x^3, & \text{if } x \in [2,3]; \\ \frac{1}{6}(4-x)^3, & \text{if } x \in [3,4]. \end{cases} \tag{2.1.12}$$

Interval $[x_i, x_{i+1})$ is called i^{th} *knot span.* Let m and $l + 1$ be the order and number of basis functions on interval $[a, b]$. A few important properties of B-Spline basis functions can be derived as follows:

1. $N_{i,m}(x)$ is a polynomial of degree $m - 1$ on $[x_i, x_{i+m}]$, for all i;

2. $N_{i,m}(x)$ has support $[x_i, x_{i+m}]$, i.e., $N_{i,m}(x) = 0$ for $x \notin [x_i, x_{i+m}]$;

3. $N_{i,m}(x) > 0$, for $x \in (x_i, x_{m+i})$;

4. Partition of unity: $\sum_{j=0}^{l} N_{j,m}(x) = 1$ for any $x \in [a, b]$;

5. At most m of $N_{i,m}$ are nonzero for an arbitrary knot span $[x_i, x_{i+1})$;

6. Derivative: $N'_{i,m}(x) = \frac{m}{x_{i+m} - x_i} N_{i,m-1}(x) - \frac{m}{x_{i+m+1} - x_{i+1}} N_{i+1,m-1}(x)$ and $N^{(k)}_{i,m} = m(\frac{N^{(k-1)}_{i,m-1}}{x_{i+m} - x_i} - \frac{N^{(k-1)}_{i+1,m-1}}{x_{i+m+1} - x_{i+1}})$;

7. Continuity: basis function $N_{k,m}(x)$ is C^{m-1-r} continuous at a knot with multiplicity of r.

2.1.4 B-Spline Curves

From Bézier curves to B-Spline curves, a few important properties are achieved. First, it discards the idea of representing a curve with only one Bézier curve. Instead, it applies a piecewise polynomial for the representation. In other words, it divides the whole curve into curve segments where the curves connect with certain degree of continuity at the breaking points. The curve with continuity r at the breakpoint x_i satisfies $C_i^{(j)}(x_i) = C_{i+1}^{(j)}(x_i)$ for $0 \leq j \leq r$ [53]. Hence, moving a local control point affects only two neighboring fragment curves and this gives local support. Secondly, we are free to have polynomial curve segments with a low degree. In fact we can even choose polynomial functions with diverse degrees for a basis. For example, if some part of the curve needs to be smoother (such as a sharp part), we can choose a basis function with higher degree for that curve segment. Figure 2.6 shows a spline curve with three cubic curve segments.

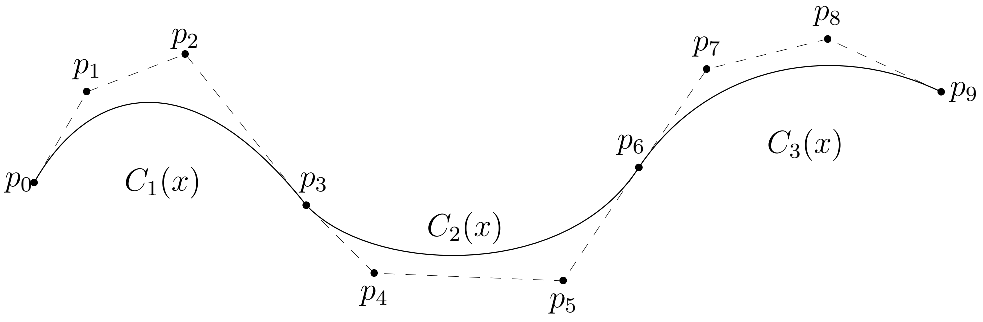


Figure 2.6: Cubic B-Spine with 3 curve segments.

In B-Spline curve

$$C(x) = \sum_{i=0}^{n} p_i f_i(x),$$

basis functions have different knot spans. If knot spans are clustered at the endpoints of a single interval, it becomes a Bézier curve. So Bézier curves are special cases of B-Spline curves.

B-Spline curves share many important properties with Bézier curves, but B-Spline curves have more desirable characters than Bézier curves do. The list below shows several of the most important properties of B-Spline curves.

1. B-Spline curve $C(x)$ is a piecewise polynomial curve with each curve segment of degree $m-1$;
2. Let $s+1$ be number of knots, $l+1$ be number of basis functions, and m be the order of basis functions, then $l = s - m$;
3. Endpoint interpolation: $C(a) = p_0$ and $C(b) = p_n$ on the interval $[a, b]$;
4. Local support: moving p_i only affects the part of curve on interval $[x_i, x_{i+m})$;
5. Invariant under affine transformation $\alpha(\sum_{i=0}^{m-1} p_i N_{i,m}) = \sum_{i=0}^{m-1} \alpha(p_i) N_{i,m}$;
6. Convex hull properties: the curve is constrained in the convex hull formed by its control polygon;
7. Variation diminishing property: if the curve is in a plane, then no straight line intersects a B-Spline curve more times than it intersects the curve's control polygons;
8. Differentiability: $C(x)$ is infinitely differentiable in knot intervals, and $m-1-k$ continuously differentiable at a knot with multiplicity k;
9. Derivatives:

$$C^{(k)}(x) = \sum_{i=0}^{n-k} N_{i,m-k}^{(k)}(x) p_i^{(k)} \tag{2.1.13}$$

where $p_i^{(k)}$ are control points corresponding to $N_{i,m-k}^{(k)}$ such that

$$p_i^{(k)} = \begin{cases} p_i & k = 0 \\ \frac{m-k}{x_{i+m}-x_{i+k}} N_{i,p-k}(x)(p_{i+1}^{(k-1)} - p_i^{(k-1)}) & k > 0 \end{cases}$$

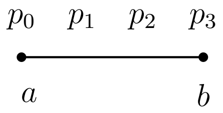
2.1.5 Bézier Nets

A Bézier curve defined by

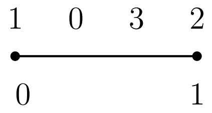
$$C(x) = \sum_{i=0}^{n} p_i B_{i,n}(x), \quad 0 \le x \le 1$$

is determined by coefficients p_i as soon as basis functions $B_{i,n}(x)$ are given. Therefore, we need to exclusively investigate the relation among coefficients on the interval with geometric insights. A *Bézier Net* provides a simple approach for the purpose.

For example, for a cubic Bézier curve $f(x) = \sum_{k=0}^{3} p_k B_k^3(x)$ where p_i are coefficients and $x \in [a, b]$, the Bézier net is:

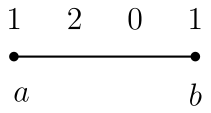


Specifically,



$$\longrightarrow 1(1-x)^3 + 3(1-x)x^2 + 2x^3,$$

and



$$\longrightarrow 1u^3 + 2u^2v + 1v^3$$

with

$$u = \frac{b-x}{b-a}, \quad v = \frac{x-a}{b-a}.$$

Compared with the polynomial notation, Bézier nets give a simpler and more straightforward insight for a spline.

Figure 2.7 gives the Bézier net for cubic B-Splines defined in (2.1.12) on interval $[t_i, t_{i+4}]$ by [13]. We will adopt this notation throughout this dissertation.

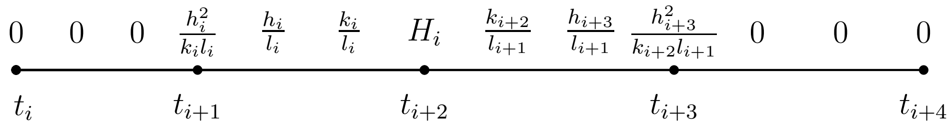


Figure 2.7: Bézier Net for cubic spline $N_{i,4}$.

Parameters h_i, k_i, l_i and H_i are defined by

$$\begin{cases} h_i = t_{i+1} - t_i \\ k_i = h_{i+1} + h_i = t_{i+2} - t_i \\ l_i = h_{i+2} + h_{i+1} + h_i = t_{i+3} - t_i \\ H_i = 1 - \frac{h_{i+2}^2}{k_{i+1} l_i} - \frac{h_{i+1}^2}{k_{i+1} l_{i+1}} \end{cases} \tag{2.1.14}$$

Specifically, if the knots are uniformed with $t_{i+1} - t_i = 1$, the Bézier net becomes one for uniform cubic B-Splines which is illustrated in Figure 2.8.

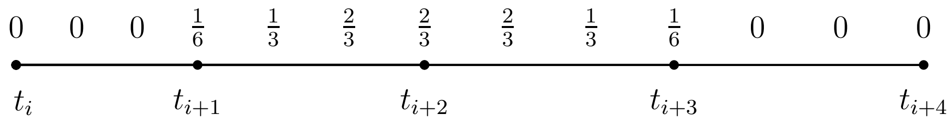


Figure 2.8: Bézier Net for uniform cubic B-Spline $N_{i,4}$.

Bézier nets also give a clearly geometric insight into continuous relations among coefficients. For example, if coefficient sets of two adjacent curve segments jointed at $x = t_i$ are $(p_0, \ldots, p_n)$ and $(q_0, \ldots, q_n)$ (n is the degree), then C^0 at $x = t_i$ forces the last coefficient of left curve segment and the first one of the right curve segment to be identical. This means $p_n = q_0$ (see Figure 2.9).

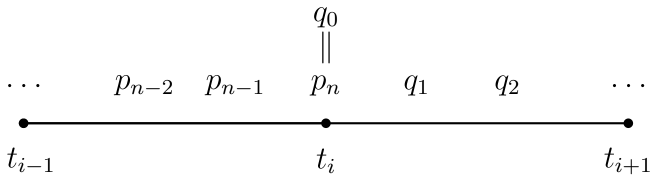


Figure 2.9: Geometric insight of Bézier Net for a B-Spline in C^0.

If the spline has C^1 at $x = t_i$, it must satisfy the following condition by (2.1.7):

$$\frac{n(p_n - p_{n-1})}{h_{i-1}} = \frac{n(q_1 - p_n)}{h_i}. \tag{2.1.15}$$

This tells us that p_{n-1}, p_n and q_1 are collinear if the B-Spline has C^1 at t_i. Figure 2.10 shows such a vision for a B-Spline of degree n in C^1.

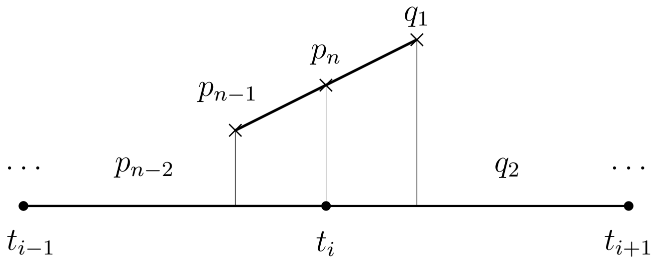


Figure 2.10: Geometric insight of Bézier Net for a B-Spline in C^1.

Similarly, if C^2 is achieved at $x = t_i$, it must satisfy the following condition

$$\frac{n(n-1)(p_{n-2} - 2p_{n-1} + p_n)}{h_{i-1}^2} = \frac{n(n-1)(p_n - 2q_1 + q_2)}{h_i^2}, \tag{2.1.16}$$

and its geometric meaning is shown in Figure 2.11.

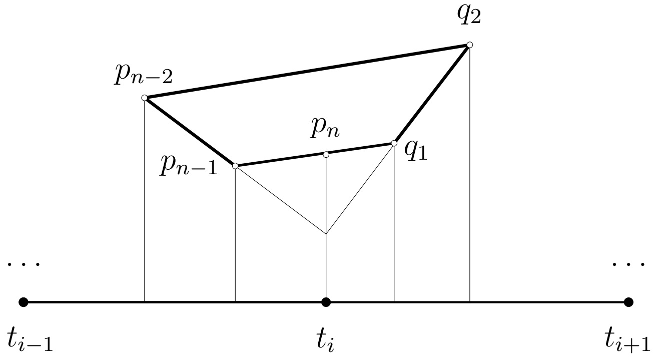


Figure 2.11: Geometric insight of Bézier Net for a B-Spline in C^2.

2.2 NURBs

So far, what we've discussed are B-Spline polynomial curves. Unfortunately, for the curve of $C(t) = \sum_{i=0}^{n} p_i N_{i,m}(t)$, B-Spline polynomial basis functions as weights are not good enough to provide closer approximations for free-form shapes. For example, polynomials can not represent conic shapes such as circles, ellipses and hyperbolas which are important shapes in the industry. In Mathematics, conic curves can be represented by rational functions. This leads to a new kind of curve:

$$C(t) = \sum_{i=0}^{n} p_i \frac{w_i N_{i,m}(t)}{\sum_{j=0}^{n} w_j N_{j,m}(t)} \qquad a \leq t \leq b. \tag{2.2.1}$$

Here w_i are weights, p_i are control points or coefficients, and $N_{i,m}(t)$ are B-Spline basis functions of order m defined at knot sequence $\mathbf{t} = \{a = t_0, \ldots, t_0, t_1, t_2, \ldots, t_n, \ldots, t_n = b\}$. Notice that these NURBs have the identical denominators.

2.2.1 Definition of NURBs

We hence can define NURBs as $R_{i,m}(t)$ in the equation (2.2.1)

$$R_{i,m}(t) = \frac{w_i N_{i,m}}{\sum_{k=0}^{n} w_k N_{k,m}}. \tag{2.2.2}$$

In this way, $C(t)$ in (2.2.1) has the same form as the polynomial one

$$C(t) = \sum_{i=0}^{n} p_i R_{i,m}(t), \qquad t \in [a,b]. \tag{2.2.3}$$

$R_{i,m}(t)$ are called *Nonuniform Rational B-Splines,*or, *NURBs.*

If $w_i = 1$ for all i, $R_{i,m}(t)$ turn into $N_{i,m}(t)$. So B-Spline curves are a special case of NURB curves.

2.2.2 Geometric Interpretation of NURBs

We can use homogeneous coordinates to represent a rational curve in r-dimensional space as a polynomial curve in $(r+1)$-dimensional space [53]. Let $P_i(\frac{x}{w}, \frac{y}{w}, \frac{z}{w}) \in \mathbb{R}^3$ and $P_i^w(x, y, z, w) \in \mathbb{R}^4$ $(x, y, z, w \in \mathbb{R},\ w \neq 0)$, then figure 2.14 shows a rational Bézier curve mapped from a polynomial Bézier curve with three dimensions.

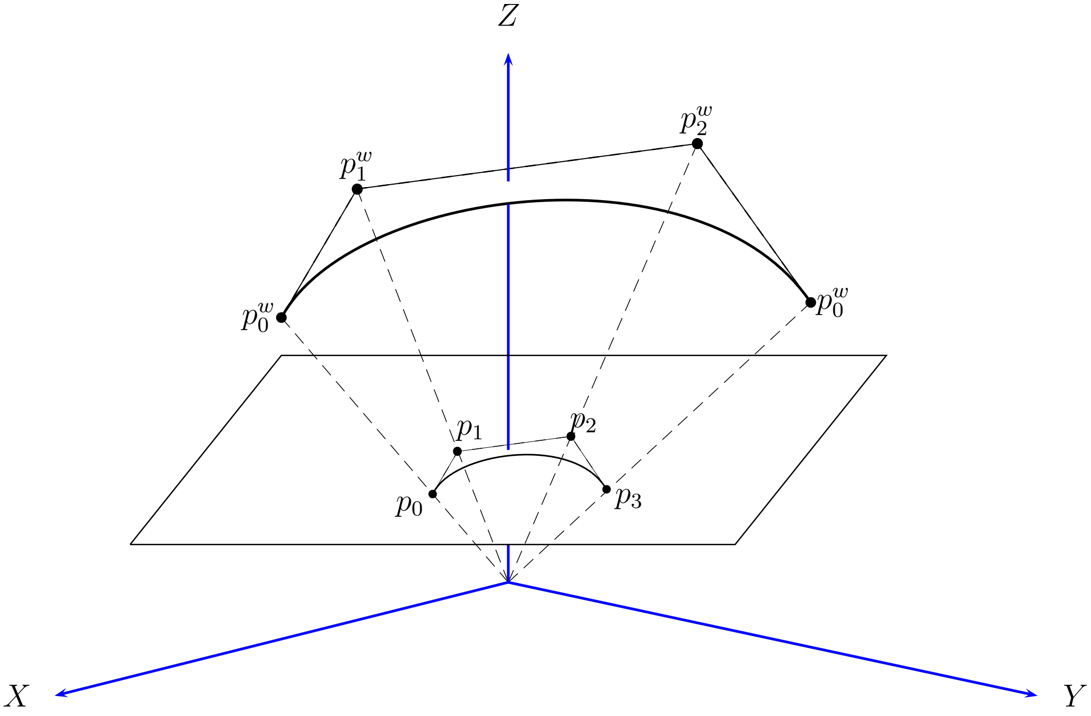


Figure 2.12: Geometric interpretation for a Bézier curve.

2.2.3 The Properties of NURBs

As before, we need to check if rational basis functions have a number of important properties that are desired for spline basis functions.

1. Nonnegativity: $R_{i,m}(t)$ are nonnegative for all i, m and $t \in [t_i, t_{i+m}]$;
2. Local support: $R_{i,m}(t)$ are zero for $t \notin [t_i, t_{i+m}]$;
3. Partition of unity: $\sum_{j=i-m}^{i} R_{j,m}(t) = 1$ for all $t \in [t_i, t_{i+m}]$;
4. Interpolating endpoints: $R_{0,m}(0) = 1$ and $R_{l,m}(1) = 1$;
5. Invariant under affine transformation: $\alpha(\sum_{i=0}^{m-1} p_i R_{i,m}) = \sum_{i=0}^{m-1} \alpha(p_i) R_{i,m}$. Moreover, it is invariant under perspective projection [39].
6. Convex hull property: $C(t)$ lies in the convex hull of the control points $p_0, \ldots, p_l$.
7. Differentiability: $C(t)$ is infinitely differentiable in knot intervals, with a nonzero denominator, and is $m-k$ continuously differentiable at a knot with multiplicity k.

2.3 Multiresolution Analysis (MRA)

Multiresolution analysis is a technique in processing mass data. It offers a hierarchy structure from coarse to fine levels and allows users to retrieve only as much information as need from the original data set.

2.3.1 The Refinement Functions and the Subspaces V_j

At MRA level j, $\phi_j(x)$ are called *refinable functions* if they satisfy the following *refinement relation(condition)* or *two-scale equation*:

$$\phi_j(x) = \sum_k p_{j,k} \phi_{j+1,k}(x), \tag{2.3.1}$$

where $p_{j,k}$ are coefficients for $\phi_{j+1,k}$.

Refinable functions are important to MRA since one can show that spaces spanned by them are nested. Many functions can be refinable functions. Among them, Daubechies refinable functions are classic ones. However, Daubechies refinable functions do not have analytical formula, therefore are not desirable in computer graphics. Cardinal B-Splines are ideal basis functions in the field, and the cubic B-Spline is the most popular basis function.

A multiresolution scheme is based on a sequence of nested subspaces $\{V_j, j \in \mathbb{Z}\}$ of $L^2(R)$. The spaces should be simple, while "big" enough to approximate the given data. To achieve this, we choose a B-Spline function $\phi_0(t) \in L^2(R)$ and its integer shifts $\phi(t-k)$ (denoted as $\phi_{0,k}(x)$) as basis functions for V_0, $\phi_{1,k}(t)$ as the basis functions for V_1, and so on. One can show that since they are spanned by refinement equations, the sequence spaces $\{V_j\}$ is therefore nested. $\{V_j\}$ are called *Multiresolution Analysis*, or, *MRA*, with the following properties [34, 38]:

1. $\ldots \subset V_0 \subset V_1 \subset \ldots \subset V_{n-1} \subset V_n \ldots \subset L^2(R)$;
2. V_j =span$\{\phi_{j,k}(x), k \in \mathbb{Z}\}$, V_{j+1} =span$\{\phi_{j+1,k}, k \in \mathbb{Z}\}$ with (2.3.1) holds;
3. $\bigcup_{j=-\infty}^{\infty} V_j$ is dense in $L^2(R)$ and $\bigcap_{j=-\infty}^{\infty} V_j = 0$.

2.3.2 The Wavelet Functions and the Detail Subspaces W_j

Since $V_0 \subset V_1$, there exists a subspace W_0 of V_1 such that

$$V_1 = V_0 + W_0, \tag{2.3.2}$$

(see figure 2.13). Subspace W_0 contains the fine detail information lost when a curve is transformed from its fine level 1 to coarse level 0. If we let $V_0 = \text{span}\{\phi_{0,i}(x), i \in \mathbb{Z}\}$ and $W_0 = \text{span}\{\psi_{0,i}(x), i \in \mathbb{Z}\}$, then $\phi_{0,i}(x)$ are called *scaling functions*, and $\psi_{0,i}(x)$ *wavelet functions*, or simply, *wavelets*.

