CHAPTER
TWO

B-Spline Basis Functions

2.1 Introduction

Curves consisting of just one polynomial or rational segment are often inadequate. Their shortcomings are:

- a high degree is required in order to satisfy a large number of constraints; e.g., $(n-1)$-degree is needed to pass a polynomial Bézier curve through n data points. However, high degree curves are inefficient to process and are numerically unstable;
- a high degree is required to accurately fit some complex shapes;
- single-segment curves (surfaces) are not well-suited to interactive shape design; although Bézier curves can be shaped by means of their control points (and weights), the control is not sufficiently local.

The solution is to use curves (surfaces) which are *piecewise polynomial*, or *piecewise rational.* Figure 2.1 shows a curve, $\mathbf{C}(u)$, consisting of m $(=3)$ nth-degree polynomial *segments.* $\mathbf{C}(u)$ is defined on $u \in [0,1]$. The parameter values $u_0 = 0 < u_1 < u_2 < u_3 = 1$ are called *breakpoints.* They map into the endpoints of the three polynomial segments. We denote the segments by $\mathbf{C}_i(u)$, $1 \le i \le m$. The segments are constructed so that they join with some level of continuity (not necessarily the same at every breakpoint). Let $\mathbf{C}_i^{(j)}$ denote the jth derivative of $\mathbf{C}_i$. $\mathbf{C}(u)$ is said to be C^k continuous at the breakpoint u_i if $\mathbf{C}_i^{(j)}(u_i) = \mathbf{C}_{i+1}^{(j)}(u_i)$ for all $0 \le j \le k$.

Any of the standard polynomial forms can be used to represent $\mathbf{C}_i(u)$. Figure 2.2 shows the curve of Figure 2.1 with the three segments in cubic Bézier form. $\mathbf{P}_i^j$ denotes the ith control point of the jth segment.

If the degree equals three and the breakpoints $U = \{u_0, u_1, u_2, u_3\}$ remain fixed, and if we allow the twelve control points, $\mathbf{P}_i^j$, to vary arbitrarily, we obtain the vector space, $\mathcal{V}$, consisting of all piecewise cubic polynomial curves

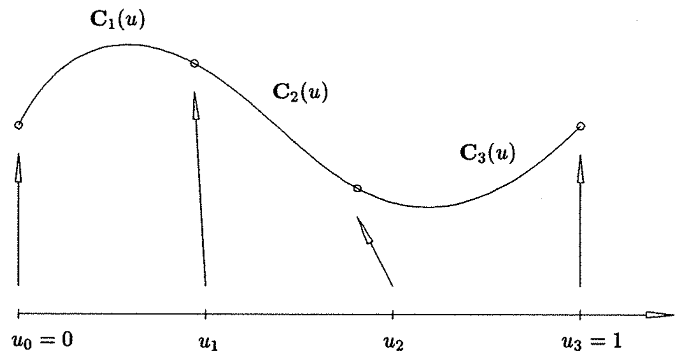


Figure 2.1. A piecewise cubic polynomial curve with three segments.

on U. $\mathcal{V}$ has dimension twelve, and a curve in $\mathcal{V}$ may be discontinuous at u_1 or u_2. Now suppose we specify (as in Figure 2.2) that $\mathbf{P}^1_3 = \mathbf{P}^2_0$ and $\mathbf{P}^2_3 = \mathbf{P}^3_0$. This gives rise to $\mathcal{V}^0$, the vector space of all piecewise cubic polynomial curves on U which are at least C^0 continuous everywhere. $\mathcal{V}^0$ has dimension ten, and $\mathcal{V}^0 \subset \mathcal{V}$.

Imposing C^1 continuity is a bit more involved. Let us consider $u = u_1$. Assume that $\mathbf{P}^1_3 = \mathbf{P}^2_0$. Let

$$v = \frac{u - u_0}{u_1 - u_0} \quad \text{and} \quad w = \frac{u - u_1}{u_2 - u_1}$$

be local parameters on the intervals $[u_0, u_1]$ and $[u_1, u_2]$, respectively. Then $0 \le v,\ w \le 1$. C^1 continuity at u_1 implies

$$\frac{1}{u_1 - u_0}\,\mathbf{C}^{(1)}_1(v = 1) = \mathbf{C}^{(1)}_1(u_1) = \mathbf{C}^{(1)}_2(u_1) = \frac{1}{u_2 - u_1}\,\mathbf{C}^{(1)}_2(w = 0)$$

and from Eq. (1.10) it follows that

$$\frac{3}{u_1 - u_0}(\mathbf{P}^1_3 - \mathbf{P}^1_2) = \frac{3}{u_2 - u_1}(\mathbf{P}^2_1 - \mathbf{P}^2_0)$$

Thus

$$\mathbf{P}^1_3 = \frac{(u_2 - u_1)\mathbf{P}^1_2 + (u_1 - u_0)\mathbf{P}^2_1}{u_2 - u_0} \tag{2.1}$$

Equation (2.1) says that $\mathbf{P}^1_3$ and $\mathbf{P}^2_3$ can be written in terms of $\mathbf{P}^1_2, \mathbf{P}^2_1$ and $\mathbf{P}^2_2, \mathbf{P}^3_1$, respectively. Hence, $\mathcal{V}^1$, the vector space of all C^1 continuous piecewise cubic polynomial curves on U, has dimension eight, and $\mathcal{V}^1 \subset \mathcal{V}^0 \subset \mathcal{V}$.

This makes it clear that storing and manipulating the individual polynomial segments of a piecewise polynomial curve is not the ideal method for handling

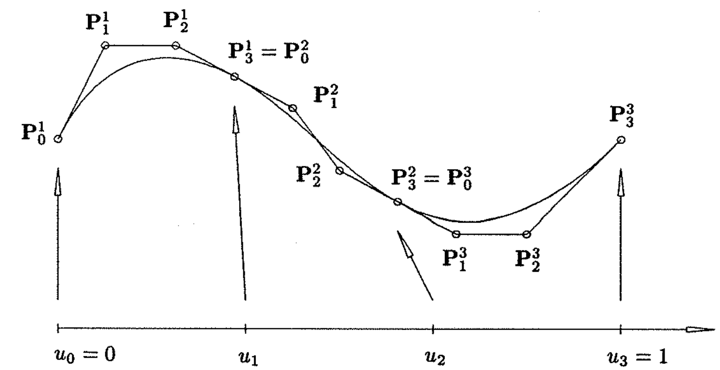


Figure 2.2. The curve of Figure 2.1 shown with the polynomial segments represented in Bézier form.

such curves. First, redundant data must be stored: twelve coefficients, where only eight are required for C^1 continuous cubic curves, and only six for C^2 continuous cubic curves. Second, for the Bézier form the continuity of $\mathbf{C}(u)$ depends on the positions of the control points, hence there is little flexibility in positioning control points while maintaining continuity. If a designer wants C^1 continuity and is satisfied with the segments $\mathbf{C}_1(u)$ and $\mathbf{C}_3(u)$, but wants to modify the shape of $\mathbf{C}_2(u)$, he is out of luck: none of $\mathbf{C}_2(u)$'s control points can be modified. Third, determining the continuity of a curve requires computation (such as Eq. [2.1]).

We want a curve representation of the form

$$\mathbf{C}(u) = \sum_{i=0}^{n} f_i(u)\,\mathbf{P}_i \tag{2.2}$$

where the $\mathbf{P}_i$ are *control points*, and the $\{f_i(u),\ i = 0, \ldots, n\}$ are *piecewise polynomial functions* forming a basis for the vector space of all piecewise polynomial functions of the desired degree and continuity (for a fixed breakpoint sequence, $U = \{u_i\},\ 0 \le i \le m$). Note that continuity is determined by the basis functions, hence the control points can be modified without altering the curve's continuity. Furthermore, the $\{f_i\}$ should have the 'usual' nice analytic properties, e.g. those listed in Section 1.3. This ensures that the curves defined by Eq. (2.2) have nice geometric properties similar to Bézier curves, e.g., convex hull, variation diminishing, transformation invariance. Another important property that we seek in our basis functions is that of *local support*; this implies that each $f_i(u)$ is nonzero only on a limited number of subintervals, not the entire domain, $[u_0, u_m]$. Since $\mathbf{P}_i$ is multiplied by $f_i(u)$, moving $\mathbf{P}_i$ affects curve shape only on the subintervals where $f_i(u)$ is nonzero.