Due to the fact $\psi_{0,i}(t) \in V_1$, $\psi_{0,i}(t)$ can be represented by the basis functions of V_1

$$\psi_{0,i}(t) = \sum_{k \in \mathbb{Z}} q_{0,i;k}\phi_{1,k}(t), \tag{2.3.3}$$

$$\begin{tikzcd}
{} \arrow[r] & V_0 \arrow[r] & V_1 \arrow[r] & V_2 \arrow[r] & V_3 \arrow[r, dashed, dash] & {} \\
& \oplus & \oplus & \oplus & & \\
& W_0 \arrow[uur] & W_1 \arrow[uur] & W_2 \arrow[uur] & &
\end{tikzcd}$$

Figure 2.13: Nesting spaces in MRA.

where coefficients $q_{0,i;k}$ are *wavelet coefficients.*

2.3.3 Decomposition and Reconstruction

In computer graphics, we usually study the parametric curves on intervals. Therefore, we use the interval version of the basis functions as in [31, 34, 41, 32, 37]. In V_j, let

$$\Phi_j = [\phi_{j,0}, \phi_{j,1}, ..., \phi_{j,l_j}]^T \tag{2.3.4}$$

be the vectors of basis scaling functions, and $\mathbf{c}_j = [c_{j,0}, c_{j,1}, ..., c_{j,l_j}]^T$ be vectors of control points where $l_j + 1$ are the number of basis functions in V_j, then the curve function can be defined by

$$f_j = \sum_{k=0}^{l_j} c_{j,k}\phi_{j,k} = [c_{j,0}, c_{j,1}, ..., c_{j,l}] \begin{bmatrix} \phi_{j,0} \\ \phi_{j,1} \\ \vdots \\ \phi_{j,l_j} \end{bmatrix} = \mathbf{c}_j{}^T \Phi_j. \tag{2.3.5}$$

Similarly, its coarse level approximation f_{j-1} is

$$f_{j-1} = \sum_{k=0}^{l_{j-1}} c_{j-1,k}\phi_{j-1,k} = \mathbf{c}_{j-1}^T \Phi_{j-1}. \tag{2.3.6}$$

Control points vectors $\mathbf{c}_{j-1}$ can be captured through $\mathbf{c}_j$ by a matrix A_j, known as *analysis matrix,* which will be given in (2.3.17) soon.

$$\mathbf{c}_{j-1} = \mathbf{A}_j \mathbf{c}_j. \tag{2.3.7}$$

Likewise, if g_{j-1} are functions representing the lost detail (i.e., $g_{j-1} \in W_{j-1}$) and

$$\Psi_{j-1} = [\psi_{j-1,0}, \psi_{j-1,1}, ..., \psi_{j-1,l'_{j-1}}]^T \tag{2.3.8}$$

are vectors of basis functions of space W_{j-1} with $l'_{j-1}+1$ the number of wavelet basis functions, and if vectors $\mathbf{d}_{j-1} = [d_{j-1,0}, d_{j-1,1}, ..., d_{j-1,l'_{j-1}}]^T$ capture the detail lost during the transformation from $\mathbf{c}_j$ to $\mathbf{c}_{j-1}$, then we can derive the following relation by (2.3.3):

$$g_{j-1} = \sum_{k=0}^{l'_{j-1}} d_{j-1,k}\psi_{j-1,k} = \mathbf{d}_{j-1}^T \Psi_{j-1}. \tag{2.3.9}$$

Also, $\mathbf{d}_{j-1}$ can be captured through $\mathbf{c}_j$ by another analysis matrix B_j which will be given in (2.3.17):

$$\mathbf{d}_{j-1} = B_j \mathbf{c}_j. \tag{2.3.10}$$

The process of splitting control points $\mathbf{c}_j$ into control points with low resolution $\mathbf{c}_{j-1}$ and the detail $\mathbf{d}_{j-1}$ is known as *decomposition.*

Before we discuss the reconstruction process, we first rewrite the equations (2.3.1) and (2.3.3) in matrix formats

$$\Phi_{j-1}(t) = P_j \Phi_j(t) \tag{2.3.11}$$

and

$$\Psi_{j-1}(t) = Q_j \Phi_j(t), \tag{2.3.12}$$

where P_j, and Q_j are *synthesis matrices.*

Because

$$f_j = f_{j-1} + g_{j-1}, \tag{2.3.13}$$

putting (2.3.6), (2.3.9), (2.3.11), and (2.3.12) in the equation (2.3.13), we get

$$\mathbf{c}_j^T = \mathbf{c}_{j-1}^T P_j + \mathbf{d}_{j-1}^T Q_j. \tag{2.3.14}$$

This shows that coefficients $\mathbf{c}_j$ (i.e., control points in computer graphics) can be recovered from $\mathbf{c}_{j-1}$ and $\mathbf{d}_{j-1}$. The process is called *reconstruction.*

If we consider the relations in (2.3.7) and (2.3.10) in the equation (2.3.14), the following relation holds

$$\mathbf{c}_j{}^T = \mathbf{c}_j^T A_j^T P_j + \mathbf{c}_j^T B_j^T Q_j. \tag{2.3.15}$$

The *Perfect Reconstruction Condition* is therefore derived

$$I = A_j^T P_j + B_j^T Q_j. \tag{2.3.16}$$

Or equivalently,

$$[A_j^T \quad B_j^T] \begin{bmatrix} P_j \\ Q_j \end{bmatrix} = I. \tag{2.3.17}$$

2.3.4 Dyadic MRA

Dyadic refinement is a special case in MRA in which if the original space V_0 =span$\{\phi(x-k), k \in \mathbb{Z}\}$, then V_j =span$\{\phi(2^j x - k), k \in \mathbb{Z}\}$. It is used most widely in applications and we'd like to discuss its properties in MRA in this subsection.

Dyadic refinement relation is:

$$\phi(x) = 2\sum_k p_k \phi(2x-k), \tag{2.3.18}$$

and wavelet function is

$$\psi(x) = 2\sum_k q_k \phi(2x-k). \tag{2.3.19}$$

For any integer $m \geq 1$ as the order of $N(x)$, there exists a sequence $p_{k,m}$ such that

$$N_m(x) = \textstyle\sum_k p_{k,m} N_m(2x-k),$$

where

$$p_{k,n} = 2^{-n+1}\binom{n}{k} \tag{2.3.20}$$

and

$$p_{k,n} = \frac{1}{2}(p_{k,n-1} + p_{k-1,n-1}). \tag{2.3.21}$$

A so called *Pascal's Triangle* describes the relation in (2.3.21). See Figure 2.14.

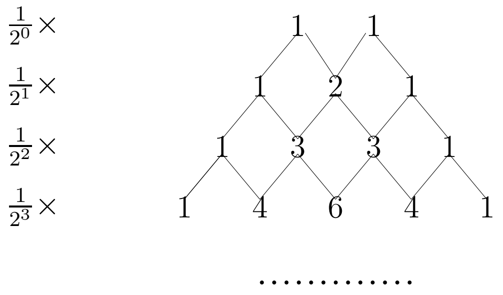

$\frac{1}{2^0}\times$ 1 1
$\frac{1}{2^1}\times$ 1 2 1
$\frac{1}{2^2}\times$ 1 3 3 1
$\frac{1}{2^3}\times$ 1 4 6 4 1
....

Figure 2.14: Pascal triangle.

2.4 Wavelet Analysis

To understand wavelet analysis, let's have a brief review of signal processing. In industry, a raw signal is a function about amplitude on time. But in many cases, frequency content of a signal contains more desirable and important information. So engineers use certain mathematical tool(s) to transform time-amplitude raw signals into frequency-amplitude ones. Among all available tools, Fourier Transform(FT) is the most popular one. However, FT (as well as other available tools) only tells us what frequency components are contained in a raw signal. It cannot tell when these components exist. If two raw signals have same frequency components but at different time intervals, FT won't be able to tell the difference! So FT is a good tool only when the frequency content of a signal does not change in time. Such a signal is called a *stationary signal.*

In real world, most of signals are not stationary. So we need some other tool(s) for getting frequency-time signals. However, *Heisenberg's Uncertainty Principle* breaks our dream. It states that the values of time and frequency cannot both be known with arbitrary precision simultaneously. In other words, it is impossible to exactly know what frequency happens at what time instant. What we can know is what frequency bands happen at what time intervals. To make a tradeoff, if these time intervals are reasonably small enough, we can achieve our goal approximately. Technically, we divide a signal into small enough segments (by a window function) such that each signal segment can be supposed to be stationary and consequently, then FT can be applied. Such technique is called *short time Fourier Transform (STFT).*

The width of window function in STFT is called the *support* of the window. But STFT has a resolution dilemma here: we wish the support is small, but it leads to poor frequency resolution; if the support is big, it not only violates the condition of stationarity, but also leads to poor time resolution. The solution to it is wavelet transform (WT). The big difference between STFT and WT is that in WT, the support of window varies with frequency components, while in STFT it keeps the same. A continuous wavelet transform of a function $f(x) \in L^2(R)$ is defined as:

$$\gamma(s,\tau) = \langle f(x), \psi_{s,\tau}(x)\rangle = \int_{-\infty}^{\infty} f(x)\overline{\psi_{s,\tau}(x)}dx, \tag{2.4.1}$$

where s and τ are variables of scale and translation. $\psi_{s,\tau}(x)$ is called *mother wavelet* and is defined as:

$$\psi_{s,\tau}(x) = \frac{1}{\sqrt{|s|}}\psi(\frac{x-\tau}{s}) \quad \text{with } s,\tau \in \mathbb{R},\ s \neq 0. \tag{2.4.2}$$

2.4.1 Wavelet Properties

Wavelet $\psi(t)$ should satisfy so-called *Admissibility condition*

$$\int_{-\infty}^{\infty} \frac{|\Psi(\omega)|^2}{\omega} d\omega < \infty, \tag{2.4.3}$$

where $\Psi(\omega)$ is FT of $\psi(t)$. The condition implies that the FT of $\psi(t)$ vanishes at zero frequency

$$|\Psi(\omega)|^2_{\omega=0} = 0. \tag{2.4.4}$$

This means that wavelets must have a band-pass like spectrum, and also means

$$\int_{-\infty}^{\infty} \psi(t)dt = 0. \tag{2.4.5}$$

One of the very important characters of wavelets is vanishing moments. If we expand the wavelets transform in formula (2.4.1) into Taylor series at $\tau = 0$, we then have

$$\gamma(s,0) = \int \sum_{p=0}^{n} \frac{f^{(p)}_{(0)}}{p!} t^p \frac{1}{\sqrt{s}} \psi(\frac{t}{s})dt + O(n+1) = \frac{1}{\sqrt{s}}(\sum_{p=0}^{n} \frac{f^{(p)}_{(0)}}{p!} \int t^p \psi(\frac{t}{s})dt) + O(n+1).$$

If we let

$$M_p = \int t^p \psi(t) dt, \tag{2.4.6}$$

then

$$\gamma(s,0) = (\frac{1}{\sqrt{s}} M_0 S + f_{(0)}^{(1)} M_1 S^2 + \frac{1}{2!} f_{(0)}^{(2)} M_2 S^3 + \ldots + \frac{1}{n!} f_{(0)}^{(n)} M_n S^{n+1}) + O(S^{n+2}),$$

i.e.,

$$\gamma(s,0) == \frac{1}{\sqrt{s}} \sum_{j=0}^{n} \frac{1}{j!} f_{(0)}^{(j)} M_j S^{j+1} + O(S^{n+2}). \tag{2.4.7}$$

We have $M_0 = 0$ because of admissibility condition. If we let $M_n = 0$ (or be vanishing), then $M_j = 0, \forall j = 0, 1, \cdots, n$. Thus $\gamma(s,0) = O(S^{n+2})$. This makes the wavelet transformation coefficients $\gamma(s,\tau)$ decay as fast as $O(S^{n+2})$ for a smooth function $f(t)$. Because of the reason, M_n is thus called n *moments.*

Vanishing moments play an important role in wavelet construction. If a wavelet has n vanishing moments, then n approximation order of the wavelet transformation can be achieved.

2.4.2 Categorization of Wavelets in MRA

As we have mentioned in Section 2.3.2, wavelets $\psi_j(x-k)$ span space W_j which contains fine details lost in the transformation from fine level V_{j+1} to coarse level V_j. According to the different relations between V_j and W_j, there are following types of wavelets.

Orthogonal Wavelets An orthogonal wavelet is a function $\psi(x)$ such that $\{\psi(x-k), k \in \mathbb{Z}\}$ is an orthonormal basis, i.e.,

$$\langle \psi(x), \psi(x-k) \rangle = \delta_k, \quad k \in \mathbb{Z}. \tag{2.4.8}$$

Notice that each orthogonal wavelet function is also orthogonal to scaling functions at the same level.

Semiorthogonal Wavelets A semiorthogonal wavelet is a function $\psi(x)$ such that $\psi(x-k), k \in \mathbb{Z}$ are not orthogonal to each other, yet each wavelet function is orthogonal to scaling functions at the same level.

Biorthogonal Wavelets A biorthogonal wavelet transformation is invertible but wavelets are not necessarily orthogonal. Suppose in $L^2(R)$, $\widetilde{V_j}$ is the dual space of V_j, and $\tilde{\phi}_{j,k}(x)$ and $\phi_{j,k}(x)$ are their dual scaling functions and scaling functions respectively. Correspondingly, $\psi_{j,k'}(x)$ and $\tilde{\psi}_{j,k'}(x)$ span wavelet spaces W_j and $\widetilde{W_j}$. Wavelet function $\psi(x)$ is called *biorthogonal wavelet* if

$$V_j \perp \widetilde{W_j} \quad and \quad W_j \perp \widetilde{V_j}.$$

Or equivalently,

$$\begin{array}{lll} \langle \phi_{j,k}, \tilde{\psi}_{j,l} \rangle = 0, & \langle \psi_{j,k}, \tilde{\phi}_{j,l} \rangle = 0 & \text{biorthogonal condition} \\ \langle \tilde{\phi}_{j,k}, \phi_{j,l} \rangle = \delta_{k,l}, & \langle \tilde{\psi}_{j,k}, \psi_{j,l} \rangle = \delta_{k,l} & \text{dual condition} \end{array}$$

In matrix format, it means

$$\begin{bmatrix} \widetilde{P} & \widetilde{Q} \end{bmatrix} \begin{bmatrix} P \\ Q \end{bmatrix} = I. \tag{2.4.9}$$

Biorthogonal wavelets have more flexible freedom in construction; Moreover, they are invertible. So they are widely applied. In this paper, we'll construct biorthogonal wavelets based on NURBs as scaling functions.

Wavelet frame Wavelet frames are wavelets when the wavelet functions in space W_j are not necessarily orthogonal, neither space V_j orthogonal to W_j. Wavelet frames give the most freedom in constructing wavelets in MRA, and therefore become more and more important in applications.

Chapter 3

Natural Cubic B-Splines with Arbitrary Knots and Two–Scale Matrices

In the industry, many applications require the curvature to be continuous. This requires our B-Spline basis polynomials should be at least of degree three. In fact, cubic B-Splines are the most widely used ones in applications. In our model, the scaling functions are cubic NURBs with natural cubic B-Splines at the boundaries. We will investigate natural cubic B-Splines in this chapter. Based on natural cubic B-Spline bases, we'll build edge functions, and finally two–scale matrices are given.

3.1 Natural Cubic B-Spline Bases

Natural cubic B-Splines are cubic splines whose second derivatives at the two endpoints are zero. It is well known that among interpolating cubic splines, the shapes of natural cubic B-Splines have the minimum strain energy.

To build the natural cubic B-Splines, let's suppose that a spline interpolates $n + 1$ control points

$$(t_0, y_0), (t_1, y_1), (t_2, y_2), \ldots, (t_n, y_n),$$

with a piecewise cubic polynomial

$$S(x) = \begin{cases} S_1(x) & t_0 \leq x \leq t_1 \\ S_2(x) & t_1 \leq x \leq t_2 \\ \vdots & \\ S_n(x) & t_{n-1} \leq x \leq t_n \end{cases}$$

where

$$S_i(x) = A_i + B_i(x - t_i) + C_i(x - t_i)^2 + D_i(x - t_i)^3 \quad \text{for i=1 , \ldots , } n. \tag{3.1.1}$$

Each of these n segments $S_i(x)$ is a cubic polynomial determined by four coefficients (A_i, B_i, C_i and D_i). There are totally $4n$ coefficients to be determined. We wish our spline to have C^2 smoothness. Therefore, at each of $n - 1$ interior control points, the spline would interpolate the point, and it's first and second derivatives are continuous:

$$\begin{aligned} S_i(t_i) &= y_i, & S_{i+1}(t_i) &= y_i \\ S_i'(t_i) &= S_{i+1}'(t_i), & S_i''(t_i) &= S_{i+1}''(t_i) \qquad \text{for } i = 1, 2, \ldots, n-1. \end{aligned} \tag{3.1.2}$$

At two endpoints, the spline interpolates endpoints and its second derivatives are zero:

$$\begin{aligned} S_0(t_0) &= y_0, \quad S_n(t_n) = y_n \\ S_0''(t_0) &= 0, \quad S_n''(t_n) = 0. \end{aligned} \tag{3.1.3}$$

(3.1.2) and (3.1.3) give us total $4(n - 1) + 4 = 4n$ equations for determining $4n$ coefficients. Let $z_i = S''(x_i)$, and $h_i = t_{i+1} - t_i, i \in [0, n - 1]$. Since each segment polynomial is cubic, the second derivative is a linear function in $[t_i, t_{i+1}]$, i.e., $S_i''(x) = \frac{z_{i+1} - z_i}{h_i}(x - t_i) + z_i$. Solving the partial differential equation, we can get z_i from the following equation:

$$\begin{bmatrix} 2(h_0 + h_1) & h_1 & & \\ h_1 & 2(h_1 + h_2) & \ldots & \\ & \ldots & \ldots & h_{n-2} \\ & & h_{n-2} & 2(h_{n-2} + h_{n-1}) \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_{n-1} \end{bmatrix} = 6 \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_{i-1} \end{bmatrix} \tag{3.1.4}$$

where $v_i = 6(\frac{y_{i+1}-y_i}{h_i} - \frac{y_i - y_{i-1}}{h_{i-1}})$.

Finally, we get piecewise polynomial for each segment natural cubic B-Spline:

$$S_i(x) = \frac{z_{i+1}}{6h_i}(x-t_i)^3 + \frac{z_i}{6h_i}(t_{i+1}-x)^3 + (\frac{y_{i+1}}{h_i} - \frac{h_i}{6}z_{i+1})(x-t_i) + (\frac{y_i}{h_i} - \frac{h_i}{6}z_i)(t_{i+1}-x). \quad (3.1.5)$$

For example, if we have a set of knots which are equally spaced by 1 such that control points are $(0, y_0)$, $(1, y_1)$,..., (n, y_n), then $h_i = t_{i+1} - t_i = 1, i \in [0, n-1]$. The equation for z_i is

$$\begin{bmatrix} 4 & 1 & 0 & & \\ 1 & 4 & 1 & & \\ & 1 & \ddots & 1 & \\ & & 1 & 4 & 1 \\ & & & 1 & 4 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_{n-2} \\ z_{n-1} \end{bmatrix} = 6 \begin{bmatrix} y_2 - 2y_1 + y_0 \\ y_3 - 2y_2 + y_1 \\ \vdots \\ y_{n-1} - 2y_{n-2} + y_{n-3} \\ y_n - 2y_{n-1} + y_{n-2} \end{bmatrix}, \quad (3.1.6)$$

and the segment polynomial $S_i(x)$ is

$$S_i(x) = \frac{z_{i+1}}{6}(x-t_i)^3 + \frac{z_i}{6}(t_{i+1}-x)^3 + (y_{i+1} - \frac{z_{i+1}}{6})(x-t_i) + (y_i - \frac{z_i}{6})(t_{i+1}-x). \quad (3.1.7)$$

3.1.1 Natural Cubic B-Splines at Boundaries

We are interested in constructing edge functions with natural cubic B-Splines. Without lose of generality, we only consider the edge functions at the left boundary. There are two questions we have to answer: How many edge functions do we need at the boundary and what are they?

To answer the first question, suppose we have regular cubic B-Splines as edge functions. Since a regular cubic B-Spline covers four knot spans, there are at most three edge functions which cover the first one, two and three knot spans at the boundary, respectively. Now we make them 'natural'. This means their linear combination together with one interior cubic B-Spline should satisfy the condition that the second derivatives at the endpoint are zero. It turns out the edge function covering the first knot span is gone. In fact, with natural splines of any order as edge functions, the number of B-Splines on the interval is equal to the number of knots (including both interior and boundary knots). Based on this observation, we need two cubic natural edge functions at the boundary.

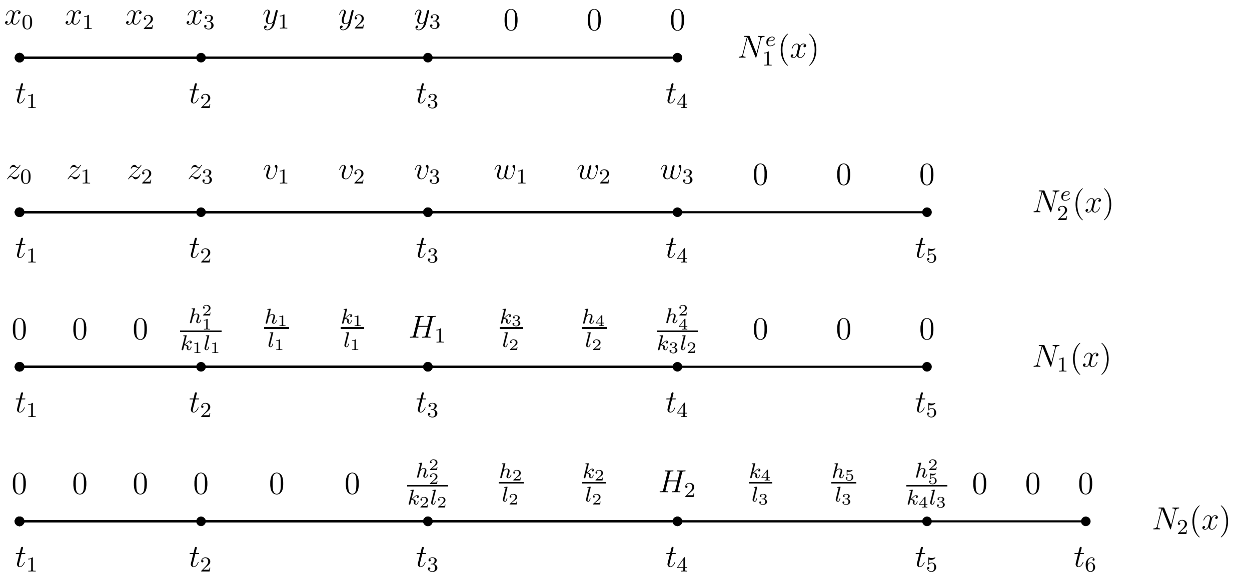

x_0 x_1 x_2 x_3 y_1 y_2 y_3 0 0 0
$N_1^e(x)$
t_1 t_2 t_3 t_4
z_0 z_1 z_2 z_3 v_1 v_2 v_3 w_1 w_2 w_3 0 0 0
$N_2^e(x)$
t_1 t_2 t_3 t_4 t_5
0 0 0 $\frac{h_1^2}{k_1 l_1}$ $\frac{h_1}{l_1}$ $\frac{k_1}{l_1}$ H_1 $\frac{k_3}{l_2}$ $\frac{h_4}{l_2}$ $\frac{h_4^2}{k_3 l_2}$ 0 0 0
$N_1(x)$
t_1 t_2 t_3 t_4 t_5
0 0 0 0 0 0 $\frac{h_2^2}{k_2 l_2}$ $\frac{h_2}{l_2}$ $\frac{k_2}{l_2}$ H_2 $\frac{k_4}{l_3}$ $\frac{h_5}{l_3}$ $\frac{h_5^2}{k_4 l_3}$ 0 0 0
$N_2(x)$
t_1 t_2 t_3 t_4 t_5 t_6

Figure 3.1:Bézier nets at the left boundary.

We now find these two edge functions. Given a knot vector $\mathbf{t} = \{t_1, \ldots, t_6\}$, let $N_1^e(x)$ and $N_2^e(x)$ be unknown natural cubic edge functions, and $N_1(x)$ and $N_2(x)$ be cubic B-Splines on interval $[t_1, t_5]$ and $[t_2, t_6]$ respectively. By Figure 2.7, the Bézier nets of these functions are shown in Figure 3.1, where h_i, k_i, l_i are defined by (2.1.14), and x_i, y_i, z_i, v_i and w_i are unknown coefficients of edge functions which we'll find out.

We wish $N_1^e(x)$ have C^1 and C^2 at $t = t_2$ and $t = t_3$, and $(N_1^e)^{(2)}(t_1) = 0$ because of its natural property. Considering (2.1.7), the following equations are derived

$$\begin{cases} y_3 = 0 \\ \frac{3(y_3 - y_2)}{h_2} = \frac{-3y_3}{h_3} \\ \frac{6(y_1 - 2y_2 + y_3)}{(h_2)^2} = \frac{6y_3}{(h_3)^2} \\ \frac{3(x_3 - x_2)}{h_1} = \frac{3(y_1 - x_3)}{h_2} \\ \frac{6(x_1 - 2x_2 + x_3)}{(h_1)^2} = \frac{6(x_3 - 2y_1 + y_2)}{(h_2)^2} \\ \frac{6(x_0 - 2x_1 + x_2)}{(h_1)^2} = 0 \end{cases} \tag{3.1.8}$$

An immediate observation from above is $y_1 = y_2 = y_3 = 0$. By the Partition of Unity, at any knot t_k the values of all basis functions should sum to 1. Hence in Figure 3.1, the following results hold

$$\begin{cases} w_1 = w_2 = w_3 = 0 \\ v_1 = \frac{k_2}{l_1}, v_2 = \frac{h_3}{l_1}, v_3 = \frac{h_3^2}{k_2 l_1} \end{cases} \tag{3.1.9}$$

Similarly, we could build a group of equations for coefficients of $N_2^e(x)$ at t_1 and t_2

$$\begin{cases} \frac{6(z_0 - 2z_1 + z_2)}{h_1^2} = 0 \\ \frac{3(z_3 - z_2)}{h_1} = \frac{3(v_1 - z_3)}{h_2} \\ \frac{6(z_1 - 2z_2 + z_3)}{(h_1)^2} = \frac{6(z_3 - 2v_1 + v_2)}{(h_2)^2} \end{cases} \tag{3.1.10}$$

Solving the groups of equations (3.1.8)-(3.1.10), we get the Bézier nets for natural cubic B-Splines at the boundary in Figure 3.2.

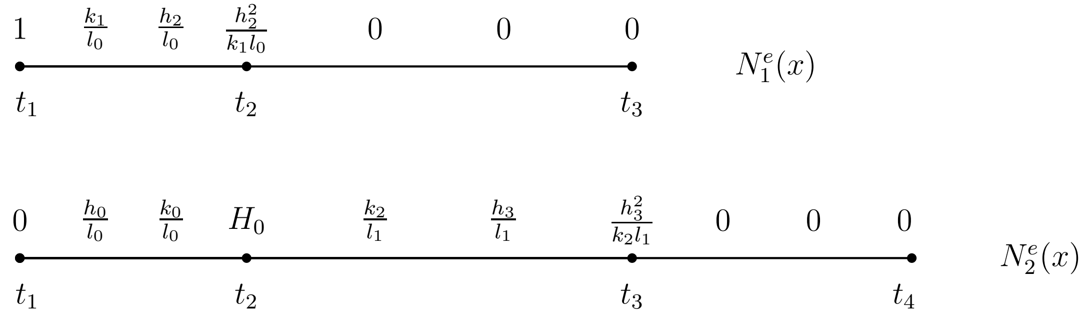


Figure 3.2: Natural cubic B-Splines at the boundary.

Specifically, if knots are uniformed with $t_{i+1} - t_i = 1$, the Bézier nets for cubic natural splines at the boundary become what is in Figure 3.3.

3.2 Two–Scale Matrices

3.2.1 Knot Insertion

As we've already discussed, we can construct a piecewise function as a B-Spline given control points with breakpoints at the knots. It is understandable that with more control points, the control polygon will get closer to the curve, and the B-Spline approximation will therefore

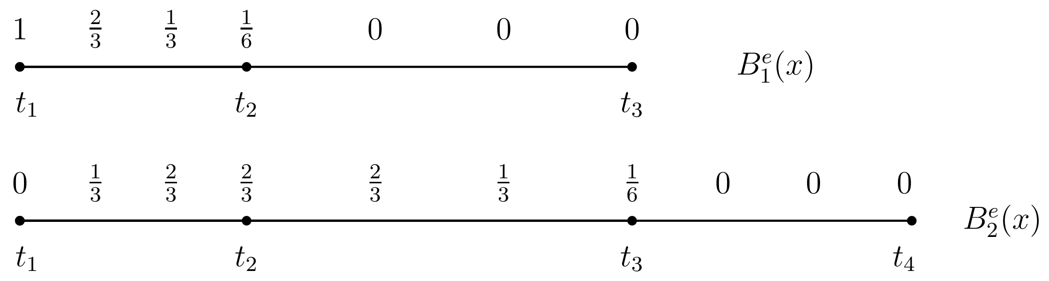


Figure 3.3: Bézier nets for uniform natural cubic B-Splines.

be better. Thus for the complicated part of a curve, we can get better approximation by adding more control points there. Noticing that inserting a knot in knot sequence leads to one more control point, we want to devise an algorithm such that when a knot is inserted, only the neighboring control points are updated and their number increases by one, while the rest control points are unchanged and the shape of curve remains the same.

Initially, we are interested in inserting one arbitrary knot between at any knot span. Given a knot sequence of $s+1$ knots $\mathbf{t} = \{t_0, t_1, \ldots, t_s\}$, and $l+1$ control points $(p_0, p_1, \ldots, p_l)$ with a degree of n, we let B-Spline basis functions be $B_{i,n}(x)$, $i = 0, 1, \ldots, n$. Suppose a new knot τ is inserted such that $\tau \in (t_r, t_{r+1})$. In the interval (t_r, t_{r+1}) only basis functions for control points $p_{r-n}, \ldots, p_r$ are nonzero, therefore only these control points are affected by the new inserted knot and the rest remains unchanged. After the insertion, the new curve has $s+2$ knots $\mathbf{t}^1 = \{t_0, t_1, \ldots, t_r, \tau, t_{r+1}, \ldots, t_s\}$ and $l+2$ new control points $(p_0^1, p_1^1, \ldots, p_{l+1}^1)$. Böehm [10] gives an algorithm for one knot insertion in the following:

$$p_i^1 = \begin{cases} p_i & i = 0, 1, \ldots, r-n+1 \\ (1-\alpha_i^1)p_{i-1} + \alpha_i^1 p_i & i = r-n+2, \ldots, r \\ p_{i-1} & i = r+1, \ldots, l+1 \end{cases} \tag{3.2.1}$$

where

$$\alpha_i^1 = \frac{\tau - t_i}{t_{i+k} - t_i} \qquad i = r-k+2, \ldots, r. \tag{3.2.2}$$

Figure 3.4 shows the diagram of one-knot insertion for cubic basis functions.

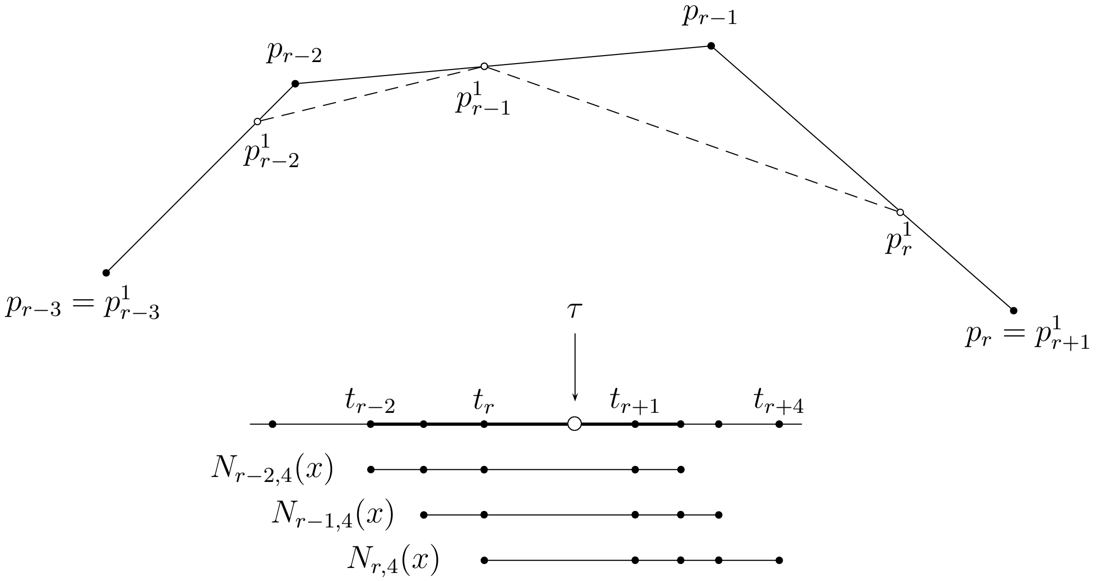


Figure 3.4: One knot insertion for cubic basis functions: two coefficients p_{r-2} and p_{r-1} have been replaced by three new ones p^1_{r-2}, p^1_{r-1}, and p^1_r.

3.2.2 Subdivision Templates

Now the question is how to find the new set of coefficients after the knot insertion. We consider the case of cubic B-Splines. Suppose on $[t_i, t_{i+1}]$ the control points are (p_0, p_1, p_2, p_3). When a new knot τ is inserted, it cuts the segment curve on the knot span into two sub-curve segments which are on $[t_i, \tau]$ and $[\tau, t_{j+1}]$ respectively. To find these two sets of coefficients, we apply de Casteljau's algorithm, but in barycentric system.

Let

$$u = \frac{\tau - t_i}{t_{i+1} - t_i}, \tag{3.2.3}$$

de Casteljau's triangle is demonstrated in Figure 3.5.

A simpler diagram is given in Figure 3.6. In a clearer way, it discloses the relations among knots at different levels. First, $f(\tau) = p^3_0$. Secondly, two sets of new coefficients on $[t_i, \tau]$ and $[\tau, t_{i+1}]$ are $(p_0, p^1_0, p^2_0, p^3_0)$ and $(p^3_0, p^2_1, p^1_2, p_3)$. Notice that

$$p^k_j = \sum_{i=0}^{k} p_{i+j}(1-u)^{k-i}u^i, \quad k = 1, 2, 3, \;\; j = 0, 1, \ldots, 3-k. \tag{3.2.4}$$

The corresponding Bézier nets in Figure 3.7 render an insight into two new sets of coefficients in $[t_i, \tau]$ and $[\tau, t_{i+1}]$.

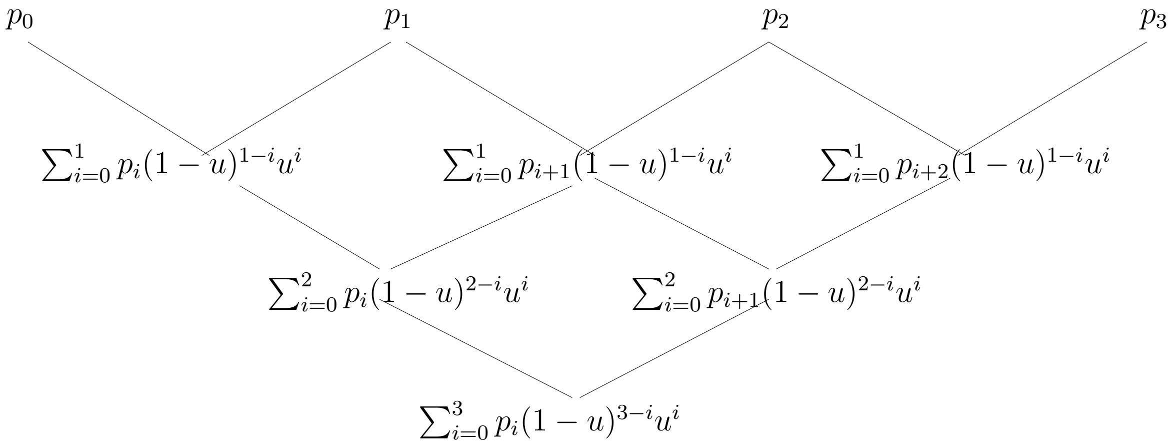

p_0
p_1
p_2
p_3
$\sum_{i=0}^{1} p_i(1-u)^{1-i}u^i$
$\sum_{i=0}^{1} p_{i+1}(1-u)^{1-i}u^i$
$\sum_{i=0}^{1} p_{i+2}(1-u)^{1-i}u^i$
$\sum_{i=0}^{2} p_i(1-u)^{2-i}u^i$
$\sum_{i=0}^{2} p_{i+1}(1-u)^{2-i}u^i$
$\sum_{i=0}^{3} p_i(1-u)^{3-i}u^i$

Figure 3.5: de Casteljau's algorithm in barycentric system (1).

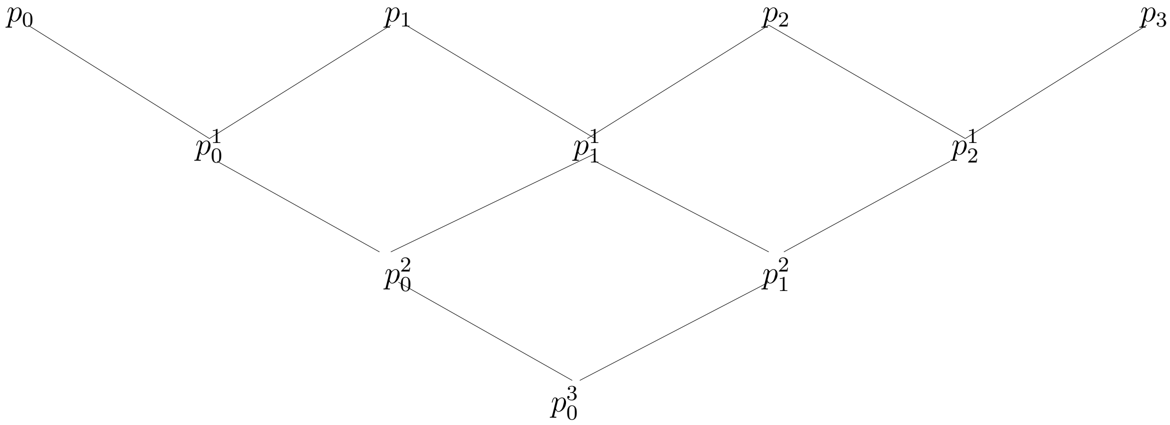


Figure 3.6: de Casteljau's algorithm in Barycentric System (2); p_j^k are defined in (3.2.4).

In Figure 3.7, the upper diagram is the cubic Bézier net on interval $[t_i, t_{i+1}]$ before the knot insertion. The set of its coefficients is (p_0, p_1, p_2, p_3). The lower diagram is one after inserting a new knot τ is inserted in the knot span. The new knot results in two sets of new coefficients $(p_0, p_0^1, p_0^2, p_0^3)$ and $(p_0^3, p_1^2, p_2^1, p_3)$ on knot span $[t_i, u]$ and $[u, t_{i+1}]$, with p_j^k are defined in (3.2.4).

3.2.3 Two-Scale Matrices

In MRA, let vectors of scaling functions be

$$\Phi_0(x) = \begin{pmatrix} N_{0,1}^e(x) & N_{0,2}^e(x) & N_{0,1}(x) & N_{0,2}(x) & N_{0,3}(x) & \ldots \end{pmatrix}^T \qquad (3.2.5)$$

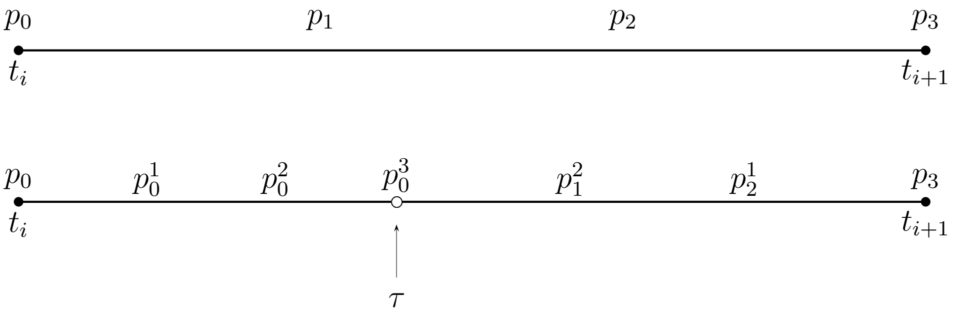


Figure 3.7: Bézier nets before and after one knot insertion on $[t_i, t_{i+1}]$.

and

$$\Phi_1(x) = \begin{pmatrix} N^e_{1,1}(x) & N^e_{1,2}(x) & N_{1,1}(x) & N_{1,2}(x) & N_{1,3}(x) & \dots \end{pmatrix}^T . \tag{3.2.6}$$

then they satisfy the following refinement relation

$$\Phi_0(x) = P_0 \Phi_1(x). \tag{3.2.7}$$

where P_0 is two–scale matrix. In this section, we will find two–scale matrices when a new knot is inserted on different knot spans.

First we consider the case of $\tau \in (t_1, t_2)$ where τ is a new knot. In this case, three basis functions are influenced: $N^e_{0,1}(x)$, $N^e_{0,2}(x)$ and $N_{0,1}(x)$. We denote two–scale matrix as $P_{0,[t_1,t_2]}$.