Finally, given appropriate piecewise polynomial basis functions, we can construct piecewise rational curves

$$\mathbf{C}^w(u) = \sum_{i=0}^{n} f_i(u)\, \mathbf{P}_i^w \tag{2.3}$$

and nonrational and rational tensor product surfaces

$$\mathbf{S}(u,v) = \sum_{i=0}^{n} \sum_{j=0}^{m} f_i(u) g_j(v)\, \mathbf{P}_{i,j}$$

$$\mathbf{S}^w(u,v) = \sum_{i=0}^{n} \sum_{j=0}^{m} f_i(u) g_j(v)\, \mathbf{P}_{i,j}^w \tag{2.4}$$

For the remainder of this chapter we study the so-called B-spline basis functions. In Chapters 3 and 4 we combine these functions with three-dimensional and four-dimensional control points to obtain nonrational and rational curves and surfaces, respectively.

2.2 Definition and Properties of B-spline Basis Functions

There are a number of ways to define the B-spline basis functions and to prove their important properties, e.g., by divided differences of truncated power functions [Curr47; Scho46], by blossoming [Rams87], and by a recurrence formula due to deBoor, Cox, and Mansfield [Cox72; DeBo72, 78]. We use the recurrence formula, since it is the most useful for computer implementation.

Let $U = \{u_0, \ldots, u_m\}$ be a nondecreasing sequence of real numbers, i.e., $u_i \leq u_{i+1}$, $i = 0, \ldots, m-1$. The u_i are called *knots*, and U is the *knot vector*. The ith B-spline basis function of p-degree (*order* $p+1$), denoted by $N_{i,p}(u)$, is defined as

$$N_{i,0}(u) = \begin{cases} 1 & \text{if } u_i \leq u < u_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u) \tag{2.5}$$

Note that

- $N_{i,0}(u)$ is a step function, equal to zero everywhere except on the half-open interval $u \in [u_i, u_{i+1})$;
- for $p > 0$, $N_{i,p}(u)$ is a linear combination of two $(p-1)$-degree basis functions (Figure 2.3);

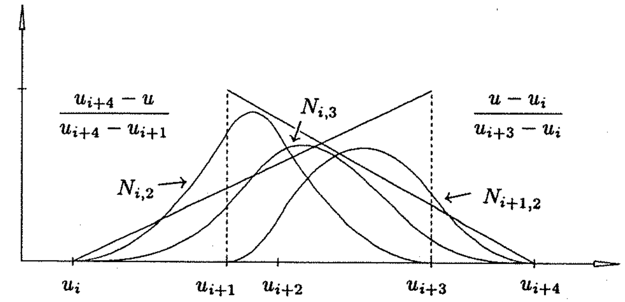


Figure 2.3. The recursive definition of B-spline basis functions.

- computation of a set of basis functions requires specification of a knot vector, U, and the degree, p;
- Equation (2.5) can yield the quotient $0/0$ (see examples later); we define this quotient to be zero;
- the $N_{i,p}(u)$ are piecewise polynomials, defined on the entire real line; generally only the interval $[u_0, u_m]$ is of interest;
- the half-open interval, $[u_i, u_{i+1})$, is called the *ith knot span*; it can have zero length, since knots need not be distinct;
- the computation of the pth-degree functions generates a truncated triangular table

$$\begin{array}{cccc}
N_{0,0} & & & \\
 & N_{0,1} & & \\
N_{1,0} & & N_{0,2} & \\
 & N_{1,1} & & N_{0,3} \\
N_{2,0} & & N_{1,2} & \\
 & N_{2,1} & & N_{1,3} \\
N_{3,0} & & N_{2,2} & \vdots \\
 & N_{3,1} & \vdots & \\
N_{4,0} & \vdots & & \\
\vdots & & &
\end{array}$$

For brevity we often write $N_{i,p}$ instead of $N_{i,p}(u)$.

A word about terminology. In Section 2.1 we used the term breakpoint and required $u_i < u_{i+1}$ for all i. In the remainder of this book we use the term knot and assume $u_i \leq u_{i+1}$. The breakpoints correspond to the set of *distinct* knot values, and the knot spans of nonzero length define the individual polynomial segments. Hence, we use the word knot with two different meanings: a distinct value (breakpoint) in the set U, and an element of the set U (there can exist additional knots in U having the same value). It should be clear from the context which meaning is intended.

Examples

Ex2.1 Let $U = \{u_0 = 0, u_1 = 0, u_2 = 0, u_3 = 1, u_4 = 1, u_5 = 1\}$ and $p = 2$. We now compute the B-spline basis functions of degrees 0, 1, and 2

$$N_{0,0} = N_{1,0} = 0 \qquad -\infty < u < \infty$$

$$N_{2,0} = \begin{cases} 1 & 0 \le u < 1 \\ 0 & \text{otherwise} \end{cases}$$

$$N_{3,0} = N_{4,0} = 0 \qquad -\infty < u < \infty$$

$$N_{0,1} = \frac{u-0}{0-0}N_{0,0} + \frac{0-u}{0-0}N_{1,0} = 0 \qquad -\infty < u < \infty$$

$$N_{1,1} = \frac{u-0}{0-0}N_{1,0} + \frac{1-u}{1-0}N_{2,0} = \begin{cases} 1-u & 0 \le u < 1 \\ 0 & \text{otherwise} \end{cases}$$

$$N_{2,1} = \frac{u-0}{1-0}N_{2,0} + \frac{1-u}{1-1}N_{3,0} = \begin{cases} u & 0 \le u < 1 \\ 0 & \text{otherwise} \end{cases}$$

$$N_{3,1} = \frac{u-1}{1-1}N_{3,0} + \frac{1-u}{1-1}N_{4,0} = 0 \qquad -\infty < u < \infty$$

$$N_{0,2} = \frac{u-0}{0-0}N_{0,1} + \frac{1-u}{1-0}N_{1,1} = \begin{cases} (1-u)^2 & 0 \le u < 1 \\ 0 & \text{otherwise} \end{cases}$$

$$N_{1,2} = \frac{u-0}{1-0}N_{1,1} + \frac{1-u}{1-0}N_{2,1} = \begin{cases} 2u(1-u) & 0 \le u < 1 \\ 0 & \text{otherwise} \end{cases}$$

$$N_{2,2} = \frac{u-0}{1-0}N_{2,1} + \frac{1-u}{1-1}N_{3,1} = \begin{cases} u^2 & 0 \le u < 1 \\ 0 & \text{otherwise} \end{cases}$$

Note that the $N_{i,2}$, restricted to the interval $u \in [0,1]$, are the quadratic Bernstein polynomials (Section 1.3 and Figure 1.13b). For this reason, the B-spline representation with a knot vector of the form

$$U = \{\underbrace{0,\ldots,0}_{p+1},\underbrace{1,\ldots,1}_{p+1}\}$$

is a generalization of the Bézier representation.