As usual, on interval $[t_1, t_s]$ let cubic B-Spines be $N_{i,4}(x), i \in [1, s-4]$, let natural cubic B-Splines $N^e_{0,1}(x)$ and $N^e_{0,2}(x)$ be edge functions at the left boundary, and $N^e_{0,s-3}(x)$ and $N^e_{0,s-2}(x)$ be edge ones at the right boundary.

For the purpose of computation, we virtually extend the boundary knots to the left as it is shown in Figure 3.8.

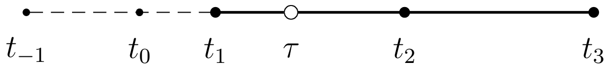


Figure 3.8: Extend the boundary by adding two virtual knots t_0 and t_{-1} such that $t_1 - t_0 = \tau - t_1$ and $t_0 - t_{-1} = t_2 - \tau$.

Figure 3.2 shows the Bézier net of $N^e_{0,1}$ defined on knot sequence $\{t_1, \dots\}$. However, insert-

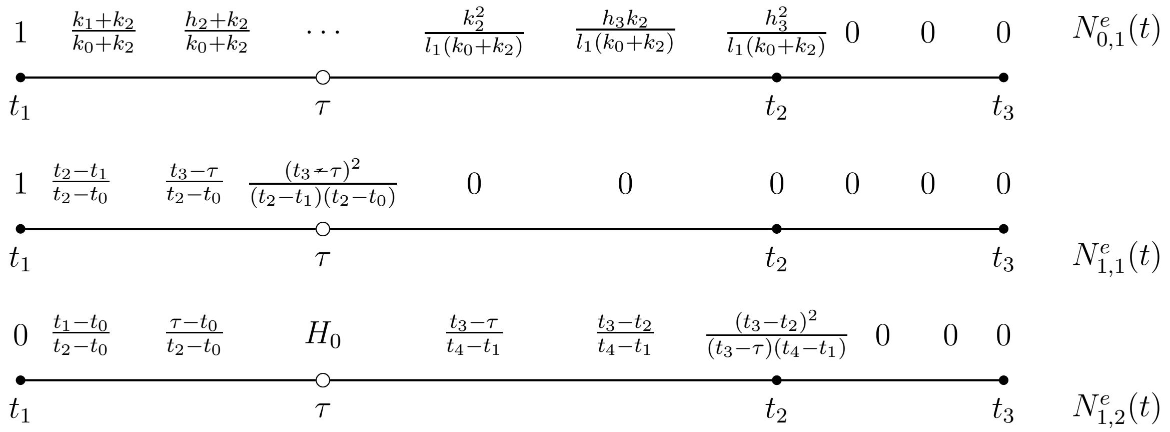

1
$\frac{k_1+k_2}{k_0+k_2}$
$\frac{h_2+k_2}{k_0+k_2}$
$\cdots$
$\frac{k_2^2}{l_1(k_0+k_2)}$
$\frac{h_3k_2}{l_1(k_0+k_2)}$
$\frac{h_3^2}{l_1(k_0+k_2)}$
0
0
0
$N^e_{0,1}(t)$
t_1
τ
t_2
t_3
1
$\frac{t_2-t_1}{t_2-t_0}$
$\frac{t_3-\tau}{t_2-t_0}$
$\frac{(t_3-\tau)^2}{(t_2-t_1)(t_2-t_0)}$
0
0
0
0
0
0
t_1
τ
t_2
t_3
$N^e_{1,1}(t)$
0
$\frac{t_1-t_0}{t_2-t_0}$
$\frac{\tau-t_0}{t_2-t_0}$
H_0
$\frac{t_3-\tau}{t_4-t_1}$
$\frac{t_3-t_2}{t_4-t_1}$
$\frac{(t_3-t_2)^2}{(t_3-\tau)(t_4-t_1)}$
0
0
0
t_1
τ
t_2
t_3
$N^e_{1,2}(t)$

Figure 3.9: Bézier nets of $N^e_{0,1}(x)$ and $N^e_{1,1}(x)$ with one knot insertion on (t_1, t_2).

ing a new knot τ on (t_1, t_2) results in a new knot sequence on the same interval, which results in a new Bézier net for the same function $N^e_{0,1}$. Figure 3.9 shows the equivalent Bézier net of $N^e_{0,1}$ defined on new knot sequence $\{t_0, \tau, t_1, \ldots\}$ (See the top net in Figure 3.9). Consequently, $N^e_{0,1}(x)$ satisfies the following refinement relation:

$$N^e_{0,1}(t) = AN^e_{1,1}(t) + BN^e_{1,2}(t),$$

where A and B are certain coefficients. Applying the partition of unity at knots t_1, τ and t_2, we find solutions for A and B

$$N^e_{0,1}(t) = N^e_{1,1}(t) + \frac{t_3 - \tau}{t_3 - t_0} N^e_{1,2}(t).$$

Similarly, we can find the refinement relations for $N^e_{0,2}(x)$ and $N_{0,1}(x)$ as well.

Finally, if we define

$$\alpha_i = \frac{\tau - t_i}{t_{i+3} - t_i} \quad i = 0, 1, \tag{3.2.8}$$

the two–scale matrix $\mathbf{P_0}$ for one knot insertion on $[t_1, t_2]$ is derived as following

$$\mathbf{P_{[t1,t2]}} = \begin{bmatrix} 1 & 1-\alpha_o & & & & \\ & \alpha_0 & 1-\alpha_1 & & & \\ & & \alpha_1 & 1 & & \\ & & & & 1 & \\ & & & & & \ddots \end{bmatrix}. \tag{3.2.9}$$

Likewise, if the knot is inserted on $[t_2, t_3]$, the two–scale matrix becomes

$$\mathbf{P_{[t2,t3]}} = \begin{bmatrix} 1 & 1-\alpha_o & & & & & \\ & \alpha_0 & 1-\alpha_1 & & & & \\ & & \alpha_1 & 1-\alpha_2 & & & \\ & & & \alpha_2 & 1 & & \\ & & & & & 1 & \\ & & & & & & \ddots \end{bmatrix}. \tag{3.2.10}$$

If the knot is inserted not in the boundary area, i.e., $\tau \in [t_i, t_{i+1}]$, where $3 \le i \le s-3$, the two–scale relation is

$$\begin{bmatrix} N_{0,i-3}(t) \\ N_{0,i-2}(t) \\ N_{0,i-1}(t) \\ N_{0,i}(t) \end{bmatrix} = \begin{bmatrix} 1 & 1-\alpha_{i-2} & & & \\ & \alpha_{i-2} & 1-\alpha_{i-1} & & \\ & & \alpha_{i-1} & 1-\alpha_i & \\ & & & \alpha_i & 1 \end{bmatrix} \begin{bmatrix} N_{1,i-3}(t) \\ N_{1,i-2}(t) \\ N_{1,i-1}(t) \\ N_{1,i}(t) \\ N_{1,i+1}(t) \end{bmatrix}. \tag{3.2.11}$$

We finally attack the two–scale matrix with arbitrary knot insertion in each knot span. The insertion procedure is described in Figure 3.10. Notice that for the sake of computation, we virtually extend the boundary left to knot t_0.

Using the same technique we have for one-knot insertion, after a series of computation draft, we get the two–scale matrix for the arbitrary knot insertion in each knot span as the

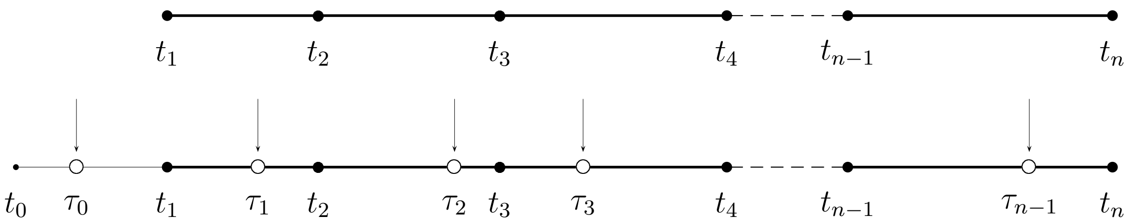


Figure 3.10: Arbitrary knot insertion at each knot span. Inserted knots are $\tau_1, \ldots, \tau_{n-1}$. Virtually extend the left boundary by adding two virtual knots t_0 and τ_0 such that $t_0 = 2t_1 - t_2$ and $\tau_0 = 2t_1 - \tau_1$.

following

$$\mathbf{P} = \begin{bmatrix} 1 & 1-\alpha_0 & \beta_1 & & & & & \\ & \alpha_0 & 1-\beta_1-\gamma_1 & 1-\alpha_1 & \beta_2 & & & \\ & & \gamma_1 & \alpha_1 & 1-\beta_2-\gamma_2 & 1-\alpha_2 & \beta_3 & \\ & & & & \gamma_2 & \alpha_2 & 1-\beta_3-\gamma_3 & \\ & & & & & & \gamma_3 & \\ & & & & & & & \ddots \end{bmatrix}, \quad (3.2.12)$$

where

$$\alpha_i = \frac{\tau_{i+1} - t_i}{t_{i+3} - t_i} \quad (3.2.13)$$

$$\beta_i = \frac{t_{i+2} - \tau_{i+1}}{t_{i+2} - t_i}(1 - \alpha_{i-1}) \quad (3.2.14)$$

$$\gamma_i = \frac{\tau_i - t_i}{t_{i+2} - t_i}\alpha_i. \quad (3.2.15)$$

For example, if we insert a middle point on each of the uniform interval, then the two–scale matrix in (3.2.12) becomes

$$\mathbf{P} = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{8} & & & & & & \\ & \frac{1}{2} & \frac{6}{8} & \frac{1}{2} & \frac{1}{8} & & & & \\ & & \frac{1}{8} & \frac{4}{8} & \frac{6}{8} & \frac{4}{8} & \frac{1}{8} & & \\ & & & & \frac{1}{8} & \frac{4}{8} & \frac{6}{8} & \frac{4}{8} & \frac{1}{8} \\ & & & & & & \ddots & & \end{bmatrix}. \tag{3.2.16}$$

This is the two–scale matrix of middle-point insertion for uniform cubic B-Splines with natural cubic B-Splines as edge functions.

Chapter 4

Characterization of MRA of NURBs

4.1 Weighted MRA NURBs

We wish NURBs could be our scaling functions in MRA.

Is it possible?

Recall that the essence of MRA is nesting spaces, i.e., $V_i \subset V_{i+1}$ (see Section 2.3). As the scaling functions span these spaces, the nesting character of spaces requires a refinement relation among the scaling functions. Therefore if we could find a refinement relation for NURBs, they will be qualified for scaling functions.

At MRA level j, let $N_{m,j,k}$ be normalized B-Splines of order m; and it is defined by knot sequence $\mathbf{t}_j$=$\{t_{j,k}, t_{j,k+1},. . ., t_{j,k+m}\}$. $N_{m,j,k}$ satisfy the two–scale relation

$$N_{m,j,k}(t) = \sum_i p_{j,k;i} N_{m,j+1,i}(t), \tag{4.1.1}$$

where $p_{j,k;i}$ are two–scale sequences. We can rewrite the equation above in the matrix format

$$\mathbf{N}_j = \mathbf{P}_j \mathbf{N}_{j+1}, \tag{4.1.2}$$

where $\mathbf{P_j}$ are the two–scale matrices, and $\mathbf{N}_j$ and $\mathbf{N}_{j+1}$ are vectors B-Splines of order m at levels j and $j + 1$.

Let weight vectors be

$$\mathbf{w}_j = (w_{j,k})^T, \quad w_{j,k} > 0, \text{ and } j, k \in \mathbb{Z}.$$

At level j, let $R_{m,j,k}(t)$ be NURBs in which B-Splines have order m, and $R_{m,j,k}(t)$ are defined by knot sequence $\mathbf{t}_j$=$\{t_{j,k}, t_{j,k+1}, \ldots, t_{j,k+m}\}$ with weight vector $\mathbf{w}_j$. We therefore have the following NURBs basis functions. Notice that at the same MRA level, all NURBs share the same denominator.

$$R_{m,j,k} = \frac{w_{j,k}N_{m,j,k}(t)}{\sum_i w_{j,i}N_{m,j,i}(t)} \quad \text{for all } j, k \tag{4.1.3}$$

We need to find their refinement relations such that

$$R_{m,j,k} = \sum_i p^w_{j,k;i} R_{m,j+1,i}(t) \tag{4.1.4}$$

where $p^w_{j,k;i}$ are two–scale sequences.

In their paper [18], Chui and Lian found that in order to have refinement relation, NURBs must satisfy certain condition. They stated it as the following theorem.

If $V_j = span\{R_{m,j,k}(.)\}$, then $V_j \subset V_{j+1}$ *if and only if* the weight vectors $\mathbf{w}_j$ and $\mathbf{w}_{j+1}$ satisfy

$$\frac{w_{j+1,l}}{w_{j+1,l-1}} = \frac{\sum_u w_{j,u}p_{j,u;l}}{\sum_v w_{j,v}p_{j,v;l-1}} \quad \text{for all } l \tag{4.1.5}$$

where $p_{j,k;l}$ are the two–scale sequences in (4.1.1).

And the two–scale sequences can be derived by

$$p^w_{j,k;l} = \frac{w_{j,k}p_{j,u;l}}{\sum_v w_{j,v}p_{j,v;l}} \quad \text{for all } k, l, \tag{4.1.6}$$

and therefore

$$\sum_i w_{j+1,i}N_{m,j+1,i}(t) = \frac{w_{j+1,1}}{w_{j,u}\sum_u p_{j,u;1}} \sum_v w_{j,v}N_{m,j,v}(t), \quad \text{for all } j. \tag{4.1.7}$$

Moreover, if an initial condition is imposed by

$$w_{j+1,1} = \sum_i w_{j,i} p_{j,i;1} \quad \text{for all } j, \tag{4.1.8}$$

then

$$w_{j+1,l} = \sum_i w_{j,i} p_{j,i;l} \quad \text{for all } j. \tag{4.1.9}$$

Consequently, (4.1.7) leads to the following important result

$$\sum_n w_{j+1,n} N_{\mathbf{t}_{j+1},n}(t) = \sum_n w_{j,u} N_{\mathbf{t}_j,n}(t) \quad \text{for all } j. \tag{4.1.10}$$

This means that NURBs at all different levels share the same denominator. Furthermore, (4.1.6) turns to a simpler relation:

$$p^w_{j,k;l} = \frac{w_{j,k}}{w_{j+1,l}} p_{j,k;l} \quad \text{for all } j, k, l. \tag{4.1.11}$$

We adopt this important theorem in our NURBs by assuming that all our NURBs satisfy the conditions of (4.1.5) and (4.1.8). If we introduce the notation

$$\omega_j = \sum_i w_{j,i} N_{m,\mathbf{t}_j,i}(t) \quad \text{for all } j, \tag{4.1.12}$$

then (4.1.10) implies

$$\omega_j = \omega_0 \quad \text{for all } j, \tag{4.1.13}$$

and (4.1.3) becomes

$$R_{m,j,k} = \frac{w_{j,k} N_{m,j,k}(t)}{\omega_0} \quad \text{for all } j, k. \tag{4.1.14}$$

Since we focus on NURBs with order $m = 4$ in this dissertation, as a default, we omit m in our cubic NURBs notation. I.E.,

$$R_{j,k}(t) = \sum_i p^w_{j,k;i} R_{j+1,i}(t) \tag{4.1.15}$$

represents

$$R_{4,j,k}(t) = \sum_i p^w_{j,k;i} R_{4,j+1,i}(t). \tag{4.1.16}$$

In addition, if we let $\mathbf{w}_j = [w_{j,1}, w_{j,2}, \ldots]$ be the vector of weights for scaling functions at level j, then (4.1.9) implies the following equation

$$\mathbf{w}_{j+1} = \mathbf{P}_j^T \mathbf{w}_j. \tag{4.1.17}$$

This gives the relation of two weight vectors at adjacent levels. If the weight vector at the original level is given, the weight vectors for the rest levels could be attained through two–scale matrices of corresponding B-Splines.

4.2 Weighted MRA NURBs at Boundaries

In this section, we will construct cubic NURBs with natural cubic NURBs as edge functions.

Given a non-uniform knot sequence, let R_0 be scaling function vector at level 0, and the corresponding weight vector be $\mathbf{w}_0 = [w_{0,0}, w_{0,1}, w_{0,2}, w_{0,3}, \ldots]^T$. By (4.1.14) the following equations are derived:

$$R_0 = \begin{bmatrix} R^e_{0,1} \\ R^e_{0,2} \\ R_{0,1} \\ R_{0,2} \\ R_{0,3} \\ \vdots \end{bmatrix} = \begin{bmatrix} \frac{w_{0,0} N^e_{0,1}(t)}{\omega_0} \\ \frac{w_{0,1} N^e_{0,2}(t)}{\omega_0} \\ \frac{w_{0,2} N_{0,1}(t)}{\omega_0} \\ \frac{w_{0,3} N_{0,2}(t)}{\omega_0} \\ \frac{w_{0,4} N_{0,3}(t)}{\omega_0} \\ \vdots \end{bmatrix} = \frac{1}{\omega_0} \begin{bmatrix} w_{0,0} & & & & & \\ & w_{0,1} & & & & \\ & & w_{0,2} & & & \\ & & & w_{0,3} & & \\ & & & & w_{0,4} & \\ & & & & & \ddots \end{bmatrix} \begin{bmatrix} N^e_{0,1}(t) \\ N^e_{0,2}(t) \\ N_{0,1}(t) \\ N_{0,2}(t) \\ N_{0,3}(t) \\ \vdots \end{bmatrix} \tag{4.2.1}$$

If we let

$$W_j = \begin{bmatrix} w_{j,0} & & & & \\ & w_{j,1} & & & \\ & & w_{j,2} & & \\ & & & w_{j,3} & \\ & & & & \ddots \end{bmatrix} \tag{4.2.2}$$

and

$$\mathbf{N_j} = \begin{bmatrix} N^e_{j,1}(t) & N^e_{j,2}(t) & N_{j,1}(t) & N_{j,2}(t) & N_{j,3}(t) & \dots \end{bmatrix}^{\mathbf{T}}, \tag{4.2.3}$$

then

$$R_0 = \frac{1}{\omega_0} W_0 \mathbf{N_0}. \tag{4.2.4}$$

Likewise, at the next fine level, we have

$$R_1 = \begin{bmatrix} R^e_{1,1} \\ R^e_{1,2} \\ R_{1,1} \\ R_{1,2} \\ R_{1,3} \\ \vdots \end{bmatrix} = \begin{bmatrix} \frac{w_{1,0} N^e_{1,1}(t)}{\omega_0} \\ \frac{w_{1,1} N^e_{1,2}(t)}{\omega_0} \\ \frac{w_{1,2} N_{1,1}(t)}{\omega_0} \\ \frac{w_{1,3} N_{1,2}(t)}{\omega_0} \\ \frac{w_{1,4} N_{1,3}(t)}{\omega_0} \\ \vdots \end{bmatrix} = \frac{1}{\omega_0} W_1 \mathbf{N_1}. \tag{4.2.5}$$

4.3 Two–Scale Matrices

With NURBs as our scaling functions, our next step is to find the two–scale matrix P^w_0 such that

$$R_0 = P^w_0 R_1. \tag{4.3.1}$$

By putting (4.2.4), (4.2.5) and (4.1.2) into the equation above, we get

$$W_0 P_0 = P^w_0 W_1$$

or

$$P^w_0 = W_0 P_0 W_1^{-1}. \tag{4.3.2}$$

We finally derive the following equation for P_0^w

$$P_0^w = \begin{bmatrix} w_{0,0} & & & & \\ & w_{0,1} & & & \\ & & w_{0,2} & & \\ & & & w_{0,3} & \\ & & & & \ddots \end{bmatrix} P_0 \begin{bmatrix} \frac{1}{w_{1,0}} & & & & \\ & \frac{1}{w_{1,1}} & & & \\ & & \frac{1}{w_{1,2}} & & \\ & & & \frac{1}{w_{1,3}} & \\ & & & & \ddots \end{bmatrix}. \qquad (4.3.3)$$

Equation (4.3.3) helps us to get two-scale matrices in knot insertion. We will discuss two cases: one knot insertion and arbitrary knot insertion in each knot span.

First we consider one-knot insertion case. Let a new knot τ be inserted on $[t_1, t_2]$; then putting (3.2.9) in (4.3.3), we get

$$\mathbf{P^w_{0[t_1,t_2]}} = \begin{bmatrix} \frac{w_{0,0}}{w_{1,0}} & \frac{w_{0,0}}{w_{1,1}}(1-\alpha_0) & & & & \\ & \frac{w_{0,1}}{w_{1,1}}\alpha_0 & \frac{w_{0,1}}{w_{1,2}}(1-\alpha_1) & & & \\ & & \frac{w_{0,2}}{w_{1,2}}\alpha_1 & \frac{w_{0,2}}{w_{1,3}} & & \\ & & & & \frac{w_{0,3}}{w_{1,4}} & \\ & & & & & \ddots \end{bmatrix}. \qquad (4.3.4)$$

If we apply (4.1.17) to replace weights at fine level, and for the simplicity, let weights $\{w_{0,0}, w_{0,1}, w_{0,2}, w_{0,3}, \ldots\}$ be $\{w_0, w_1, w_2, \ldots\}$, we then get

$$\mathbf{P^w_{0[t_1,t_2]}} = \begin{bmatrix} 1 & \frac{w_0(1-\alpha_0)}{w_0(1-\alpha_0)+w_1\alpha_0} & & & & \\ & \frac{w_1\alpha_0}{w_0(1-\alpha_0)+w_1\alpha_0} & \frac{w_1(1-\alpha_1)}{w_1(1-\alpha_1)+w_2\alpha_1} & & & \\ & & \frac{w_2\alpha_1}{w_1(1-\alpha_1)+w_2\alpha_1} & 1 & & \\ & & & & 1 & \\ & & & & & \ddots \end{bmatrix}. \qquad (4.3.5)$$

Let

$$\alpha_i^w = \frac{w_{i+1}\alpha_i}{w_i(1-\alpha_i) + w_{i+1}\alpha_i} \qquad (4.3.6)$$

with α_i defined in (3.2.8), then the matrix above is

$$\mathbf{P^w_{0[t_1,t_2]}} = \begin{bmatrix} 1 & 1-\alpha_0^w & & & & \\ & \alpha_0^w & 1-\alpha_1^w & & & \\ & & \alpha_1^w & 1 & & \\ & & & & 1 & \\ & & & & & \ddots \end{bmatrix}. \tag{4.3.7}$$

Similarly, with the same definition of α_i^w and α_i, we get the two–scale matrix when the knot is inserted on $[t_2, t_3]$

$$\mathbf{P^w_{0[t_2,t_3]}} = \begin{bmatrix} 1 & 1-\alpha_0^w & & & & & \\ & \alpha_0^w & 1-\alpha_1^w & & & & \\ & & \alpha_1^w & 1-\alpha_2^w & & & \\ & & & \alpha_2^w & 1 & & \\ & & & & & 1 & \\ & & & & & & \ddots \end{bmatrix}, \tag{4.3.8}$$

and on $[t_i, t_{i+1}]$ where $i \geq 3$,

$$\mathbf{P^w_{0[t_i,t_{i+1}]}} = \begin{bmatrix} \ddots & & & & & & & \\ & 1 & & & & & & \\ & & 1 & 1-\alpha_{i-2}^w & & & & \\ & & & \alpha_{i-2}^w & 1-\alpha_{i-1}^w & & & \\ & & & & \alpha_{i-1}^w & 1-\alpha_i^w & & \\ & & & & & \alpha_i^w & 1 & \\ & & & & & & & 1 \\ & & & & & & & & \ddots \end{bmatrix}. \tag{4.3.9}$$

Finally if we insert an arbitrary knot in every knot span, the two-scale matrix becomes

$$\mathbf{P}^{\mathbf{w}}_{\mathbf{0[t_1,t_s]}} = \begin{bmatrix} 1 & 1-\alpha_0^w & \beta_1^w & & & & & \\ & \alpha_0^w & 1-\beta_1^w-\gamma_1^w & 1-\alpha_1^w & \beta_2^w & & & \\ & & \gamma_1^w & \alpha_1^w & 1-\beta_2^w-\gamma_2^w & 1-\alpha_2^w & \beta_3^w & \\ & & & & \gamma_2^w & \alpha_2^w & 1-\beta_3^w-\gamma_3^w & \\ & & & & & & \gamma_3^w & \\ & & & & & & & \ddots \end{bmatrix}, \tag{4.3.10}$$

where

$$\beta_i^w = \frac{\beta_i w_i}{\beta_i w_{i-1} + (1-\beta_i-\gamma_i)w_i + \gamma_i w_{i+1}} \tag{4.3.11}$$

and

$$\gamma_i^w = \frac{\gamma_i w_{i+1}}{\beta_i w_{i-1} + (1-\beta_i-\gamma_i)w_i + \gamma_i w_{i+1}}. \tag{4.3.12}$$

4.4 Examples

Example 1

If weights are all 1, then rational two–scale matrices for cubic NURBs in (4.3.7), (4.3.8) and (4.3.9) are identical to ones for cubic B-Splines in (3.2.9),(3.2.10) and (3.2.11).

Example 2

Let knot sequence be $\mathbf{t}_0 = \{0,0,0,0,1.5,2,3.2,4,5.5,6.3,7,7,7,7\}$, and weight sequence be $\{w_0, w_1, \ldots, w_7\} = \{1,4,2,3,1,5,2,4\}$. The insertion knot is $\tau = 1.2$. We'll compute $P^w_{0,[0,1.5]}$ by (4.3.7).

From given conditions, we have $t_1 = 0, t_2 = 1.5, t_3 = 2, \ldots, t_8 = 7$, and by the Figure 3.8 we have $t_0 = -1.2$

(3.2.8) tells me that

$$\alpha_0 = \frac{\tau - t_0}{t3 - t0} = \frac{3}{4}, \qquad \alpha_1 = \frac{\tau - t_1}{t_4 - t_1} = \frac{3}{8}$$

then with (4.3.6), we have

$$\alpha_0^w = \frac{w_1\alpha_0}{w_0(1-\alpha_0)+w_1\alpha_0} = \frac{3}{4} \qquad \alpha_1^w = \frac{w_2\alpha_1}{w_1(1-\alpha_1)+w_2\alpha_1} = \frac{3}{13}.$$

So the two–scale rational matrix is

$$P_{0[0,1.5]}^w = \begin{bmatrix} 1 & \frac{1}{4} & & & & \\ & \frac{3}{4} & \frac{10}{13} & & & \\ & & \frac{3}{13} & 1 & & \\ & & & & 1 & \\ & & & & & \ddots \end{bmatrix}$$

Chapter 5

NURBlets

In multiresolution analysis (MRA), there are two basic tasks:

- At level j, choose proper scaling functions $\Phi_j(t)$ and find the two–scale matrix P_j in refinement relation.
- Find the proper matrix Q_j to construct wavelets $\Psi_j(t)$ with desired properties.

We have finished the first task. We have constructed our scaling functions with cubic NURBs on intervals, and their two-scale matrix have been given by P_j^w. In this chapter, we'll implement the second task, that is, to construct the weighted biorthogonal wavelets (NURBlets) with one vanishing moment.

Before start our model, we first investigate the lifting scheme, a technique that we'll apply in constructing NURBlets.

5.1 Lifting Scheme

Lifting scheme was first proposed by Sweldens [63]. It is a method to construct biorthogonal wavelets and is quite different from classical ones. Lifting scheme does not depend on Fourier transform but classical ones do. While classical schemes can only translate and dilate one function, lifting scheme can work on more than one at one time [63]. Since it has been proposed, it has aroused the great attention from researchers. Some of recent approaches are [61, 40, 36, 1, 44, 45].

5.1.1 Why Lifting Scheme

The answer is classical schemes are not available to NURBlet construction. Fourier transform does not work when wavelets are not shifts; neither is it available when wavelets are on boundary domains, or with weights, while all these characters are significant in our modeling. Lifting scheme, on the other hand, provides an alternative.

It also has some other advantages [23, 63]:

- It is faster. Lifting scheme has recursive computations on both low and high pass filters, which speeds the calculation.
- No auxiliary memory is needed for calculation of wavelet transform.
- The inverse wavelet transform can be found easily by doing opposite operations of the forward transformation.

5.1.2 Sketch of Lifting Scheme

Lifting scheme constructs biorthogonal wavelets by "lifting" them from lazy wavelets. The idea is following([64]).

In MRA, suppose we have a set of data $\{c_{0,0}, c_{0,1}, \ldots\}$ at level 0, and decompose it to the coarser level -1. We filter some detailed data and want the rest to be able to capture the general information still. Suppose we decompose even samples to the coarser level, and filter the odd ones:

$$c_{-1,i} = c_{0,2i} \qquad i \in \mathbb{Z}. \tag{5.1.1}$$

We would like to recover $\{c_{-1,i}\}$ back to $\{c_{0,i}\}$. Obviously there is a difference between two sets of coefficients, and we denote the coefficients $d_{-1,j}$ be such differences and call them *wavelet coefficients*. It is trivial to see that the smaller these wavelet coefficients are, the better the approximation would be. The question is, how do we find the differences? An intuitively "lazy" way is to let the lost information (i.e., the odd samples) be the differences,

$$d_{-1,i} = c_{0,2i+1} \qquad k \in \mathbb{Z}. \tag{5.1.2}$$

Such wavelets are called "*lazy wavelets*". Though the "lazy" choice does not meet our expectation for ideal wavelet coefficients (as smaller as possible), it is a good starting point to tackle the problem.

Considering the fact that the lost information (odd samples) contains certain correlation among their neighboring samples, a reasonable yet simple guess is that an odd sample has a relation with at least its two neighboring even samples. For the simplicity, we assume that it is an average of two neighboring even samples. This leads to the following "lifting"

$$d_{-1,i} = c_{0,2i+1} - \frac{c_{0,2i} + c_{2i+2}}{2}. \tag{5.1.3}$$

If $c_{0,i}$ is piecewise colinear between even samples, the wavelet coefficients above are zero, in which case the approximation is the best. Figure 5.1 shows the relation.

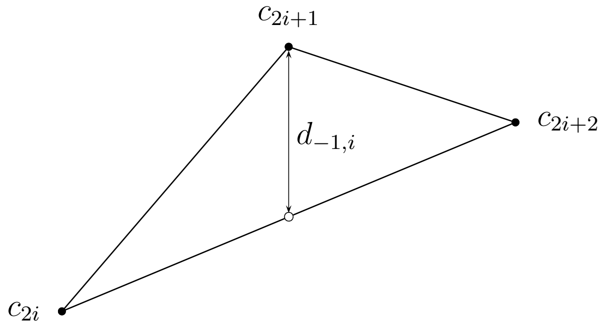


Figure 5.1: Wavelet coefficients are the average of two neighboring even samples in the original sample: Failure to be linear.

Moreover, we wish the averages of data for each level to be the same. This means $\sum_i c_{-1,i} = \frac{1}{2}\sum_i c_{0,i}$. So we "lift" the $c_{-1,i}$ with the help of wavelet coefficients $d_{-1,i}$ such that

$$c_{-1,i} = c_{0,2i} + \frac{1}{4}(d_{-1,i-1} + d_{-1,i}). \tag{5.1.4}$$

This is called "lifting scheme" because we "lift" our wavelets from "lazy" ones. Figure 5.2 shows the relation among coefficients.

In summery, in lifting scheme, wavelet transformation has two steps:

1. Use (5.1.3) to computer wavelet coefficients as a failure to be linear. In other words,

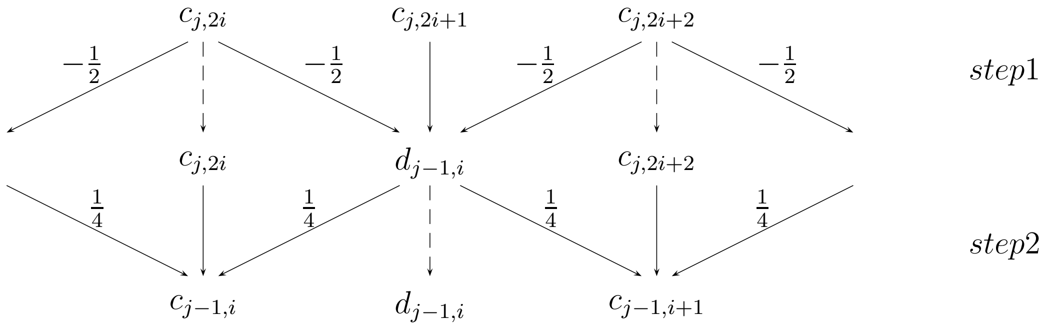


Figure 5.2: The lifting Scheme.

we'll construct wavelets doing nothing but contain wavelet properties.

2. Use (5.1.4) to lift the coefficients we got from the first step so that they attain certain properties we desire for.

5.1.3 Lifting Scheme

Now we present lifting scheme formally([23, 63]):

At the original level let $\phi_{0,k}$ and $\tilde{\phi}_{0,k}$ be scaling functions and their dual ones; let $\psi_{0,k}$ and $\tilde{\psi}_{0,k}$ be lazy wavelet functions and corresponding dual ones. Then $\psi_{0,\text{lift}}$, $\tilde{\psi}_{0,\text{lift}}$ and $\phi_{0,\text{lift}}$ are ones after lifting scheme such that

$$\psi_{0,\text{lift}}(x) = \psi_{0,k}(x) - \sum_k s_k \phi_{0,k}(x) \tag{5.1.5}$$

$$\tilde{\psi}_{0,\text{lift}}(x) = \sum_k \tilde{g}_{0,k} \tilde{\phi}_{1,k} \tag{5.1.6}$$

$$\tilde{\phi}_{0,\text{lift}}(x) = \sum_k \tilde{h}_{0,k} \tilde{\phi}_{1,k} + \sum_k s_{-k} \tilde{\psi}_{0,k} \tag{5.1.7}$$

where s_k can been freely chosen, and $\tilde{h}_{0,k}$ and $\tilde{g}_{0,k}$ are coefficients in refinement relations $\tilde{\phi}_0(x) = \sum_k \tilde{h}_{0,k} \tilde{\phi}_{1,k}(x)$ and $\tilde{\psi}_0(x) = \sum_k \tilde{g}_{0,k} \tilde{\phi}_{1,k}$.

5.2 Single-Knot Wavelets for Non-Uniform Rational B-Splines

Recently, more and more efforts are made on non-uniform B-Splines research ([44, 45, 49, 12, 58, 11, 48, 8]). Among them, some proposals for non-uniform biorthogonal wavelets are offered ([8, 49, 45]. However, so far in our knowledge, there is no effort working on non-uniform rational B-Splines(i.e., NURBlets). Among these previous work, we are particularly interested in the approach that Bertram offered [8].

In his paper, Bertram proposed a biorthogonal wavelet construction for non-uniform B-Splines. The method can remove knots in arbitrary order, and when a knot is re-inserted, it minimizes the displacement control points. His idea is following.

When a knot is inserted, more detail is added to a geometric shape. Bertram introduced a wavelet coefficient on the coarse level which is simply a displacement. It is the lazy wavelet transformation as we mentioned in section 5.1. Then he optimized the fitting for cubic wavelet with the lifting scheme. Instead of applying L^2-orthogonalization, he minimized the displacement of control points obtained from knot removal followed by reinserting a zero wavelet coefficient. He got the minimum value by orthogonalizing the lazy wavelet with the coarse scaling functions. Making an assumption that scaling functions on finer level must form an orthogonal basis with proper weighting, he obtained the result.

Unfortunately, the assumption that scaling functions must be orthogonal is too specific to be general. But Bertram's views are valuable. We extend his approach to construct our cubic NURBlets on intervals without the assumption in the scheme.

There are two different cases in single-knot insertion in our approach —– depending on whether the knot is inserted at boundaries or not. We will start our approach from the non-boundary case.

5.2.1 One Knot Insertion in a Non-Boundary Knot Span

On the interval $[t_1, t_s]$, suppose $\Phi_1(x)$ is the vector of NURBs as scaling functions, $\Psi_1(x)$ is that of NURBlets as wavelets, and a new knot τ is inserted in (t_r, t_{r+1}), $3 \leq r \leq s-3$ at MRA

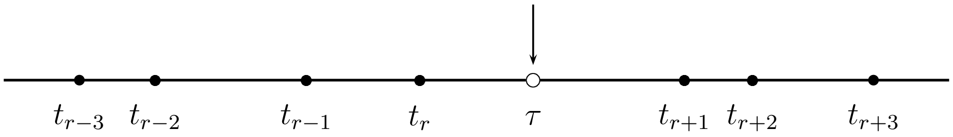


Figure 5.3: Knot sequence.

level 1. See Figure 5.3. Recall the refinement relation in scaling functions

$$\begin{aligned}
\mathbf{\Phi_0} &= \begin{bmatrix} \phi_{0,r-3} \\ \phi_{0,r-2} \\ \phi_{0,r-1} \\ \phi_{0,r} \end{bmatrix} \\
&= P^w_{0[t_r,t_{r+1}]}\mathbf{\Phi_1} = \begin{bmatrix} 1 & 1-\alpha^w_{i-2} & & & \\ & \alpha^w_{i-2} & 1-\alpha^w_{i-1} & & \\ & & \alpha^w_{i-1} & 1-\alpha^w_i & \\ & & & \alpha^w_i & 1 \end{bmatrix} \begin{bmatrix} \phi_{1,r-3} \\ \phi_{1,r-2} \\ \phi_{1,r-1} \\ \phi_{1,r} \\ \phi_{1,r+1} \end{bmatrix}.
\end{aligned} \tag{5.2.1}$$

For the simplicity, we replace α^w_{i-2}, α^w_{i-1} and α^w_i by a_1, a_2 and a_3, and P^w_0 by $P^w_{0[t_r,t_{r+1}]}$.

Our goal is to reconstruct data. In other words, we want to derive Φ_1 given Φ_0.

However, we can not get it directly from (5.2.1). The reason is matrix P^w_0 is not square, let alone it is invertible. A "lazy" way to deal with it is to add one row and extend P^w to the square matrix as the following

$$P^w_{0,lazy} = \begin{bmatrix} 1 & 1-\alpha^w_{i-2} & & & \\ & \alpha^w_{i-2} & 1-\alpha^w_{i-1} & & \\ & & 1 & & \\ & & \alpha^w_{i-1} & 1-\alpha^w_i & \\ & & & \alpha^w_i & 1 \end{bmatrix}. \tag{5.2.2}$$

This is the "lazy scheme" we've discussed in the previous section. To lift it, we left multiply it by a “lifting” matrix,

$$P^w_{lift} = \begin{bmatrix} 1 & & & & \\ & 1 & & & \\ b_0 & b_1 & 1 & b_2 & b_3 \\ & & & 1 & \\ & & & & 1 \end{bmatrix} \begin{bmatrix} 1 & 1-a_1 & & & \\ & a_1 & 1-a_2 & & \\ & & 1 & & \\ & & a_2 & 1-a_3 & \\ & & & a_3 & 1 \end{bmatrix}. \tag{5.2.3}$$

Notice that the lifting scheme in fact generalizes the “lazy” row we add in (5.2.2) such that

$$P^w_{lift} = \begin{bmatrix} 1 & 1-a_1 & & & \\ & a_1 & 1-a_2 & & \\ x_0 & x_1 & x_2 & x_3 & x_4 \\ & & a_2 & 1-a_3 & \\ & & & a_3 & 1 \end{bmatrix}, \tag{5.2.4}$$

where

$$\begin{cases} x_0 = b_0 \\ x_1 = (1-a_1)b_0 + a_1 b_1 \\ x_2 = (1-a_2)b_1 + a_2 b_2 + 1 \\ x_3 = (1-a_3)b_2 + a_3 b_3 \\ x_4 = b_3. \end{cases} \tag{5.2.5}$$

If we let

$$S^T = \begin{bmatrix} 1 & 1-a_1 & & & \\ & a_1 & 1-a_2 & & \\ & & 1 & & \\ & & a_2 & 1-a_3 & \\ & & & a_3 & 1 \end{bmatrix} \tag{5.2.6}$$

and

$$(F^{-1})^T = \begin{bmatrix} 1 & & & & \\ & 1 & & & \\ b_0 & b_1 & 1 & b_2 & b_3 \\ & & & 1 & \\ & & & & 1 \end{bmatrix}, \tag{5.2.7}$$

and if we define

$$\Phi_0^{etd} = \begin{bmatrix} \phi_{0,r-3} & \phi_{0,r-2} & \psi & \phi_{0,r-1} & \phi_{0,r} \end{bmatrix}^T,$$

where ψ is the wavelet at level 0, we then have

$$\Phi_0^{etd} = (SF^{-1})^T \Phi_1. \tag{5.2.8}$$

Equivalently, this means

$$F^T \Phi_0^{etd} = S^T \Phi_1,$$

i.e.,

$$\begin{bmatrix} \phi_{0,r-3} \\ \phi_{0,r-2} \\ \psi - \sum_{i=0}^{3} b_i \phi_{0,r-3+i} \\ \phi_{r-1} \\ \phi_r \end{bmatrix} = S^T \begin{bmatrix} \phi_{1,r-3} \\ \phi_{1,r-2} \\ \phi_{1,r-1} \\ \phi_{1,r} \\ \phi_{1,r+1} \end{bmatrix}. \tag{5.2.9}$$

The lifted wavelet therefore has the form of

$$\psi = \phi_{1,r-1} + \sum_{i=0}^{3} b_i \phi_{0,r-3+i}. \tag{5.2.10}$$

Notice that equation (5.2.4) implies

$$P_{lift}^{w} = (SF^{-1})^T. \tag{5.2.11}$$

Suppose $c_{0,i}$ and $c_{1,j}$ are coefficients at coarse and fine level, then (2.3.2) tells us that

$$\sum_{i=0}^{3} c_{0,i}\phi_{0,r+i-3}(x) + d\psi(x) = \sum_{j=0}^{4} c_{1,j}\phi_{1,r+j-3}(x), \tag{5.2.12}$$

where d is the coefficient of wavelet $\psi(x)$ at coarse level capturing the detail information.