Ex2.2 Let $U = \{u_0 = 0, u_1 = 0, u_2 = 0, u_3 = 1, u_4 = 2, u_5 = 3, u_6 = 4, u_7 = 4, u_8 = 5, u_9 = 5, u_{10} = 5\}$ and $p = 2$. The zeroth-, first-, and second-degree basis functions are computed here. The ones not identically zero are shown in Figures 2.4, 2.5, and 2.6, respectively

$$N_{0,0} = N_{1,0} = 0 \quad \text{for } -\infty < u < \infty$$

$$N_{2,0} = \begin{cases} 1 & 0 \le u < 1 \\ 0 & \text{otherwise} \end{cases}$$

$$N_{3,0} = \begin{cases} 1 & 1 \le u < 2 \\ 0 & \text{otherwise} \end{cases}$$

$$N_{4,0} = \begin{cases} 1 & 2 \le u < 3 \\ 0 & \text{otherwise} \end{cases}$$

$$N_{5,0} = \begin{cases} 1 & 3 \le u < 4 \\ 0 & \text{otherwise} \end{cases}$$

$$N_{6,0} = 0 \quad \text{for } -\infty < u < \infty$$

$$N_{7,0} = \begin{cases} 1 & 4 \le u < 5 \\ 0 & \text{otherwise} \end{cases}$$

$$N_{8,0} = N_{9,0} = 0 \quad \text{for } -\infty < u < \infty$$

$$N_{0,1} = \frac{u-0}{0-0}N_{0,0} + \frac{0-u}{0-0}N_{1,0} = 0 \qquad -\infty < u < \infty$$

$$N_{1,1} = \frac{u-0}{0-0}N_{1,0} + \frac{1-u}{1-0}N_{2,0} = \begin{cases} 1-u & 0 \le u < 1 \\ 0 & \text{otherwise} \end{cases}$$

$$N_{2,1} = \frac{u-0}{1-0}N_{2,0} + \frac{2-u}{2-1}N_{3,0} = \begin{cases} u & 0 \le u < 1 \\ 2-u & 1 \le u < 2 \\ 0 & \text{otherwise} \end{cases}$$

$$N_{3,1} = \frac{u-1}{2-1}N_{3,0} + \frac{3-u}{3-2}N_{4,0} = \begin{cases} u-1 & 1 \le u < 2 \\ 3-u & 2 \le u < 3 \\ 0 & \text{otherwise} \end{cases}$$

$$N_{4,1} = \frac{u-2}{3-2}N_{4,0} + \frac{4-u}{4-3}N_{5,0} = \begin{cases} u-2 & 2 \le u < 3 \\ 4-u & 3 \le u < 4 \\ 0 & \text{otherwise} \end{cases}$$

$$N_{5,1} = \frac{u-3}{4-3}N_{5,0} + \frac{4-u}{4-4}N_{6,0} = \begin{cases} u-3 & 3 \le u < 4 \\ 0 & \text{otherwise} \end{cases}$$

$$N_{6,1} = \frac{u-4}{4-4}N_{6,0} + \frac{5-u}{5-4}N_{7,0} = \begin{cases} 5-u & 4 \le u < 5 \\ 0 & \text{otherwise} \end{cases}$$

$$N_{7,1} = \frac{u-4}{5-4}N_{7,0} + \frac{5-u}{5-5}N_{8,0} = \begin{cases} u-4 & 4 \le u < 5 \\ 0 & \text{otherwise} \end{cases}$$

$$N_{8,1} = \frac{u-5}{5-5}N_{8,0} + \frac{5-u}{5-5}N_{9,0} = 0 \qquad -\infty < u < \infty$$

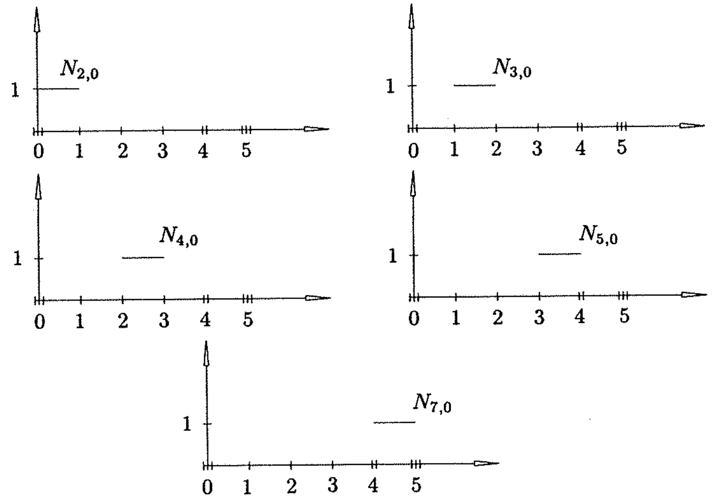


Figure 2.4. The nonzero zeroth-degree basis functions, $U = \{0,0,0,1,2,3,4,4,5,5,5\}$.

All of the following $N_{i,2}$ are zero everywhere except on the specified intervals, that is

$$N_{0,2} = \frac{u-0}{0-0}N_{0,1} + \frac{1-u}{1-0}N_{1,1} = (1-u)^2 \qquad 0 \le u < 1$$

$$N_{1,2} = \frac{u-0}{1-0}N_{1,1} + \frac{2-u}{2-0}N_{2,1} = \begin{cases} 2u - {}^{3}\!/_{2}u^2 & 0 \le u < 1 \\ {}^{1}\!/_{2}(2-u)^2 & 1 \le u < 2 \end{cases}$$

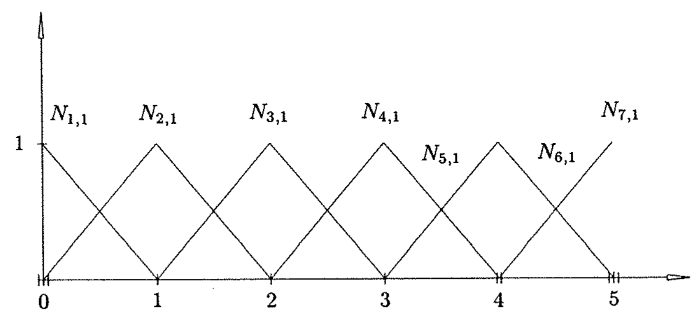


Figure 2.5. The nonzero first-degree basis functions, $U = \{0,0,0,1,2,3,4,4,5,5,5\}$.

$$N_{2,2} = \frac{u-0}{2-0}N_{2,1} + \frac{3-u}{3-1}N_{3,1} = \begin{cases} {}^{1}\!/_{2}u^2 & 0 \le u < 1 \\ -{}^{3}\!/_{2} + 3u - u^2 & 1 \le u < 2 \\ {}^{1}\!/_{2}(3-u)^2 & 2 \le u < 3 \end{cases}$$

$$N_{3,2} = \frac{u-1}{3-1}N_{3,1} + \frac{4-u}{4-2}N_{4,1} = \begin{cases} {}^{1}\!/_{2}(u-1)^2 & 1 \le u < 2 \\ -{}^{11}\!/_{2} + 5u - u^2 & 2 \le u < 3 \\ {}^{1}\!/_{2}(4-u)^2 & 3 \le u < 4 \end{cases}$$

$$N_{4,2} = \frac{u-2}{4-2}N_{4,1} + \frac{4-u}{4-3}N_{5,1} = \begin{cases} {}^{1}\!/_{2}(u-2)^2 & 2 \le u < 3 \\ -16 + 10u - {}^{3}\!/_{2}u^2 & 3 \le u < 4 \end{cases}$$

$$N_{5,2} = \frac{u-3}{4-3}N_{5,1} + \frac{5-u}{5-4}N_{6,1} = \begin{cases} (u-3)^2 & 3 \le u < 4 \\ (5-u)^2 & 4 \le u < 5 \end{cases}$$

$$N_{6,2} = \frac{u-4}{5-4}N_{6,1} + \frac{5-u}{5-4}N_{7,1} = 2(u-4)(5-u) \qquad 4 \le u < 5$$

$$N_{7,2} = \frac{u-4}{5-4}N_{7,1} + \frac{5-u}{5-5}N_{8,1} = (u-4)^2 \qquad 4 \le u < 5$$

We now list a number of important properties of the B-spline basis functions. As we see in the next chapter, it is these properties which determine the many desirable geometric characteristics in B-spline curves and surfaces. Assume degree p and a knot vector $U = \{u_0, \ldots, u_m\}$.