If we define

$$\mathbf{c_0^{etd}} = \left[\begin{array}{ccccc} c_{0,0} & c_{0,1} & d & c_{0,2} & c_{0,3} \end{array}\right]^{\mathbf{T}}, \tag{5.2.13}$$

then (5.2.12) turns out

$$(\mathbf{c}_0^{etd})^T \Phi_0^{etd}(x) = \mathbf{c}_1^T \Phi_1(x). \tag{5.2.14}$$

Considering (5.2.8), it becomes

$$\mathbf{c}_0^{etd} = FS^{-1}\mathbf{c}_1. \tag{5.2.15}$$

Hence a chart can be applied to describe our transformation. See Figure 5.4.

$$\begin{tikzcd}
\mathbf{\Phi_0} & \mathbf{\Phi_1} \\
FS^{-1}\mathbf{c}_1 \arrow[d, "X \text{ (set wavelet coeffs. zero)}"] & \mathbf{c}_1 \arrow[l] \\
XFS^{-1}\mathbf{c}_1 \arrow[r] & SF^{-1}XFS^{-1}\mathbf{c}_1
\end{tikzcd}$$

Figure 5.4: Knot removal minimizing displacement of control points. The displacement is $\|\mathbf{c}_1 - SF^{-1}XFS^{-1}\mathbf{c}_1\|_{l^2}$.

We decompose coefficients from fine level to coarse level. After adopt an X matrix to cancel wavelet coefficients, we compose the coefficients back. The difference between original coefficients $\mathbf{c}_1$ and ones after removing a knot is

$$\mathbf{c}_1 - SF^{-1}XFS^{-1}\mathbf{c}_1. \tag{5.2.16}$$

Based on L^2-orthogonalization, lifting matrix F is decided when a least-square fit is approached on the difference above. This is equivalent to find the minimum value of the expression

$$min \ \|I - SF^{-1}XFS^{-1}\|_{l^2}. \tag{5.2.17}$$

5.2.2 Construct F Matrix

We wish to construct F matrix such that it minimizes the difference of transformation described in the previous section. Moreover, we want it to have constrain of first vanishing moment. The procedure is followed.

Step 1 Compute $SF^{-1}XFS^{-1}I$.

By computation, we get the result of $SF^{-1}XFS^{-1}I$ in (5.2.17)

$$SF^{-1}XFS^{-1}I = I + \mathbf{v}\mathbf{a}^T, \tag{5.2.18}$$

where

$$\mathbf{v} = \begin{bmatrix} -b_0 \\ b_0(a_1 - 1) - a_1 b_1 \\ b_1(a_2 - 1) - a_2 b_2 - 1 \\ b_2(a_3 - 1) - a_3 b_3 \\ -b_3 \end{bmatrix} \tag{5.2.19}$$

and

$$\mathbf{a}^T = \left[\begin{array}{ccccc} \frac{(a_1-1)(a_2-1)}{a_1} & \frac{a_2-1}{a_1} & 1 & \frac{a_2}{a_3-1} & \frac{a_2 a_3}{1-a_3} \end{array} \right]. \tag{5.2.20}$$

Simplify (5.2.17),

$$min \ \|I - SF^{-1}XFS^{-1}I\|_{l_2} = min \ \|\mathbf{v}\mathbf{a}^T\|_{l_2}. \tag{5.2.21}$$

If denote

$$\eta = \mathbf{v}\mathbf{a}^T, \tag{5.2.22}$$

we need to minimize η.

Step 2 Minimize η

To minimize η, it is equivalent to find the minimum of $\|\eta^T\eta\|_{l_2}$. Since

$$\eta^T\eta = (\mathbf{va}^T)^T(\mathbf{va}^T) = \mathbf{av}^T\mathbf{va}^T = \mathbf{a}(\mathbf{v}^T\mathbf{v})\mathbf{a}^T, \tag{5.2.23}$$

and vector $\mathbf{a}$ is determined as soon as the knot sequence and inserted-knot are given, the minimum of $\|\eta^T\eta\|_{l_2}$ in fact depends only on the value of $\|\mathbf{v}^T\mathbf{v}\|_{l_2}$.

Step 3 Minimize $\|\mathbf{v}^T\mathbf{v}\|$ under the condition of first vanishing moment

Let

$$f = \mathbf{v}^T\mathbf{v}, \tag{5.2.24}$$

and placing (5.2.19) into (5.2.24), we then get

$$f = b_0^2 + [b_0(a_1-1) - a_1b_1]^2 + [b_1(a_2-1) - a_2b_2 - 1]^2 + [b_2(a_3-1) - a_3b_3]^2 + (b_3)^2. \tag{5.2.25}$$

Recall the first moment is defined as

$$g(x) = \int_{\omega_0} \psi_0 dx, \tag{5.2.26}$$

Applying (1.1.3),(5.2.10), and (4.1.12) into the first moment above:

$$\begin{aligned} g(x) &= \int_{\omega_0} \psi_0 dt \\ &= \int_{\omega_0} \phi_{1,r-1} dt + \int_{\omega_0} \sum_{i=0}^{3} b_i \phi_{0,r-3+i} dt \\ &= \int (\omega_0)^2 \frac{w_{1,r-1}N_{1,r-1}}{\omega_0} dt + \sum_{i=0}^{3} b_i \int (\omega_0)^2 \frac{w_{0,r-3+i}N_{0,r-3+i}}{\omega_0} dt \\ &= w_{1,r-1} \int (\sum_{k=0}^{3} w_{0,k}N_{0,k}) N_{1,r-1} dt + \sum_{i=0}^{3} b_i w_{0,r-3+i} \int (\sum_{k=0}^{3} w_{0,k}N_{0,k}) N_{0,r-3+i} dt \\ &= w_{1,r-1} \int (\mathbf{w}_0)^T \mathbf{N}_0 N_{1,r-1} dt + \sum_{i=0}^{3} b_i w_{0,r-3+i} \int (\mathbf{w}_0)^T \mathbf{N}_0 N_{0,r-3+i} dt \\ &= w_{1,r-1} (\mathbf{w}_0)^T \int \mathbf{N}_0 N_{1,r-1} dt + \sum_{i=0}^{3} b_i w_{0,r-3+i} (\mathbf{w}_0)^T \int \mathbf{N}_0 N_{0,r-3+i} dt \end{aligned} \tag{5.2.27}$$

Here $\mathbf{w}_0 = [w_{0,0}\ w_{0,1}\ \ldots]^T$ is the weight vector at level 0, $\mathbf{N}_0$ and $\mathbf{N}_1$ are the B-Spline vectors at level 0 and 1, and $\mathbf{P}_0$ is two-scale matrix of B-Splines in refinement relation $\mathbf{N}_0 = \mathbf{P}_0\mathbf{N}_1$.

Our target is to minimize $f(x)$ under the constrain of the first vanishing moment. By Lagrange multipliers, if we let

$$G(x) = f(x) - \lambda g(x), \tag{5.2.28}$$

with some real value λ, the following group of equations is derived:

$$\begin{cases} \frac{\partial G}{\partial \lambda} = 0 \\ \frac{\partial G}{\partial b_i} = 0, \qquad i = 0, 1, 2, 3. \end{cases}$$

Finally the solution (i.e., $b_i, i = 0, \ldots, 3$) can be derived through the following equation

$$\Delta \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ \lambda \end{bmatrix} = \begin{bmatrix} 0 \\ a_2 - 1 \\ -a_1 \\ 0 \\ -\delta_{1,r-1} \end{bmatrix}. \tag{5.2.29}$$

where

$$\Delta = \begin{bmatrix} (a_1-1)^2+1 & -a_1(a_1-1) & & & \delta_{0,r-3} \\ -a_1(a_1-1) & a_1^2+(a_2-1)^2 & -a_2(a_2-1) & & \delta_{0,r-2} \\ & -a_2(a_2-1) & a_2^1+(a_3-1)^2 & -a_3(a_3-1) & \delta_{0,r-1} \\ & & -a_3(a_3-1) & a_3^2+1 & \delta_{0,r} \\ \delta_{0,r-3} & \delta_{0,r-2} & \delta_{0,r-1} & \delta_{0,r} & 0 \end{bmatrix} \tag{5.2.30}$$

with

$$\delta_{i,j} = -\frac{1}{2} w_{i,j} (\mathbf{w}_0)^T \langle \mathbf{N}_0, N_{i,j} \rangle; \quad i = 0, 1; j = r-3, \ldots, r. \tag{5.2.31}$$

5.2.3 One Knot Insertion at the Boundary

Inserting one knot at boundaries has two cases depending on whether it is on the first or second knot span (without losing generality, we only consider the left boundary). We first consider the later case, i.e., $\tau \in (t_2, t_3)$ where τ is a new inserted knot.

In the refinement relation

$$\begin{aligned}
\mathbf{\Phi_0} &= \begin{bmatrix} \phi^e_{0,0} \\ \phi^e_{0,1} \\ \phi_{0,1} \\ \phi_{0,2} \end{bmatrix} = P^w_{0,[t_2,t_3]}\mathbf{\Phi_1} \\
&= \begin{bmatrix} 1 & 1-a^w_0 & & & \\ & a^w_0 & 1-a^w_1 & & \\ & & a^w_1 & 1-a^w_2 & \\ & & & a^w_2 & 1 \end{bmatrix} \begin{bmatrix} \phi^e_{1,0} \\ \phi^e_{1,1} \\ \phi_{1,1} \\ \phi_{1,2} \\ \phi_{1,3} \end{bmatrix}.
\end{aligned} \tag{5.2.32}$$

Observe that $P^w_{0,[t_2,t_3]}$ has the same size and structure as $P^w_{0,[t_r,t_{r+1}]}$, hence if we let $\alpha^w_i = a_{i+1}$ $(i = 0, 1, 2)$, we will get the same result as in the general case (see the previous subsection), except the wavelet which has the form of

$$\psi = \phi_{1,1} + (b_0\phi^e_{0,0} + b_1\phi^e_{0,1} + b_2\phi_{0,1} + b_3\phi_{0,2}). \tag{5.2.33}$$

The case of first knot span insertion (i.e., $\tau \in (t_1, t_2)$) is different simply because its two-scale matrix is 3 by 4 in the refinement relation:

$$\begin{aligned}\mathbf{\Phi_0} &= \begin{bmatrix} \phi^e_{0,0} \\ \phi^e_{0,1} \\ \phi_{0,1} \end{bmatrix} = P^w_{0,[t_1,t_2]}\mathbf{\Phi_1} \\ &= \begin{bmatrix} 1 & 1-\alpha^w_0 & & \\ & \alpha^w_0 & 1-\alpha^w_1 & \\ & & \alpha^w_1 & 1 \end{bmatrix} \begin{bmatrix} \phi^e_{1,0} \\ \phi^e_{1,1} \\ \phi_{1,1} \\ \phi_{1,2} \end{bmatrix}. \end{aligned} \tag{5.2.34}$$

We redefine F matrix by

$$(F^{-1})^T = \begin{bmatrix} 1 & & & \\ b_0 & 1 & b_1 & b_2 \\ & & 1 & \\ & & & 1 \end{bmatrix}, \tag{5.2.35}$$

and with the similar “lifting” procedure, the wavelet function becomes

$$\psi = \phi^e_{1,1} + b_0\phi^e_{0,0} + b_1\phi^e_{0,1} + b_2\phi_{0,1}. \tag{5.2.36}$$

Likewise, if $a_i = \alpha_i$ $(i = 0, 1)$, then the vector is

$$\mathbf{v} = -\begin{bmatrix} b_0 \\ (1-a_0)b_0 + a_0b_1 + 1 \\ (1-a_1)b_1 + a_1b_2 \\ b_2 \end{bmatrix}, \tag{5.2.37}$$

and the final solution for coefficients in F matrix can be retrieved from the following equation

$$\Delta \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ \lambda \end{bmatrix} = \begin{bmatrix} a_0 - 1 \\ -a_0 \\ 0 \\ -\delta_{1,1} \end{bmatrix}, \tag{5.2.38}$$

where

$$\Delta = \begin{bmatrix} (a_0-1)^2+1 & -a_0(a_0-1) & & \delta^e_{0,0} \\ -a_0(a_0-1) & a_0+(a_1-1)^2 & -a_1(a_1-1) & \delta^e_{0,1} \\ & -a_1(a_1-1) & a_1^2+1 & \delta_{0,1} \\ \delta^e_{0,0} & \delta^e_{0,1} & \delta_{0,1} & 0 \end{bmatrix}, \tag{5.2.39}$$

$$\delta_{i,j} = -\frac{1}{2} w_{i,j} \mathbf{w}_0 \langle \mathbf{N}_0, N_{i,j} \rangle \tag{5.2.40}$$

and

$$\delta^e_{i,j} = -\frac{1}{2} w_{i,j} \mathbf{w}_0 \langle \mathbf{N}_0, N^e_{i,j} \rangle. \tag{5.2.41}$$

where $i, j = 0, 1$.

5.3 Two-Knot Insertion

One-knot insertion is the first step in our approach. The ultimate goal is to insert an arbitrary knot in each knot span with certain degree of vanishing moments. Due to the complication of rational character that NURBs have, fulfilling the goal becomes difficult. In this section, we discuss using lifting scheme to construct biorthogonal wavelets on NURBs in the case of two-knot insertion.

Suppose two new knots τ_1 and τ_2 are inserted in the knot span (t_{r-1}, t_r) and (t_r, t_{r-1})

respectively. For the simplicity, we discuss the insertion in a general way such that

$$\mathbf{\Phi_0} = \mathbf{P_0}\mathbf{\Phi_1} = \begin{bmatrix} 1 & \alpha_1 & \alpha_2 & & & \\ & 1-\alpha_1 & \alpha_3 & \alpha_4 & & \\ & & 1-\alpha_2-\alpha_3 & \alpha_5 & & \\ & & & 1-\alpha_4-\alpha_5 & \alpha_6 & \\ & & & & 1-\alpha_6 & 1 \end{bmatrix} \begin{bmatrix} \phi_{1,r-4} \\ \phi_{1,r-3} \\ \phi_{1,r-2} \\ \phi_{1,r-1} \\ \phi_{1,r} \\ \phi_{1,r+1} \\ \phi_{1,r+2} \end{bmatrix}, \qquad (5.3.1)$$

where two–scale matrix P_0 is from (4.3.10).

Starting from a lazy scheme, we extend P_0 matrix to the square one:

$$P^w_{lift} = \begin{bmatrix} 1 & \alpha_1 & \alpha_2 & & & \\ & 1-\alpha_1 & \alpha_3 & \alpha_4 & & \\ & & 1 & & & \\ & & & 1 & & \\ & & 1-\alpha_2-\alpha_3 & \alpha_5 & \alpha_6 & \\ & & & 1-\alpha_4-\alpha_5 & 1-\alpha_6 & 1 \end{bmatrix}. \qquad (5.3.2)$$

Using lifting scheme, we have

$$\begin{bmatrix} 1 & \alpha_1 & \alpha_2 & & & \\ & 1-\alpha_1 & \alpha_3 & \alpha_4 & & \\ \mu_0 & \mu_1 & \mu_2 & \mu_3 & \mu_4 & \mu_5 \\ \nu_0 & \nu_1 & \nu_2 & \nu_3 & \nu_4 & \nu_5 \\ & & 1-\alpha_2-\alpha_3 & \alpha_5 & \alpha_6 & \\ & & & 1-\alpha_4-\alpha_5 & 1-\alpha_6 & 1 \end{bmatrix} = (\mathbf{F}_1^{-1})^T(\mathbf{F}_2^{-1})^T\mathbf{S}^T. \qquad (5.3.3)$$

Now we split the matrix above into the following three ones:

$$(\mathbf{F}_1^{-1})^T = \begin{bmatrix} 1 & & & & & \\ & 1 & & & & \\ \mu_0 & \mu_1 & 1 & 0 & \mu_2 & \mu_3 \\ & & & 1 & & \\ & & & & 1 & \\ & & & & & 1 \end{bmatrix}, \tag{5.3.4}$$

$$(\mathbf{F}_2^{-1})^T = \begin{bmatrix} 1 & & & & & \\ & 1 & & & & \\ & & 1 & & & \\ \nu_0 & \nu_1 & 0 & 1 & \nu_2 & \nu_3 \\ & & & & 1 & \\ & & & & & 1 \end{bmatrix}, \tag{5.3.5}$$

and

$$\mathbf{S}^T = \begin{bmatrix} 1 & \alpha_1 & \alpha_2 & & & \\ & 1-\alpha_1 & \alpha_3 & \alpha_4 & & \\ & & 1 & & & \\ & & & 1 & & \\ & & 1-\alpha_2-\alpha_3 & \alpha_5 & \alpha_6 & \\ & & & 1-\alpha_4-\alpha_5 & 1-\alpha_6 & 1 \end{bmatrix}. \tag{5.3.6}$$

If we define the left side of equation (5.3.2) as Φ_0^{etd}, we then get

$$\Phi_0^{etd} = (\mathbf{F}_1^{-1})^T(\mathbf{F}_2^{-1})^T\mathbf{S}^T\Phi_1 = (\mathbf{S}(\mathbf{F}_1\mathbf{F}_2)^{-1})^T\Phi_1. \tag{5.3.7}$$

Therefore,

$$((\mathbf{F}_1\mathbf{F}_2))^T\Phi_0^{etd} = \mathbf{S}^T\Phi_1. \tag{5.3.8}$$

i.e.,

$$\begin{bmatrix} \phi_{0,r-3} \\ \phi_{0,r-2} \\ \psi_{0,0} - \sum_{\imath=0}^{3} \mu_i \phi_{0,r-3+i} \\ \psi_{0,1} - \sum_{\imath=0}^{3} \nu_i \phi_{0,r-3+i} \\ \phi_{0,r-1} \\ \phi_{0,r} \end{bmatrix} = \begin{bmatrix} \phi_{1,r-4} + a_1\phi_{1,r-3} + a_2\phi_{1,r-2} \\ (1-\alpha_1)\phi_{1,r-3} + a_3\phi_{1,r-2} + a_4\phi_{1,r-1} \\ \phi_{1,r-2} \\ \phi_{1,r-1} \\ (1-a_2-a_3)\phi_{1,r-2} + a_5\phi_{1,r-1} + a_6\phi_{1,r} \\ (1-a_4-a_5)\phi_{1,r-1} + (1-a_6)\phi_{1,r} + \phi_{1,r+1} \end{bmatrix} \tag{5.3.9}$$

where we replace α_i with a for the sake of simplicity. The lifted wavelets are

$$\begin{cases} \psi_{0,0} = \phi_{1,r-2} + \sum_{i=0}^{3} \mu_i \phi_{0,r-3+i} \\ \psi_{0,1} = \phi_{1,r-1} + \sum_{i=0}^{3} \nu_i \phi_{0,r-3+i} \end{cases}. \tag{5.3.10}$$

After a series of matrix computation, we get the the following result

$$I - SF^{-1}XFS^{-1}I = S^T\mathbf{v}\mathbf{A}$$

where

$$\mathbf{v} = \begin{bmatrix} \mu_0 & \nu_0 \\ \mu_1 & \nu_1 \\ 1 & 0 \\ 0 & 1 \\ \mu_2 & \nu_2 \\ \mu_3 & \nu_3 \end{bmatrix} \tag{5.3.11}$$

and

$$A = \begin{bmatrix} \frac{a_2(1-a_2)-a_1a_3}{1-a_1} & \frac{a_3}{1-a_1} & -1 & 0 & \frac{1-a_2-a_3}{a_6} & \frac{(1-a_6)(1-a_2-a_3)}{-a_6} \\ \frac{a_1a_4}{1-a_1} & \frac{a_4}{1-a_1} & 0 & -1 & \frac{a_5}{a_6} & \frac{a_6(1-a_4)-a_5}{a_6} \end{bmatrix}. \tag{5.3.12}$$

Let

$$\eta = S^T\mathbf{v}\mathbf{A}, \tag{5.3.13}$$

to find its minimum value is equivalently to find the minimum value of the following equation

$$\|\eta^T\eta\|_{l^2} = \|A^T\mathbf{v}^T SS^T\mathbf{v}A\|_{l^2}. \tag{5.3.14}$$

If we let

$$f = \mathbf{v}^T SS^T\mathbf{v}, \tag{5.3.15}$$

the problem we are tackling becomes finding the least square value of f under the condition of one vanishing moment. Let $g(x) = \int \psi_0(x)dx = 0$, and denote

$$G(x) = f(x) - \lambda g(x), \tag{5.3.16}$$

then solving the group of equations

$$\begin{cases} \frac{\partial G}{\partial \mu_i} = 0, & i = 0, 1, 2, 3 \\ \frac{\partial G}{\partial \nu_i} = 0, & i = 0, 1, 2, 3 \\ \frac{\partial G}{\partial \lambda} = 0, & \end{cases}$$

we'll get the solutions for matrices $F1$ and $F2$, with which we obtain our NURBlets.

5.4 Vanishing Moments in NURBs

Vanishing moments play an important role in wavelet construction. In B-Splines, p vanishing moments means that wavelet coefficients for p^{th} order polynomial will be zero. This implies that any polynomial curves up to degree $p-1$ can be represented completely in the scaling space. As a result, scaling functions can represent more complex curves accurately, which is the property we desire for. For this reason, vanishing moments connect to polynomial reproduction closely in B-Splines.

However, in the case of weighted B-Spines, or NURBs, vanishing moments no longer have such a desirable property due to the fact of rational character. In our knowledge, vanishing moments are barely known in NURBs so far. Nevertheless, attaining vanishing moments in NURBlet construction is still strongly wanted in applications. Therefore, uncover their mysterious veils in NURBs becoming very overwhelming.

We'd like to approach the problem from one vanishing moment in the construction of biorthogonal NURBlets. As before, let $g(x)$ be one moment $g(x) = \int \psi(x)dx$, and m be a moment vector such that

$$m = \int \Phi dx, \tag{5.4.1}$$

then equation (2.3.12) implies

$$Qm = 0. \tag{5.4.2}$$

On the other hand, biorthogonal NURBlets should have the following constrains

$$\begin{aligned} P\widetilde{P} &= I \qquad & Q\widetilde{P} &= 0 \\ Q\widetilde{Q} &= I \qquad & P\widetilde{Q} &= 0. \end{aligned} \tag{5.4.3}$$

Bringing the constrains above into the case of single knot insertion, we observe the following properties:

1. Q is a row vector;
2. Q is in the kernel of $\widetilde{P}$;
3. m is in the range of $\widetilde{P}$.

An algorithm for biorthogonal NURBlet construction with one vanishing moment in single knot insertion is thus devised as the following:

1. Compute the moment vector $m = \int \Phi dx$;
2. Find a special solution $\widetilde{P}$ for $P\widetilde{P} = I$;
3. Find the kernel of P;
4. Get the general solution for $\widetilde{P}$;
5. Determine Q and $\widetilde{Q}$.

A corresponding example is given in the following section (see Example 3).

5.5 Examples

In this section, we'll give three examples presenting the algorithms of constructing biorthogonal NURBlets in the previous section. The first two examples use lifting scheme, while the last one is from the point of view of vanishing moments.

5.5.1 Example 1: One Knot Insertion at the Boundary

Step 1 Define knot sequence, new knot and weight vector $\mathbf{w}_0$

They are shown in Figure 5.5.

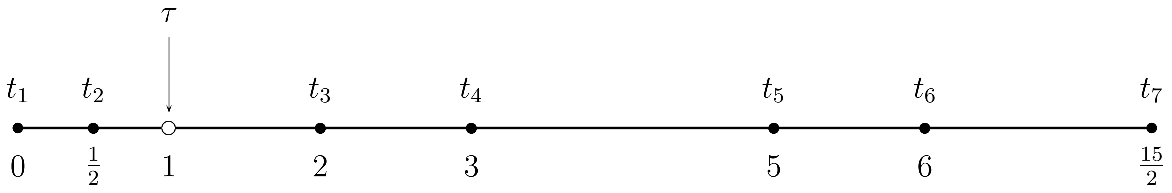


Figure 5.5: Knot sequence $\{0, 0, 0, 0, \frac{1}{2}, 2, 3, 5, 6, \frac{15}{2}, \frac{15}{2}, \frac{15}{2}, \frac{15}{2}\}$ and new knot $\tau = 1$.

$$\mathbf{w}_0 = \begin{bmatrix} \frac{1}{2} & \frac{1}{3} & 1 & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} \end{bmatrix}^T \tag{5.5.1}$$

The new knot is inserted at the boundary knot span (t_2, t_3).

Step 2 Get Bézier Nets

Based on Figure 2.8, Figure 3.2 and equation (2.1.14), Bézier nets of all cubic B-Splines at the original level and $N_{1,1}$ at the fine level are presented in Figure 5.6.

Step 3 Get the piecewise polynomial function format for the functions in step 2

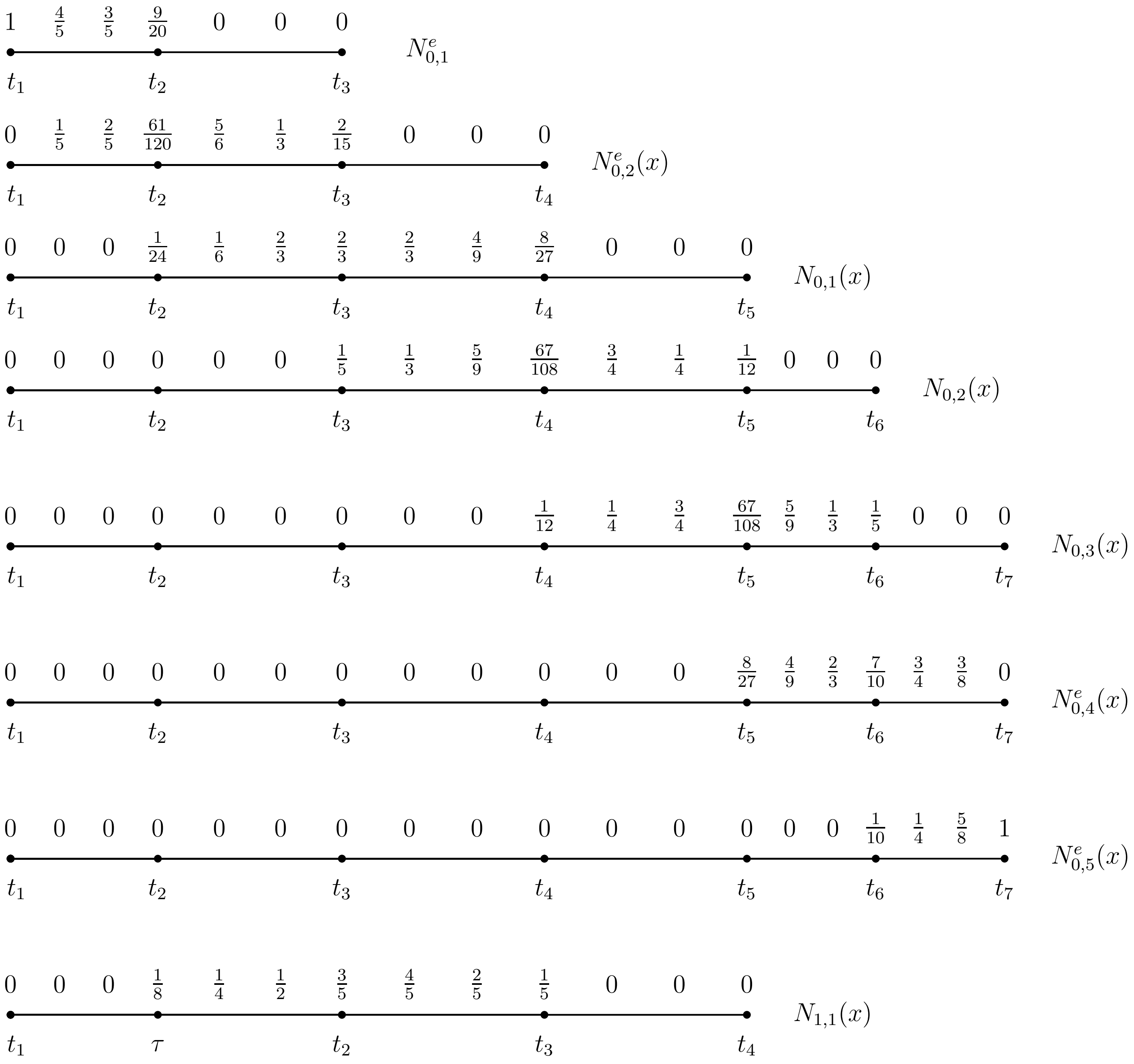


Figure 5.6: Bézier nets on interval $[0, \frac{15}{2}]$.

$$N^e_{0,1} = \begin{cases} 1 - \tfrac{6}{5}x + \tfrac{2}{5}x^3 & [0, \tfrac{1}{2}] \\ \tfrac{2}{15}(2-x)^3 & [\tfrac{1}{2}, 2] \\ 0 & \text{otherwise} \end{cases} \tag{5.5.2}$$

$$N^e_{0,2} = \begin{cases} \tfrac{6}{5}x - \tfrac{11}{15}x^3 & [0, \tfrac{1}{2}] \\ -\tfrac{2}{15} + 2x - \tfrac{8}{5}x^2 + \tfrac{1}{3}x^3 & [\tfrac{1}{2}, 2] \\ \tfrac{2}{15}(3-x)^3 & [2, 3] \\ 0 & \text{otherwise} \end{cases} \tag{5.5.3}$$

$$N_{0,1} = \begin{cases} \tfrac{1}{3}x^3 & [0, \tfrac{1}{2}] \\ \tfrac{2}{27} - \tfrac{4}{9}x + \tfrac{8}{9}x^2 - \tfrac{7}{27}x^3 & [\tfrac{1}{2}, 2] \\ -\tfrac{118}{27} + \tfrac{56}{9}x - \tfrac{22}{9}x^2 + \tfrac{8}{27}x^3 & [2, 3] \\ \tfrac{1}{27}(5-x)^3 & [3, 5] \\ 0 & \text{otherwise} \end{cases} \tag{5.5.4}$$

$$N_{0,2} = \begin{cases} \tfrac{1}{135}(2x-1)^3 & [0, \tfrac{1}{2}] \\ \tfrac{329}{135} - \tfrac{163}{45}x + \tfrac{157}{90}x^2 - \tfrac{133}{540}x^3 & [\tfrac{1}{2}, 1] \\ -\tfrac{403}{54} + \tfrac{113}{18}x - \tfrac{14}{9}x^2 + \tfrac{13}{108}x^3 & [1, 2] \\ \tfrac{1}{12}(6-x)^3 & [2, 3] \\ 0 & \text{otherwise} \end{cases} \tag{5.5.5}$$

$$N_{0,3} = \begin{cases} \tfrac{1}{12}(x-2)^3 & [2, 3] \\ \tfrac{29}{6} - \tfrac{9}{2}x + \tfrac{4}{3}x^2 - \tfrac{13}{108}x^3 & [3, 5] \\ 41 + 23x - \tfrac{25}{6}x^2 + \tfrac{133}{540}x^3 & [5, 6] \\ -\tfrac{1}{5}(\tfrac{2}{3}x - 5)^3, & [6, \tfrac{15}{2}] \\ 0 & \text{otherwise} \end{cases} \tag{5.5.6}$$

$$N^e_{0,4} = \begin{cases} \frac{8}{27}(\frac{1}{2}x - \frac{3}{2})^3, & [3,5] \\ \frac{73}{2} - \frac{43}{2}x + \frac{25}{6}x^2 - \frac{71}{270}x^3 & [5,6] \\ -\frac{95}{2} + \frac{41}{2}x - \frac{17}{6}x^2 + \frac{17}{135}x^3 & [6, \frac{15}{2}] \\ 0 & \text{otherwise} \end{cases} \tag{5.5.7}$$

$$N^e_{0,5} = \begin{cases} \frac{1}{10}(x-5)^3, & [5,6] \\ \frac{47}{2} - \frac{21}{2}x + \frac{3}{2}x^2 - \frac{1}{15}x^3 & [6, \frac{15}{2}] \\ 0 & \text{otherwise} \end{cases} \tag{5.5.8}$$

$$N_{1,1} = \begin{cases} x^3 & [0, \frac{1}{2}] \\ \frac{2}{5} - \frac{12}{5}x + \frac{24}{5}x^2 - \frac{11}{5}x^3 & [\frac{1}{2}, 1] \\ -\frac{13}{5} + \frac{33}{5}x - \frac{21}{5}x^2 + 4/5x^3 & [1,2] \\ \frac{1}{5}(3-x)^3 & [2,3] \\ 0 & \text{otherwise} \end{cases} \tag{5.5.9}$$

Step 4 Compute the denominator ω in NURBs

At a MRA level, in order to find NURBs, we need to find their common denominator — the summation of all weighted B-Splines at the same level. In fact in our modeling, all denominators of NURBs at different MRA levels are identical (see (4.1.10)). Therefore if we find the common denominator of NURBs at the original MRA level, we get the denominators of NURBs for all levels.

By (4.1.12), we have

$$\begin{aligned}
\omega &= \omega_0 \\
&= w_{0,0}N^e_{0,0} + w_{0,1}N^e_{0,1} + w_{0,2}N_{0,1} + w_{0,3}N^e_{0,2} + w_{0,3}N^e_{0,0} + w_{0,5}N^e_{0,4} + w_{0,6}N^e_{0,5} \\
&= \begin{cases}
\frac{1}{2} - \frac{1}{5}x + \frac{13}{45}x^3 & [0, \frac{1}{2}] \\
\frac{101}{180} - \frac{17}{30}x + \frac{11}{15}x^2 - \frac{1}{5}x^3 & [\frac{1}{2}, 2] \\
-\frac{97}{36} + \frac{259}{60}x - \frac{41}{24}x^2 + \frac{149}{720}x^3 & [2, 3] \\
\frac{1283}{360} - \frac{233}{120}x + \frac{17}{45}x^2 - \frac{161}{6480}x^3 & [3, 5] \\
-\frac{253}{226800}x^3 + \frac{11}{504}x^2 - \frac{17}{105}x + \frac{251}{420} & [5, 6] \\
\frac{37}{84} - \frac{1}{12}x + \frac{11}{1260}x^2 - \frac{11}{28350}x^3 & [6, \frac{15}{2}] \\
0 & \text{otherwise.}
\end{cases}
\end{aligned} \qquad (5.5.10)$$

Step 5 Compute one moment $g(x)$

(1) Compute α_i, $i = 0, 1, 2$ by (3.2.8).

(2) Construct the two-scale matrix for B-Splines on $[0, \frac{15}{2}]$ by (4.3.8).

$$P_0 = \begin{bmatrix} 1 & \frac{2}{5} & & & \\ & \frac{3}{5} & \frac{2}{3} & & \\ & & \frac{1}{3} & \frac{8}{9} & \\ & & & \frac{1}{9} & 1 \end{bmatrix}$$

(3) Based on (4.1.17), the weight vector at level 1 is attained

$$\mathbf{w}_1 = \begin{bmatrix} \frac{1}{2} & \frac{2}{5} & \frac{5}{9} & \frac{11}{12} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} \end{bmatrix}^T.$$

Hence the weight for $N_{1,1}(x)$ is $\frac{5}{9}$.

(4) Compute the one moment

$$\begin{aligned} g(x) &= \int_{\omega} (\phi_{1,1} + b_0\phi^e_{0,0} + b_1\phi^e_{0,1} + b_2\phi_{0,1} + b_3\phi_{0,2})dx \\ &= \int \omega(w_{1,1}N_{1,1} + b_0 w_{0,0}N^e_{0,0} + b_1 w_{0,1}N^e_{0,1} + b_2 w_{0,2}N_{0,1} + b_3 w_{0,3}N_{0,2})dx \\ &= \frac{924277}{3628800} + \frac{83077}{672000}b0 + \frac{2857793}{18144000}b1 + \frac{10977107}{13996800}b2 + \frac{49906487}{342921600}b3 \end{aligned} \tag{5.5.11}$$

With equations (4.3.8),(4.3.6) and (3.2.8), we get our two-scale matrix $P^w_{[t2,t3]}$ as the following:

$$\mathbf{P^w_{0[t2,t3]}} = \begin{bmatrix} 1 & \frac{1}{2} & & & \\ & \frac{1}{2} & \frac{2}{5} & & \\ & & \frac{3}{5} & \frac{32}{33} & \\ & & & \frac{1}{33} & 1 \end{bmatrix} \tag{5.5.12}$$

Step 6 Compute $f(x)$

(4.3.6), (5.2.24) and the result of (1) in step 5 result in $f(x)$ in the form of

$$f(x) = \frac{5}{4}b0^2 + \frac{1}{2}b0b1 + \frac{41}{100}b1^2 + \frac{12}{25}b1b2 + \frac{4}{5}b1 + \frac{35401}{27225}b2^2 + \frac{6}{5}b2 + 1 + \frac{64}{10892}b3 + \frac{1090}{1089}b3^2.$$

Step 7 Construct F matrix

We finally get the coefficients of F matrix by setting $G(x) = f(x) - \lambda g(x)$ and solving the group of equations of (5.2.29)

$$\begin{cases} b0 = 0.2033 \\ b1 = -0.9229 \\ b2 = -0.1776 \\ b3 = 0.0328 \end{cases}$$

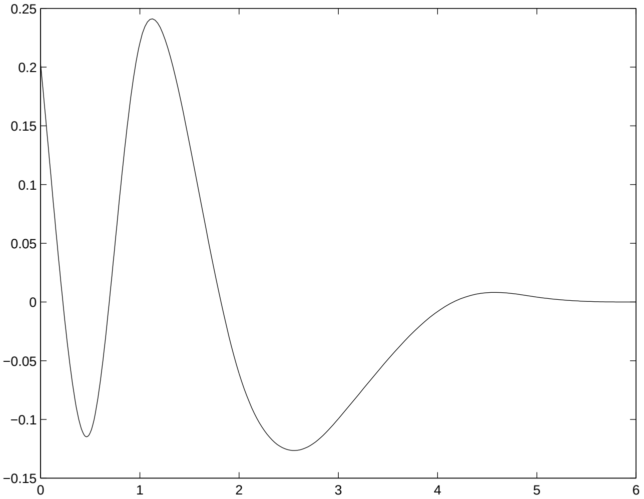


Figure 5.7: Wavelet function when insert a new knot on (t_2, t_3).

Eventually, the NURBlet function is derived

$$\psi_0(x) = \begin{cases} \frac{9.1489-44.2015x+68.6356x^3}{45-18x+26x^3} & x \in [0, \frac{1}{2}] \\ \frac{-64.5200+365.7358x-554.6657x^2+232.5060x3}{-101+102x-132x^2+36x^3} & x \in [\frac{1}{2}, 1] \\ \frac{235.4800-534.2642x+345.3343x^2-67.4940x^3}{-101+102x-132x^2+36x^3} & x \in [1, 2] \\ \frac{0.0019-0.0022x+779.0440x^2-90.0876}{-1940+3108x-1230x^2+149x^3} & x \in [2, 3] \\ \frac{0.0058-0.0036x+738.0904x^2-50.2654x^3}{-23094+12582x-2448x^2+161x^3} & x \in [3, 5] \\ \frac{185.3879(x-6)^3)}{253x^3-4950x^2+36720x-135540} & x \in [5, 6] \end{cases}$$

The graph of the wavelet is shown in Figure 5.7.

5.5.2 Example 2: One Knot Insertion in the Non-Boundary Knot Span

Given the same knot sequence as the previous example, we change new knot from $\tau = 1$ to $\tau = \frac{10}{3}$. This is the case of single knot insertion at non-boundary knot span (See Figure 5.8).

Repeat the same procedure as the previous example, we get the NURBlet as the following

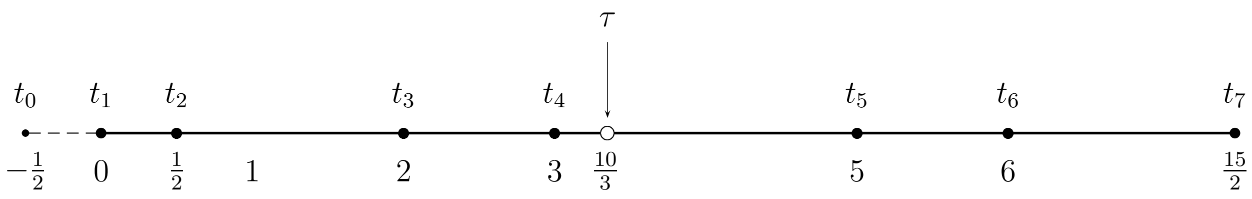


Figure 5.8: Insert a new knot on (t_4, t_5).

$$\psi_0(x) = \begin{cases} \frac{4.3562x^3}{45-18x+26x^3} & x \in [0, \frac{1}{2}] \\ \frac{-2.3507+14.1042x-28.2085x^{+}10.0933x^3}{-101+102x-132x^2+36x^3} & x \in [\frac{1}{2}, 2] \\ \frac{-0.0013+0.0019x-895.4889x^2+127.6807x^3}{-1940+3108x-1230x^2+149x^3} & x \in [2, 3] \\ \frac{-0.0009+0.0009x-0.0028x^2+0.0029x^3}{-23094+12582x-2448x^2+161x^3} & x \in [3, \frac{10}{3}] \\ \frac{0.0016-0.0099x+0.0021x^2-144.1322x^3}{-23094+12582x-2448x^2+161x^3} & x \in [\frac{10}{3}, 5] \\ \frac{-0.0002+0.0001x-0.0020x^2+0.0011x^3}{253x^3-4950x^2+36720x-135540} & x \in [5, 6] \\ \frac{0.0010-0.0041x+555.0565x^2-24.6692x^3}{-24975+4725x-495x^2+22x^3} & x \in [6, \frac{15}{2}] \\ 0 \quad \text{otherwise.} \end{cases}$$

Figure 5.9 presents its graph.