P2.1 $N_{i,p}(u) = 0$ if u is outside the interval $[u_i, u_{i+p+1})$ (local support property). This is illustrated by the triangular scheme shown here. Notice

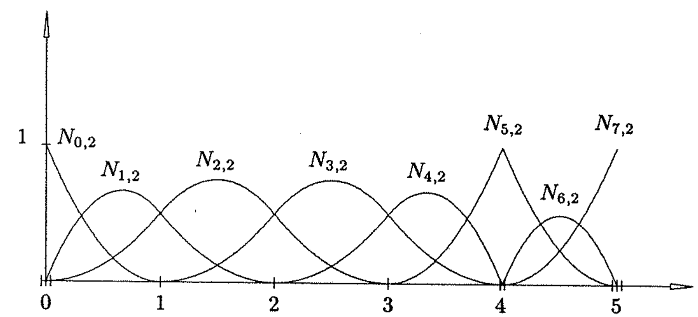


Figure 2.6. The nonzero second-degree basis functions, $U = \{0,0,0,1,2,3,4,4,5,5,5\}$.

that $N_{1,3}$ is a combination of $N_{1,0}$, $N_{2,0}$, $N_{3,0}$, and $N_{4,0}$. Thus, $N_{1,3}$ is nonzero only for $u \in [u_1, u_5)$

$$\begin{tikzcd}
N_{1,0} & & N_{0,2} & \\
& N_{1,1} \arrow[ul] \arrow[dl] & & N_{0,3} \\
N_{2,0} & & N_{1,2} \arrow[ul] \arrow[dl] & \\
& N_{2,1} \arrow[ul] \arrow[dl] & & N_{1,3} \arrow[ul] \arrow[dl] \\
N_{3,0} & & N_{2,2} \arrow[ul] \arrow[dl] & \\
& N_{3,1} \arrow[ul] \arrow[dl] & & N_{2,3} \\
N_{4,0} & & N_{3,2} &
\end{tikzcd}$$

P2.2 In any given knot span, $[u_j, u_{j+1})$, at most $p+1$ of the $N_{i,p}$ are nonzero, namely the functions $N_{j-p,p}, \ldots, N_{j,p}$. On $[u_3, u_4)$ the only nonzero zeroth-degree function is $N_{3,0}$. Hence, the only cubic functions not zero on $[u_3, u_4)$ are $N_{0,3}, \ldots, N_{3,3}$. This property is illustrated here

$$\begin{tikzcd}
& N_{1,1} & & N_{0,3} \\
N_{2,0} & & N_{1,2} \arrow[ur] \arrow[dr] & \\
& N_{2,1} \arrow[ur] \arrow[dr] & & N_{1,3} \\
N_{3,0} \arrow[ur] \arrow[dr] & & N_{2,2} \arrow[ur] \arrow[dr] & \\
& N_{3,1} \arrow[ur] \arrow[dr] & & N_{2,3} \\
N_{4,0} & & N_{3,2} \arrow[ur] \arrow[dr] & \\
& N_{4,1} & & N_{3,3}
\end{tikzcd}$$

P2.3 $N_{i,p}(u) \geq 0$ for all i, p, and u (nonnegativity). This is proven by induction on p. It is clearly true for $p = 0$; assume it is true for $p-1$, $p \geq 0$,

with i and u arbitrary. By definition

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u) \tag{2.6}$$

By P2.1, $N_{i,p-1}(u) = 0$ if $u \notin [u_i, u_{i+p})$. But $u \in [u_i, u_{i+p})$ implies

$$\frac{u - u_i}{u_{i+p} - u_i}$$

is nonnegative. By assumption, $N_{i,p-1}(u)$ is nonnegative, and thus the first term of Eq. (2.6) is nonnegative. The same is true for the second term, and hence the $N_{i,p}(u)$ are nonnegative;

P2.4 For an arbitrary knot span, $[u_i, u_{i+1})$, $\sum_{j=i-p}^{i} N_{j,p}(u) = 1$ for all $u \in [u_i, u_{i+1})$ (partition of unity). To prove this, consider

$$\begin{aligned}\sum_{j=i-p}^{i} N_{j,p}(u) &= \sum_{j=i-p}^{i} \frac{u - u_j}{u_{j+p} - u_j} N_{j,p-1}(u) \\ &\quad + \sum_{j=i-p}^{i} \frac{u_{j+p+1} - u}{u_{j+p+1} - u_{j+1}} N_{j+1,p-1}(u)\end{aligned}$$

Changing the summation variable in the second sum from $i-p$ to $i-p+1$, and considering that $N_{i-p,p-1}(u) = N_{i+1,p-1}(u) = 0$, we have

$$\begin{aligned}\sum_{j=i-p}^{i} N_{j,p}(u) &= \sum_{j=i-p+1}^{i} \left[\frac{u - u_j}{u_{j+p} - u_j} + \frac{u_{j+p} - u}{u_{j+p} - u_j} \right] N_{j,p-1}(u) \\ &= \sum_{j=i-p+1}^{i} N_{j,p-1}(u)\end{aligned}$$

Applying the same concept recursively yields

$$\begin{aligned}\sum_{j=i-p}^{i} N_{j,p}(u) &= \sum_{j=i-p+1}^{i} N_{j,p-1}(u) = \sum_{j=i-p+2}^{i} N_{j,p-2}(u) \\ &= \cdots = \sum_{j=i}^{i} N_{j,0}(u) = 1\end{aligned}$$

P2.5 All derivatives of $N_{i,p}(u)$ exist in the interior of a knot span (where it is a polynomial, see Figure 2.7). At a knot $N_{i,p}(u)$ is $p-k$ times continuously differentiable, where k is the multiplicity of the knot. Hence, increasing

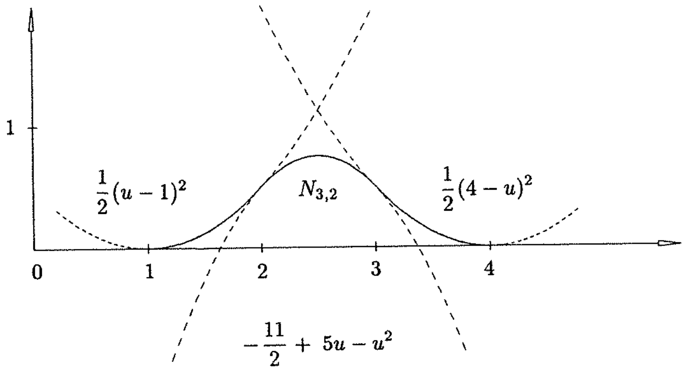


Figure 2.7. The decomposition of $N_{3,2}$ into its polynomial pieces (parabolas).

degree increases continuity, and increasing knot multiplicity decreases continuity;

P2.6 Except for the case $p = 0$, $N_{i,p}(u)$ attains exactly one maximum value.

It is important to understand the effect of multiple knots. Consider the functions $N_{0,2}$, $N_{1,2}$, $N_{2,2}$, $N_{5,2}$, and $N_{6,2}$ of Figure 2.6. Recalling that $U = \{0,0,0,1,2,3,4,4,5,5,5\}$, from Eq. (2.5) and P2.1, we see that these functions are computed on the following knot spans and are zero outside these spans

$$\begin{aligned} N_{0,2} &: \{0,0,0,1\} \\ N_{1,2} &: \{0,0,1,2\} \\ N_{2,2} &: \{0,1,2,3\} \\ N_{5,2} &: \{3,4,4,5\} \\ N_{6,2} &: \{4,4,5,5\} \end{aligned}$$

Now the word 'multiplicity' is understood in two different ways:

- the multiplicity of a knot in the knot vector;
- the multiplicity of a knot with respect to a specific basis function.

For example, $u = 0$ has multiplicity three in the previous knot vector U. But with respect to the functions $N_{0,2}$, $N_{1,2}$, $N_{2,2}$, and $N_{5,2}$, $u = 0$ is a knot of multiplicity 3, 2, 1, and 0, respectively. From P2.5, the continuity of these functions at $u = 0$ is $N_{0,2}$ discontinuous; $N_{1,2}$ C^0 continuous; $N_{2,2}$ C^1 continuous; and $N_{5,2}$ totally unaffected ($N_{5,2}$ and all its derivatives are zero at $u = 0$, from both sides). $N_{1,2}$ 'sees' $u = 0$ as a double knot, hence it is C^0 continuous. $N_{2,2}$ 'sees' all its knots with multiplicity 1, thus it is C^1 continuous everywhere. Clearly, another effect of multiple knots (as seen by the functions) is to reduce the number of 'apparent' intervals on which a function is nonzero; e.g., $N_{6,2}$ is nonzero only on $u \in [4,5)$, and it is only C^0 continuous at $u = 4$ and $u = 5$.

2.3 Derivatives of B-spline Basis Functions

The derivative of a basis function is given by

$$N'_{i,p} = \frac{p}{u_{i+p} - u_i} N_{i,p-1}(u) - \frac{p}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u) \tag{2.7}$$

(See Figure 2.8 for a graphical illustration.) We prove this by induction on p. For $p = 1$, $N_{i,p-1}$ and $N_{i+1,p-1}$ are either 0 or 1, and thus $N'_{i,p}$ is either

$$\frac{1}{u_{i+1} - u_i} \quad \text{or} \quad -\frac{1}{u_{i+2} - u_{i+1}}$$

(see Figure 2.5). Now assume that Eq. (2.7) is true for $p - 1$, $p > 1$. Using the product rule, $(fg)' = f'g + fg'$, to differentiate the basis function

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u)$$

yields

$$\begin{aligned} N'_{i,p} = {} & \frac{1}{u_{i+p} - u_i} N_{i,p-1} + \frac{u - u_i}{u_{i+p} - u_i} N'_{i,p-1} \\ & - \frac{1}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1} + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N'_{i+1,p-1} \end{aligned} \tag{2.8}$$

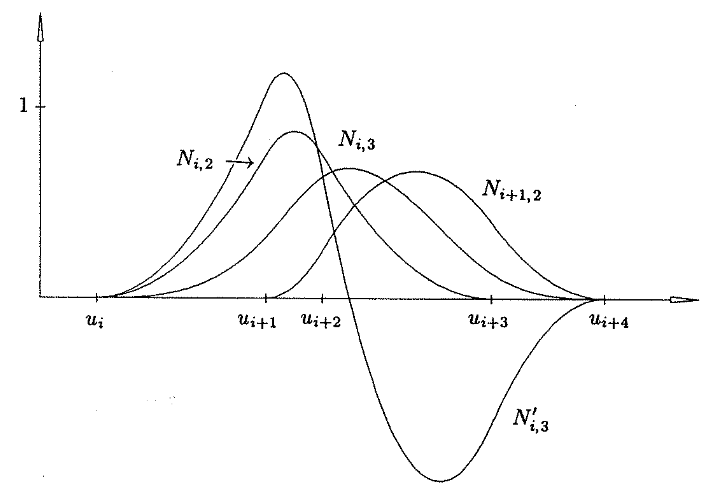


Figure 2.8. The recursive definition of B-spline derivatives.

Substituting Eq. (2.7) into Eq. (2.8) for $N'_{i,p-1}$ and $N'_{i+1,p-1}$ yields

$$\begin{aligned}
N'_{i,p} &= \frac{1}{u_{i+p}-u_i}N_{i,p-1} - \frac{1}{u_{i+p+1}-u_{i+1}}N_{i+1,p-1} \\
&\quad + \frac{u-u_i}{u_{i+p}-u_i}\left(\frac{p-1}{u_{i+p-1}-u_i}N_{i,p-2} - \frac{p-1}{u_{i+p}-u_{i+1}}N_{i+1,p-2}\right) \\
&\quad + \frac{u_{i+p+1}-u}{u_{i+p+1}-u_{i+1}}\left(\frac{p-1}{u_{i+p}-u_{i+1}}N_{i+1,p-2} - \frac{p-1}{u_{i+p+1}-u_{i+2}}N_{i+2,p-2}\right) \\
&= \frac{1}{u_{i+p}-u_i}N_{i,p-1} - \frac{1}{u_{i+p+1}-u_{i+1}}N_{i+1,p-1} \\
&\quad + \frac{p-1}{u_{i+p}-u_i}\,\frac{u-u_i}{u_{i+p-1}-u_i}N_{i,p-2} \\
&\quad + \frac{p-1}{u_{i+p}-u_{i+1}}\left(\frac{u_{i+p+1}-u}{u_{i+p+1}-u_{i+1}} - \frac{u-u_i}{u_{i+p}-u_i}\right)N_{i+1,p-2} \\
&\quad - \frac{p-1}{u_{i+p+1}-u_{i+1}}\,\frac{u_{i+p+1}-u}{u_{i+p+1}-u_{i+2}}N_{i+2,p-2}
\end{aligned}$$

Noting that

$$\begin{aligned}
\frac{u_{i+p+1}-u}{u_{i+p+1}-u_{i+1}} - \frac{u-u_i}{u_{i+p}-u_i} &= -1 + \frac{u_{i+p+1}-u}{u_{i+p+1}-u_{i+1}} + 1 - \frac{u-u_i}{u_{i+p}-u_i} \\
&= -\frac{u_{i+p+1}-u_{i+1}}{u_{i+p+1}-u_{i+1}} + \frac{u_{i+p+1}-u}{u_{i+p+1}-u_{i+1}} \\
&\quad + \frac{u_{i+p}-u_i}{u_{i+p}-u_i} - \frac{u-u_i}{u_{i+p}-u_i} \\
&= \frac{u_{i+p}-u}{u_{i+p}-u_i} - \frac{u-u_{i+1}}{u_{i+p+1}-u_{i+1}}
\end{aligned}$$

we obtain

$$\begin{aligned}
N'_{i,p} &= \frac{1}{u_{i+p}-u_i}N_{i,p-1} - \frac{1}{u_{i+p+1}-u_{i+1}}N_{i+1,p-1} \\
&\quad + \frac{p-1}{u_{i+p}-u_i}\left(\frac{u-u_i}{u_{i+p-1}-u_i}N_{i,p-2} + \frac{u_{i+p}-u}{u_{i+p}-u_{i+1}}N_{i+1,p-2}\right) \\
&\quad - \frac{p-1}{u_{i+p+1}-u_{i+1}}\left(\frac{u-u_{i+1}}{u_{i+p}-u_{i+1}}N_{i+1,p-2} + \frac{u_{i+p+1}-u}{u_{i+p+1}-u_{i+2}}N_{i+2,p-2}\right)
\end{aligned}$$

By the Cox-deBoor formula (Eq. [2.5]), the expressions in the parentheses can be replaced by $N_{i,p-1}$ and $N_{i+1,p-1}$, respectively. It follows that

$$\begin{aligned}
N'_{i,p} &= \frac{1}{u_{i+p}-u_i}N_{i,p-1} - \frac{1}{u_{i+p+1}-u_{i+1}}N_{i+1,p-1} \\
&\quad + \frac{p-1}{u_{i+p}-u_i}N_{i,p-1} - \frac{p-1}{u_{i+p+1}-u_{i+1}}N_{i+1,p-1} \\
&= \frac{p}{u_{i+p}-u_i}N_{i,p-1} - \frac{p}{u_{i+p+1}-u_{i+1}}N_{i+1,p-1}
\end{aligned}$$

This completes the proof.

Now let $N^{(k)}_{i,p}$ denote the kth derivative of $N_{i,p}(u)$. Repeated differentiation of Eq. (2.7) produces the general formula

$$N^{(k)}_{i,p}(u) = p\left(\frac{N^{(k-1)}_{i,p-1}}{u_{i+p}-u_i} - \frac{N^{(k-1)}_{i+1,p-1}}{u_{i+p+1}-u_{i+1}}\right) \tag{2.9}$$

Equation (2.10) is another generalization of Eq. (2.7). It computes the kth derivative of $N_{i,p}(u)$ in terms of the functions $N_{i,p-k}, \ldots, N_{i+k,p-k}$

$$N^{(k)}_{i,p} = \frac{p!}{(p-k)!}\sum_{j=0}^{k} a_{k,j}N_{i+j,p-k} \tag{2.10}$$

with

$$\begin{aligned}
a_{0,0} &= 1 \\
a_{k,0} &= \frac{a_{k-1,0}}{u_{i+p-k+1}-u_i} \\
a_{k,j} &= \frac{a_{k-1,j}-a_{k-1,j-1}}{u_{i+p+j-k+1}-u_{i+j}} \qquad j = 1,\ldots,k-1 \\
a_{k,k} &= \frac{-a_{k-1,k-1}}{u_{i+p+1}-u_{i+k}}
\end{aligned}$$

Remarks on Eq. (2.10):

- k should not exceed p (all higher derivatives are zero);
- the denominators involving knot differences can become zero; the quotient is defined to be zero in this case (see Example Ex2.4 and Algorithm A2.3 in Section 2.5).