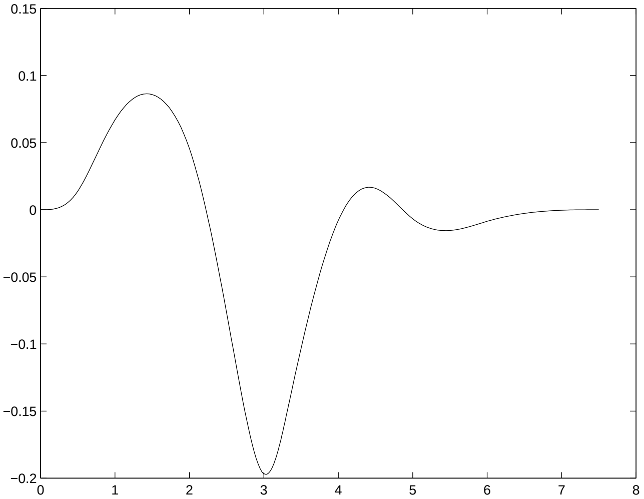


Figure 5.9: Wavelet function when insert a new knot on $(3, 5)$.

5.5.3 Example 3: Dyadic Knot Insertion on intervals

In this subsection, we'll construct biorthogonal NURBlets with one vanishing moment on an interval such that in each knot span, a dyadic knot is inserted.

On interval $[0,4]$, let ϕ_0 and ϕ_4 be edge functions, and ϕ_k $(k=1,2,3)$ be linear B-Splines as shown in Figure 5.10. Then the two-scaling matrix P is

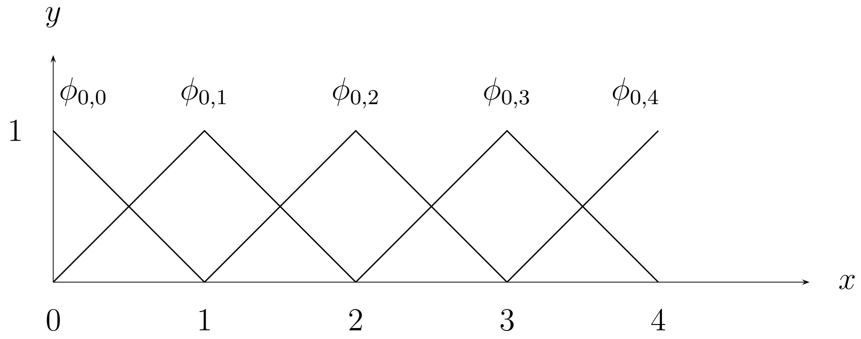


Figure 5.10: Linear B-Splines on [0,4].

$$P = \begin{bmatrix} 1 & \frac{1}{2} & & & & & & & \\ & \frac{1}{2} & 1 & \frac{1}{2} & & & & & \\ & & & \frac{1}{2} & 1 & \frac{1}{2} & & & \\ & & & & & \frac{1}{2} & 1 & \frac{1}{2} & \\ & & & & & & & \frac{1}{2} & 1 \end{bmatrix}. \tag{5.5.13}$$

P has the following quadratic polynomial character [17, 22]

$$P(z) = (\frac{1+z}{2})^2. \tag{5.5.14}$$

We desire for one vanishing moment. This means the two-scaling dual matrix $\widetilde{P}(z)$ has a factor of $1+z$ [17, **?**], i.e.,

$$\widetilde{P}(z) = (\frac{1+z}{2})\widetilde{P}_0(z). \tag{5.5.15}$$

It is well known that biorthogonal B-Splines satisfy the condition ([**?**, 17])

$$P(z)\widetilde{P}(z)+P(-z)\widetilde{P}(-z)=1.$$

Considering one vanishing moment condition in (5.5.15), the following equation is derived:

$$(\frac{1+z}{2})^3\widetilde{P}_0(z)+(\frac{1-z}{2})^3\widetilde{P}_0(-z)=1.$$

To find $\widetilde{P}_0(z)$, we extend the polynomial $(\frac{1+z}{2}+\frac{1-z}{2})^5=1$ and compare it with the equation above. This gives us a solution of $\widetilde{P}_0(z)$:

$$\begin{aligned}\widetilde{P}_0(z) &= (\frac{1+z}{2})^2+5(\frac{1+z}{2})(\frac{1-z}{2})+10(\frac{1-z}{2})^2\\ &= 2-\frac{1}{4}z-\frac{3}{2}z^2+\frac{3}{4}z^3.\end{aligned}$$

Satisfying the biorthogonal wavelet condition in (2.4.9), we find the matrices of $\widetilde{P}$, Q and $\widetilde{Q}$ matrix as the following:

$$\widetilde{P}=\begin{bmatrix} 1 & -1 & \frac{3}{4} & & \\ & 2 & -\frac{3}{2} & & \\ & & -\frac{1}{4} & \frac{3}{4} & \\ & & 2 & -\frac{3}{2} & \\ & & & -\frac{1}{4} & \frac{3}{2} \\ & & & 2 & -3 \\ & & & & 1 \\ & & & & 1 \\ & & & & \frac{1}{2} \end{bmatrix} \tag{5.5.16}$$

$$Q=\begin{bmatrix} 0 & 0 & 2 & \frac{1}{4} & -\frac{3}{2} & -\frac{3}{4} & & & \\ & & & & 2 & \frac{1}{4} & -\frac{3}{2} & -\frac{3}{4} & \\ & & & & & & 2 & -1 & -2 \\ & & & & & & & -\frac{1}{2} & 1 \end{bmatrix} \tag{5.5.17}$$

$$\widetilde{Q} = \begin{bmatrix} \frac{1}{2} & & & \\ -1 & & & \\ \frac{1}{2} & \frac{1}{2} & & \\ & -1 & & \\ & \frac{1}{2} & \frac{1}{2} & \\ & & -1 & \\ & & \frac{1}{2} & \frac{1}{2} \\ & & & -1 \\ & & & \frac{1}{2} \end{bmatrix} \tag{5.5.18}$$

Figure 5.11 shows the graphs of these biorthogonal wavelets.

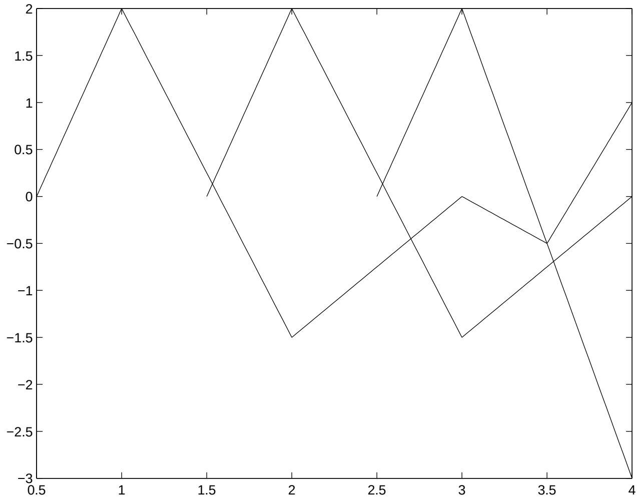


Figure 5.11: Linear biorthogonal wavelets on the interval.

Chapter 6

Conclusions and Future Work

This dissertation has presented the construction of non-uniform rational cubic B-Splines (NURBs) and biorthogonal wavelets based on NURBs with one vanishing moment on intervals.

We've constructed natural cubic B-Splines as edge functions, and found two–scale matrices for one-knot insertion and arbitrary knot insertion at each knot span respectively. With the help of lifting scheme, we then have constructed cubic biorthogonal NURBlets with one vanishing moment for one-knot insertion and two-knot insertion. Finally, we explore one vanishing moment properties.

6.1 Outline of Future Work

Vanishing moments play a very important role in wavelet construction. They indicate how smooth of an approximation that a wavelet tool gives. The higher vanishing moments are, the smaller support would be, and the better smoothness an approximation would achieve. Three vanishing moments lead to C^2 smoothness which is desired in industrial applications. Vanishing moments are related to polynomial reproduction in B-Splines. However, in non-uniform rational B-Splines, rational functions replace the polynomial ones. So there is no such thing as "polynomial reproduction" in rational B-splines. Do "vanishing moments" still exist? If so, what kind of properties do they have? In the polynomial case, n vanishing moments means the existence of factor $(1+z)^n$ in the dual P matrix. In rational case, is there any factor that must exist in the dual P matrix also? What is it then? All these are barely known to us.

Our next step is to explore such properties related to “vanishing moments” in NURBlets.

Secondly, constructing biorthogonal wavelets for cubic NURBs on intervals in 2D is desirable in real applications. With tensor products, these wavelet tools extensively describe surfaces in CAD/CAM/CAE and many other areas.

Finally, the construction of tight frame wavelets for cubic NURBs on intervals. Tight frame wavelets provide more flexibility and features such as local support than other NURBs-based wavelets. The key part in tight frame construction using NURBs is to find the vanishing moment recovery matrix [17]. With this wavelet tool, free-form surface design will have mathematical exactness, achieve C^2 smoothness and interpolation at the boundaries, which will have great and extensive applications in the industry, from fuselage design, image processing, data mining, to medical research, etc.

Bibliography

[1] T. Abdukirim, K. Niijima, S. Takano, *Design of biorthogonal wavelet filters using dyadic lifting scheme*. Bulletin of Informatics and Cybernetics, **37**(2005), 123-136.

[2] H. Akima, *A new method of interpolation and smooth curve fitting based on local procedures*. Journal of the ACM, **17**(4)(1970), 589 - 602.

[3] A. Aldroubi, M. Unser, *Discrete spline filters for multiresolution and wavelets of L_2*. SIAM J. Math. Anal., **25**(5)(1994), 1412-1432.

[4] N. Anantakrishnan, L. Piegl, *Integer de Casteljau Algorithm for Rasterizing NURBS Curves*. Comp. Graph. Forum, **11**(2)(1992), 151-162.

[5] N. Anantakrishnan and L. Piegl, *Integer subdivision algorithm for rendering NURBS curves*. The Visual Comp., **8**(3)(1992), 149-161.

[6] R. Bartels, J. Beatty, B. Barsky, *An Introduction to splines for use in Computer Graphics and Geometric Modeling*. Morgan kaufmann publisher, Inc, California, 1987.

[7] M. Bertram, M. Duchaineau, B. Hamann, K. Joy, *Generalized B-Spline Subdivision-Surface Wavelets for Geometry Compression*, IEEE Trans. on Visualization and Computer Graphics **10**(3)(2004), 326-338.

[8] M. Bertram, *Single-knot wavelets for non-uniform B-Splines*, Comp. Aided Geom. Design, **22**(2005), 849-864.

[9] W. Böehm, G. Farin, J. Kahmann, *A survey of curve and surface methods in CAGD*. Comp. Aided Geom. Design, **1**(1)(1984), 1-60.

[10] W. Böehm, *Inserting new knots into B-Spline curves*, Comp. Aided Design, **12**(4)(1980), 199-201 bf 26(2009), 472-479.

[11] G. Casciola, L. Romani, *A general matrix representation for non-uniform B-Spline subdivision with boundary control*, ALMA-DL, Digital Library of the University of Bologna, AMS Acta c.i. 2532.

[12] T. Cashman, N. dodgson, M. Sabin, *Selective knot insertion for symmetric, non-uniform refine and smooth B-Spline subdivision*, Comp. Aided Geom. Design, **26**(4)(2009), 472-479.

[13] C. Chui, *Multivariate Splines.* SIAM, Philadelphia, 1988.

[14] C. Chui, *An Introduction to Wavelets.* Academic Press, 1992.

[15] C. Chui, *Wavelets: A Mathematical Tool for Signal Analysis.* SIAM, Philadelphia, 1997.

[16] C. Chui, W. He, *Compactly supported tight frames associated with refinable functions.* Appl. and Comp. Harmonic Anal., **8**(2000), 293-319.

[17] C. Chui, W. He, J. Stöckler, *Nonstationary tight wavelet frames, I: Bounded intervals.* Appl. and Comp. Harmonic Anal., **17**(2004), 141-197.

[18] C. Chui, J. Lian, *Multilevel structure of Nurbs and formulation of Nurblets.* Wavelet Analysis: Twenty Years' Developments, **1**(2002), 23-38.

[19] C. Chui, J. Villiers, *Spline-wavelets with arbitrary knots on a bounded interval: orthogonal decomposition and computational algorithms.* Communications in Applied Analysis. **2**(4)(1998), 457-486.

[20] M. Coffey, D. Etter, *Multiresolution analysis on bounded domains for the design of biorthogonal wavelet bases*, Signal Processing, IEEE Transactions on, **50**(3)(2002), 509-519.

[21] A. Cohen, I. Daubechies, P. Vial, *Wavelets on the interval and fast wavelet transforms.* Appl. and Comp. Harmonic Anal., **1**(1993), 54-81.

[22] I. Daubechies, *Ten lectures on Wavelets*. SIAM, (1992).

[23] I. Daubechiles, W. Sweildens, *Factoring wavelet transforms into lifting steps*, J. Fourier Anal. Appl., **4**(3)(1998), 247-269.

[24] M. Duchaineau, *Dyadic splines.* Ph.D. thesis, Department of Computer Science, University of California, Davis.

[25] C. de Boor, *On calculating with B-Splines.* J. Approx. Theory, **6**(1)(1972), 50-62

[26] C. de Boor, *Splines as linear combinations of B-Splines. A Survey.* Approx. Theory, **II**, Academic Press, New York, 1976.

[27] C. de Boor, *A practical guide to splines.* Springer, Berlin, 1978.

[28] C. de Boor, *B-form basics. In: Geometric Modeling: Algorithms and New Trends.* SIAM, 1987.

[29] C. de Boor, *Fix functionals and polar forms.* Comp. Aided Geom. Design., **7**(1990), 425-430

[30] W. Dong, G. Shi, J. Xu, *Signal-adapted directional lifting scheme for image compression*, International Symposium on Circuits and Systems (ISCAS 2008). Seattle, USA. IEEE 2008, 1392-1395.

[31] G. Elber, *Multiresolution Curve Editing with Linear Constraints*, 6th AMC/IEEE Symposium on Solid Modeling and Applications, (2001), 109-119

[32] G. Elber, C. Gotsman, *Multiresolution Control for Nonuniform B-Spline Curve Editing*, The Third Pacific Graphics Conference on Computer Graphics and Applications, Seoul, Korea (1995), 267-278.

[33] G. Farin, *From Conics to NURBS: A tutorial and Survey.* IEEE Comp. Graph. and Appl., **12**(5)(1992), 78-86.

[34] A. Finkkelstein and D. Salesin, *Multiresolution Curves*, Comput. Graph., ACM SIGGRAPH, (1994), 261-268.

[35] M. Floater, *Derivatives of rational Bézier curves.* Comp. Aided Geom. Design, **9**(1992), 161-174.

[36] H. Feng, L. Guo, J. Xiao, *The lifting scheme based on the second generation wavelets*, Wuhan Uni. J. of Natural Sci., **11**(3)(2006), 503-506.

[37] R. Goldenthal, M. Bercovier, *Spline Curve Approximation and Design by Optimal Control over the Knots*, Geometric Modelling dagstuhl, **72**(1-2)(2004), 53-64

[38] Björn Jawerth, Wim Sweldens, *An overview of wavelet based multiresolution analyses.* SIAM Review, **36**(3)(1994), 377 - 412.

[39] E. Lee, *Rational quadratic Bézier representation for conics*, Geometric Modeling: Algorithms and New Trends, Farin, G.E., Philadelphia: SIAM (1987), 3-19.

[40] H. Li, Q. Wang, L. Wu, *A novel design of lifting scheme from general wavelet.* IEEE Trans. on Signal Procesing, **49**(8)(2001), 1714-1717.

[41] D. Li, K. Qin, and H. Sun, *Curve Modeling with Constrained B-Spline wavelets*, Pacific Conference on Computer Graphics Applications (2004), 25-33

[42] T. Lyche, K. Mørken, and E. Quak, *Theory and Algorithms for Non-Uniform Spline Wavelets.* Multivarite Approximation and Applications, Cambridge University Press, 2001.

[43] T. Lyche, K. Mørken, E. Quak, *Theory and algorithms for non-uniform spline wavelets.* In: Multivariate Approximation and Applications. Cambridge University Press, Cambridge, 152-187.

[44] J. Maes, A. Bultheel, *Stable lifting construction of Non-uniform biorthogonal spline wavelets with compact support.* Electonic Transactions on Numerical Analysis, **25**(2006),224-258.

[45] J. Maes, A. Bultheel, *Easy construction of non-uniform biorthogonal spline wavelets with compact support*, International conference on numerical analysis and applied math., (2005), 357-360.

[46] S. Mallat, *A theory for multiresolution signal decomposition: the wavelet representation.* IEEE Trans. on Pat. Anal. and Mach. Intel, **11**(7)(1989), 674 -693.

[47] S. Mallat, *Multiresolution approximations and wavelet orthonormal bases of* $L^2(R)$. Trans. Amer. Math. Soc., **315**(1)(1989), 69-87.

[48] M. Miura, H. Masuda, *Selective non-uniform subdivision*, Proceedings of 10th pacific conference on comp. graphics and applications, 457.

[49] R. Pan, Z. Yao, *Biorthogonal nonuniform B-Spline wavelets based on a discrete norm*, Comp. Aided Geom.Design, **26**4(2009), 480-492.

[50] L. Piegl, *Integer de Casteljau Algorithm for Rasterizing NURBS Curves.* Comput. Graph. Forum, **11**(2)(1992), 151-162.

[51] L. Piegl, W. Tiller, *The Nurbs Book*, Springer, 2nd Edition, (1997).

[52] L. Piegl, *Knowledge-guided NURBS: principles and architecture.* Computer-Aided Design & Appli., **3**(6)(2006), 719-129.

[53] L. Piegl, W. Tiller, *Symbolic operators for NURBS.* Computer-Aided Design, **29**(5)(1997), 361-368.

[54] L. Piegl, W. Tiller, *Computing the derivative of NURBS with respect to a knot.* Comp. Aided Geom. Design, **15**(9)(1998), 925-934.

[55] L. Piegl, W. Tiller, *Biarc approximation of NURBS curves.* Computer-Aided Design, **34**(11)(2002), 807-814.

[56] E. Quak, *Non-uniform B-Splines and B-wavelets.* In: Tutorial on Multiresolution in Geometric Modeling. Springer, Berlin, 101-146.

[57] M. Reimers, *A local construction for nonuniform spline tight wavelet frames.* Approximation Theory XII, San Antonio, 2008.

[58] S. Schaefer, R. Goldman, *Non-uniform subdivision for B-Splines of arbitrary degree*, Comp. Aided Geom. Design, **26**1(2009), 75-81.

[59] L. Schumaker, *Spline functions basic theory.* John wiley & sons, Inc. (1981)

[60] H. Seide, *A new multiaffine approach to B-Splines*, Comp. Aided Geom. Design, **6**(1989), 23-32.

[61] P. Shui, Z. Bao, *M-Band biorthogonal interpolating wavelets via lifting scheme*, **52**(9)(2004), 2500-2512

[62] E. Stollnitz, T. Derose, D. Salesin, *Wavelets for Computer Graphics.* MKP,San Francisco, 1996.

[63] W. Sweldens, *The lifting Scheme: A custom-design construction of biorthogonal wavelets*, Appl. comput. Harmonic Anal., **3**(2)(1996), 186-200.

[64] W, Sweldens, *Wavelets and the lifting scheme: a 5 minute tour*, Z. Angew. Math. Mech., **76**(Suppl. 2)(1996), 41–44.

[65] W. Sweldens, *The lifting scheme: A construction of second generation wavelets*, SIAM J. Math. Anal., **29**(2)(1998), 511-546.

[66] C. Vonesch, T. Blu, M. Unser, *Generalized Daubechies wavelet families*, IEEE Trans. on Signal Processing, **55**(9)(2007), 4415-4429.