We omit a proof of Eq. (2.10) but verify that it holds for $k = 1, 2$. By definition

$$a_{1,0} = \frac{1}{u_{i+p}-u_i} \qquad a_{1,1} = -\frac{1}{u_{i+p+1}-u_{i+1}}$$

and

$$N^{(1)}_{i,p} = 2(a_{1,0}\,N_{i,p-1} + a_{1,1}\,N_{i+1,p-1})$$

Comparing this with Eq. (2.7) proves the case for $k=1$; now let $k=2$. Differentiating Eq. (2.7) yields

$$
\begin{aligned}
N_{i,p}^{(2)} &= \frac{p}{u_{i+p}-u_i} N_{i,p-1}^{(1)} - \frac{p}{u_{i+p+1}-u_{i+1}} N_{i+1,p-1}^{(1)} \\
&= \frac{p}{u_{i+p}-u_i}\left(\frac{p-1}{u_{i+p-1}-u_i} N_{i,p-2} - \frac{p-1}{u_{i+p}-u_{i+1}} N_{i+1,p-2}\right) \\
&\quad - \frac{p}{u_{i+p+1}-u_{i+1}}\left(\frac{p-1}{u_{i+p}-u_{i+1}} N_{i+1,p-2} - \frac{p-1}{u_{i+p+1}-u_{i+2}} N_{i+2,p-2}\right) \\
&= p(p-1)\Bigg[\frac{a_{1,0}}{u_{i+p-1}-u_i} N_{i,p-2} \\
&\quad - \frac{1}{u_{i+p}-u_{i+1}}\left(\frac{1}{u_{i+p}-u_i} + \frac{1}{u_{i+p+1}-u_{i+1}}\right) N_{i+1,p-2} \\
&\quad + \frac{a_{1,1}}{u_{i+p+1}-u_{i+2}} N_{i+2,p-2}\Bigg] \\
&= p(p-1)\left(a_{2,0}\, N_{i,p-2} + \frac{a_{1,1}-a_{1,0}}{u_{i+p}-u_{i+1}} N_{i+1,p-2} + a_{2,2}\, N_{i+2,p-2}\right)
\end{aligned}
$$

Noting that $k=2$ and

$$a_{2,1} = \frac{a_{1,1}-a_{1,0}}{u_{i+p}-u_{i+1}}$$

it follows that

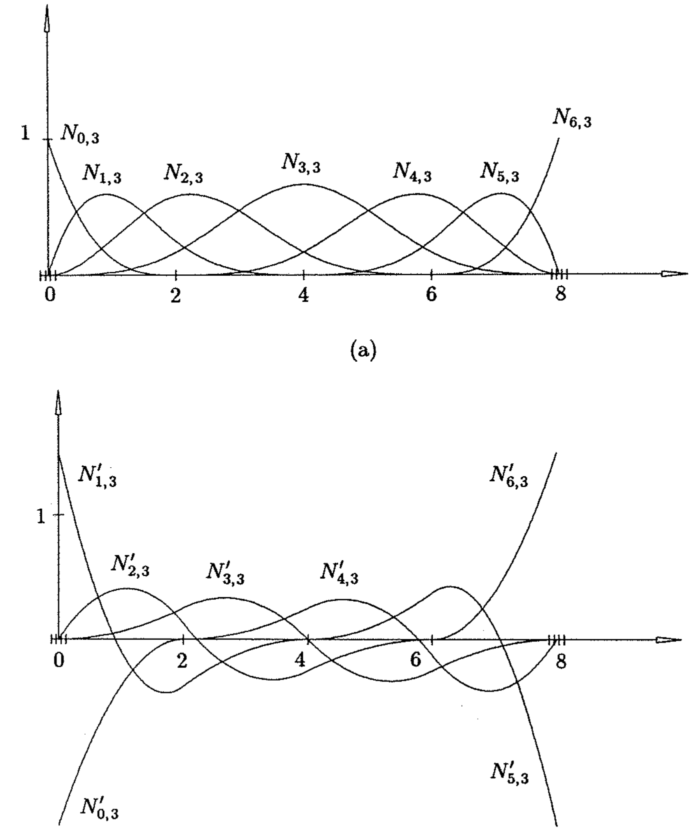
$$N_{i,p}^{(2)}(u) = 2\sum_{j=0}^{2} a_{2,j} N_{i+j,p-2}(u)$$

For completeness, we give an additional formula for computing derivatives of the B-spline basis functions (see [Butt76])

$$N_{i,p}^{(k)} = \frac{p}{p-k}\left(\frac{u-u_i}{u_{i+p}-u_i} N_{i,p-1}^{(k)} + \frac{u_{i+p+1}-u}{u_{i+p+1}-u_{i+1}} N_{i+1,p-1}^{(k)}\right) \qquad k=0,\ldots,p-1 \tag{2.11}$$

Equation (2.11) gives the kth derivative of $N_{i,p}(u)$ in terms of the kth derivative of $N_{i,p-1}$ and $N_{i+1,p-1}$.

Figures 2.9b and 2.10b show the derivatives corresponding to the basis functions in Figures 2.9a and 2.10a. Figure 2.11 shows all the nonzero derivatives of $N_{i,3}$. Note the effect of multiple knots in Figure 2.10b; $N'_{6,3}$ has a jump at the triple knot.



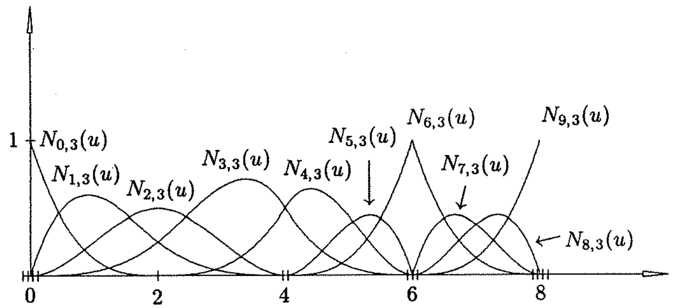
(b)

Figure 2.9. (a) Cubic basis functions; (b) derivatives corresponding to the basis functions in Figure 2.9a.

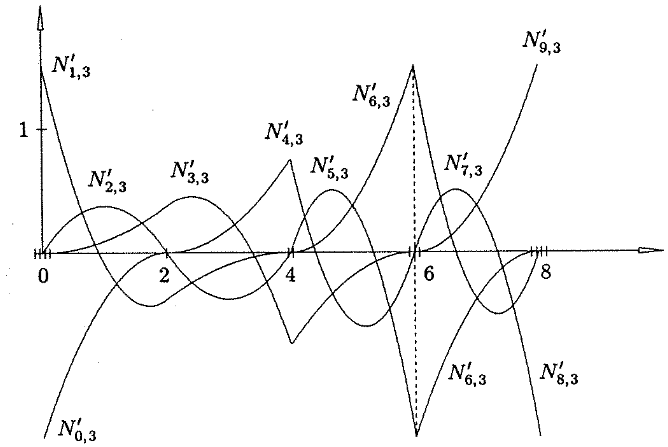
2.4 Further Properties of the Basis Functions

Let $\{u_j\}$, $0 \le j \le k$, be a strictly increasing set of breakpoints. The set of all piecewise polynomial functions of degree p on $\{u_j\}$ which are C^{r_j} continuous at $u=u_j$ forms a vector space, $\mathcal{V}$ $(-1 \le r_j \le p)$. If no continuity constraints are imposed ($r_j = -1$ for all j), then the dimension of $\mathcal{V}$ (denoted $\dim(\mathcal{V})$) is equal to $k(p+1)$. Each continuity constraint decreases the dimension by one, thus

$$\dim(\mathcal{V}) = k(p+1) - \sum_{j=0}^{k}(r_j+1) \tag{2.12}$$



(a)



(b)

Figure 2.10. (a) Cubic basis functions showing single, double, and triple knots; (b) derivatives of the functions in Figure 2.10a.

By Property P2.5, we obtain the B-spline basis functions of p-degree with knots at the $\{u_j\}$, and with the desired continuity, by setting the appropriate knot multiplicities, s_j, where $s_j = p - r_j$. Hence, we use a knot vector of the form

$$U = \{\underbrace{u_0, \ldots, u_0}_{s_0}, \underbrace{u_1, \ldots, u_1}_{s_1}, \ldots, \underbrace{u_k, \ldots, u_k}_{s_k}\}$$

Now set

$$m = \left(\sum_{j=0}^{k} s_j\right) - 1$$

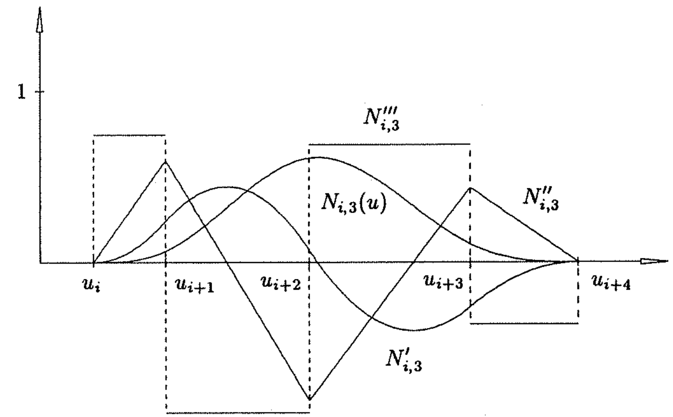


Figure 2.11. $N_{i,3}$ and all its nonzero derivatives.

Then clearly, there are m zeroth-degree functions, $N_{i,0}$, $m-1$ first degree functions, $N_{i,1}$, and in general, $m-p$ pth-degree functions, $N_{i,p}$, which have the desired continuity, $r_j = p - s_j$. Hence the $N_{i,p}$ are contained in $\mathcal{V}$. Substituting $s_j = p - r_j$ into Eq. (2.12) yields

$$\begin{aligned}
\dim(\mathcal{V}) &= k(p+1) - \sum_{j=0}^{k} (p - s_j + 1) \\
&= k(p+1) - (k+1)p + \sum_{j=0}^{k} s_j - (k+1) \\
&= -p - 1 + \sum_{j=0}^{k} s_j \\
&= m - p
\end{aligned}$$

Thus, the number of pth-degree B-spline basis functions on U is equal to $\dim(\mathcal{V})$. We now justify the term 'basis' functions by showing that the $N_{i,p}$ are linearly independent, i.e., they form a basis for the vector space, $\mathcal{V}$. The proof is by induction on p. Clearly, the zeroth-degree functions are linearly independent. Assume the $(p-1)$th-degree functions are linearly independent for $p > 0$. Set $n = m - p - 1$, and assume that

$$\sum_{i=0}^{n} \alpha_i N_{i,p}(u) = 0 \quad \text{for all } u$$

Using Eq. (2.7) we obtain

$$0 = \left(\sum_{i=0}^{n} \alpha_i N_{i,p}\right)' = \sum_{i=0}^{n} \alpha_i N'_{i,p}$$

$$= p\sum_{i=0}^{n} \alpha_i \left(\frac{N_{i,p-1}}{u_{i+p}-u_i} - \frac{N_{i+1,p-1}}{u_{i+p+1}-u_{i+1}}\right)$$

which implies that

$$0 = \sum_{i=0}^{n} \alpha_i \frac{N_{i,p-1}}{u_{i+p}-u_i} - \sum_{i=0}^{n} \alpha_i \frac{N_{i+1,p-1}}{u_{i+p+1}-u_{i+1}}$$

Now noting that $N_{0,p-1} = N_{n+1,p-1} = 0$, and changing the summation variable in the second term, we have

$$0 = \sum_{i=1}^{n} \frac{\alpha_i - \alpha_{i-1}}{u_{i+p}-u_i} N_{i,p-1}$$

which implies $\alpha_i - \alpha_{i-1} = 0$ for all i (by assumption), which in turn implies $\alpha_i = 0$ for all i. This completes the proof.

We turn our attention now to knot vectors. Clearly, once the degree is fixed the knot vector completely determines the functions $N_{i,p}(u)$. There are several types of knot vectors, and unfortunately terminology varies in the literature. In this book we consider only *nonperiodic* (or *clamped* or *open*) knot vectors, which have the form

$$U = \{\underbrace{a,\ldots,a}_{p+1}, u_{p+1},\ldots,u_{m-p-1}, \underbrace{b,\ldots,b}_{p+1}\} \tag{2.13}$$

that is, the first and last knots have multiplicity $p+1$. For nonperiodic knot vectors we have two additional properties of the basis functions:

P2.7 A knot vector of the form

$$U = \{\underbrace{0,\ldots,0}_{p+1}, \underbrace{1,\ldots,1}_{p+1}\}$$

yields the Bernstein polynomials of degree p (see Example Ex2.1 in Section 2.2);

P2.8 Let $m+1$ be the number of knots. Then there are $n+1$ basis functions, where $n = m-p-1$; $N_{0,p}(a) = 1$ and $N_{n,p}(b) = 1$. For example, $N_{0,p}(a) = 1$ follows from the fact that $N_{0,0},\ldots,N_{p-1,0} = 0$, since this implies that $N_{0,p}(a) = N_{p,0}(a) = 1$. From P2.4 it follows that $N_{i,p}(a) = 0$ for $i \neq 0$, and $N_{i,p}(b) = 0$ for $i \neq n$.

For the remainder of this book, all knot vectors are understood to be nonperiodic. We define a knot vector $U = \{u_0,\ldots,u_m\}$ to be *uniform* if all interior knots are equally spaced, i.e., if there exists a real number, d, such that $d = u_{i+1} - u_i$ for all $p \le i \le m-p-1$; otherwise it is *nonuniform*. The knot vector of Example

Ex2.2, Section 2.2 is nonuniform because of the double knot at $u = 4$. Figure 2.9a shows a set of uniform cubic basis functions, and Figures 2.10a and 2.12 show nonuniform cubic basis functions.

2.5 Computational Algorithms

In this section we develop algorithms to compute values of the basis functions and their derivatives. Let $U = \{u_0,\ldots,u_m\}$ be a knot vector of the form in Eq. (2.13), and assume we are interested in the basis functions of degree p. Furthermore, assume u is fixed, and $u \in [u_i, u_{i+1})$. We develop five algorithms that compute:

- the knot span index, i;
- $N_{i-p,p}(u),\ldots,N_{i,p}(u)$ (based on Eq. [2.5]);
- $N^{(k)}_{i-p,p}(u),\ldots,N^{(k)}_{i,p}(u)$ for $k = 0,\ldots,p$; for $k > p$ the derivatives are zero (this algorithm is based on Eq. [2.10]);
- a single basis function, $N_{j,p}(u)$, where $0 \le j \le m-p-1$;
- the derivatives of a single basis function, $N^{(k)}_{j,p}(u)$, where $0 \le j \le m-p-1$ and $k = 0,\ldots,p$ (based on Eq. [2.9]).

We present the two algorithms which compute $p+1$ functions before the two which compute only one, because they are the most important and actually are somewhat simpler.

From P2.2 and the assumption that $u \in [u_i, u_{i+1})$, it follows that we can focus our attention on the functions $N_{i-p,p},\ldots,N_{i,p}$ and their derivatives; all other functions are identically zero, and it is wasteful to actually compute them. Hence, the first step in evaluation is to determine the knot span in which u lies. Either a linear or a binary search of the knot vector can be used; we present here a binary

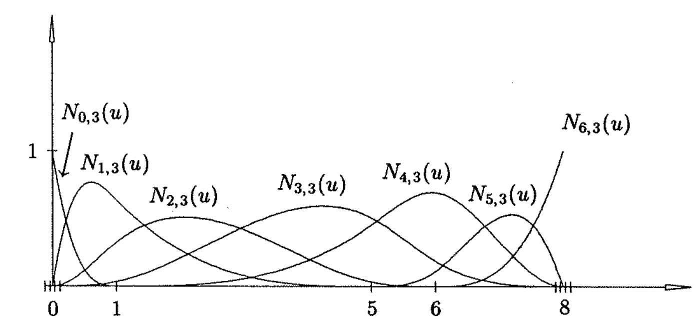


Figure 2.12. Nonuniform cubic basis functions defined on $U = \{0,0,0,0,1,5,6,8,8,8,8\}$.

search. Since we are using intervals of the form $u \in [u_i, u_{i+1})$, a subtle problem in the evaluation of the basis functions is the special case $u = u_m$. It is best to handle this at the lowest level by setting the span index to n $(= m - p - 1)$. Hence, in this case $u \in (u_{m-p-1}, u_{m-p}]$. `FindSpan` is an integer function which returns the span index.

```
ALGORITHM A2.1
  int FindSpan(n,p,u,U)
    { /* Determine the knot span index */
      /* Input: n,p,u,U */
      /* Return: the knot span index */
    if (u == U[n+1]) return(n);   /* Special case */
    low = p;   high = n+1;   /* Do binary search */
    mid =(low+high)/2;
    while (u < U[mid] || u >= U[mid+1])
      {
      if (u < U[mid])   high = mid;
        else            low = mid;
      mid = (low+high)/2;
      {
    return(mid);
    }
```

Now we tackle the second algorithm. Assuming u is in the ith span, computation of the nonzero functions results in an inverted triangular scheme

$$
\begin{array}{cccc}
 & & & N_{i-p,p} \\
 & N_{i-1,1} & & \\
N_{i,0} & & \cdots & \vdots \\
 & N_{i,1} & & \\
 & & & N_{i,p}
\end{array}
$$

Example

Ex2.3 Let $p = 2$, $U = \{0,0,0,1,2,3,4,4,5,5,5\}$, and $u = 5/2$ (see Figure 2.6). Then $i = 4$, since $u \in [u_4, u_5)$. Thus, we compute

$$
\begin{array}{ccc}
 & & N_{2,2}(5/2) \\
 & N_{3,1}(5/2) & \\
N_{4,0}(5/2) & & N_{3,2}(5/2) \\
 & N_{4,1}(5/2) & \\
 & & N_{4,2}(5/2)
\end{array}
$$

Substituting $u = 5/2$ into Eq. (2.5) (the reader should do this) yields

$$N_{4,0}\left(\frac{5}{2}\right) = 1$$

$$N_{3,1}\left(\frac{5}{2}\right) = \frac{1}{2} \qquad N_{4,1}\left(\frac{5}{2}\right) = \frac{1}{2}$$

$$N_{2,2}\left(\frac{5}{2}\right) = \frac{1}{8} \qquad N_{3,2}\left(\frac{5}{2}\right) = \frac{6}{8} \quad N_{4,2}\left(\frac{5}{2}\right) = \frac{1}{8}$$

Notice that for fixed degree the functions sum to 1 (P2.4).

It will be clear to the reader who carried out the substitutions in this example that there is a great deal of redundant computation inherent in Eq. (2.5). For example, writing out the second-degree functions in general terms, we have

$$N_{i-2,2}(u) = \frac{u - u_{i-2}}{u_i - u_{i-2}} N_{i-2,1}(u) + \frac{u_{i+1} - u}{u_{i+1} - u_{i-1}} N_{i-1,1}(u) \tag{2.14}$$

$$N_{i-1,2}(u) = \frac{u - u_{i-1}}{u_{i+1} - u_{i-1}} N_{i-1,1}(u) + \frac{u_{i+2} - u}{u_{i+2} - u_i} N_{i,1}(u) \tag{2.15}$$

$$N_{i,2}(u) = \frac{u - u_i}{u_{i+2} - u_i} N_{i,1}(u) + \frac{u_{i+3} - u}{u_{i+3} - u_{i+1}} N_{i+1,1}(u) \tag{2.16}$$

Note that

- the first term of Eq. (2.14) and the last term of Eq. (2.16) are not computed, since $N_{i-2,1}(u) = N_{i+1,1}(u) = 0$;
- the expression
$$\frac{N_{i-1,1}(u)}{u_{i+1} - u_{i-1}}$$
which appears in the second term of Eq. (2.14) appears in the first term of Eq. (2.15); a similar statement holds for the second term of Eq. (2.15) and the first term of Eq. (2.16).

We introduce the notation

$$\text{left}\,[j] = u - u_{i+1-j} \qquad \text{right}\,[j] = u_{i+j} - u$$

Equations (2.14)–(2.16) are then

$$N_{i-2,2}(u) = \frac{\text{left}\,[3]}{\text{right}\,[0] + \text{left}\,[3]} N_{i-2,1}(u) + \frac{\text{right}\,[1]}{\text{right}\,[1] + \text{left}\,[2]} N_{i-1,1}(u)$$

$$N_{i-1,2}(u) = \frac{\text{left}\,[2]}{\text{right}\,[1] + \text{left}\,[2]} N_{i-1,1}(u) + \frac{\text{right}\,[2]}{\text{right}\,[2] + \text{left}\,[1]} N_{i,1}(u)$$

$$N_{i,2}(u) = \frac{\text{left}\,[1]}{\text{right}\,[2] + \text{left}\,[1]} N_{i,1}(u) + \frac{\text{right}\,[3]}{\text{right}\,[3] + \text{left}\,[0]} N_{i+1,1}(u)$$

Based on these observations, Algorithm A2.2 computes all the nonvanishing basis functions and stores them in the array N[0],...,N[p].

```
ALGORITHM A2.2
  BasisFuns(i,u,p,U,N)
    { /* Compute the nonvanishing basis functions */
      /* Input:  i,u,p,U  */
      /* Output: N  */
    N[0]=1.0;
    for (j=1; j<=p; j++)
      {
      left[j] = u-U[i+1-j];
      right[j] = U[i+j]-u;
      saved = 0.0;
      for (r=0; r<j; r++)
        {
        temp = N[r]/(right[r+1]+left[j-r]);
        N[r] = saved+right[r+1]*temp;
        saved = left[j-r]*temp;
        }
      N[j] = saved;
      }
    }
```

We remark that Algorithm A2.2 is not only efficient, but it also guarantees that there will be no division by zero, which can occur with a direct application of Eq. (2.5).

Now to the third algorithm; in particular, we want to compute all $N_{r,p}^{(k)}(u)$, for $i-p \le r \le i$ and $0 \le k \le n$, where $n \le p$. Inspection of Eq. (2.10) reveals that the basic ingredients are:

- the inverted triangle of nonzero basis functions computed in Algorithm A2.2;
- differences of knots (the sums: right[r+1]+left[j-r]), also computed in Algorithm A2.2;
- differences of the $a_{k,j}$; note that the $a_{k,j}$ depend on the $a_{k-1,j}$ but not the $a_{s,j}$, for $s < k-1$.

Viewed as a two-dimensional array of dimension $(p+1) \times (p+1)$, the basis functions fit into the upper triangle (including the diagonal), and the knot differences fit into the lower triangle, that is

$N_{i,0}(u)$	$N_{i-1,1}(u)$	$N_{i-2,2}(u)$
$u_{i+1} - u_i$	$N_{i,1}(u)$	$N_{i-1,2}(u)$
$u_{i+1} - u_{i-1}$	$u_{i+2} - u_i$	$N_{i,2}(u)$

Example

Ex2.4 Let $p = 2$, $U = \{0,0,0,1,2,3,4,4,5,5,5\}$, and $u = 5/2$. Then $u \in [u_4, u_5)$, and the array becomes

$N_{4,0}\left(\frac{5}{2}\right) = 1$	$N_{3,1}\left(\frac{5}{2}\right) = \frac{1}{2}$	$N_{2,2}\left(\frac{5}{2}\right) = \frac{1}{8}$
$u_5 - u_4 = 1$	$N_{4,1}\left(\frac{5}{2}\right) = \frac{1}{2}$	$N_{3,2}\left(\frac{5}{2}\right) = \frac{6}{8}$
$u_5 - u_3 = 2$	$u_6 - u_4 = 2$	$N_{4,2}\left(\frac{5}{2}\right) = \frac{1}{8}$

Now compute $N_{4,2}^{(1)}(5/2)$ and $N_{4,2}^{(2)}(5/2)$; with $i = 4$ in Eq. (2.10), we have

$$a_{1,0} = \frac{1}{u_6 - u_4} = \frac{1}{2}$$

$$a_{1,1} = -\frac{1}{u_7 - u_5} = -1$$

$$a_{2,0} = \frac{a_{1,0}}{u_5 - u_4} = \frac{1}{2}$$

$$a_{2,1} = \frac{a_{1,1} - a_{1,0}}{u_6 - u_5} = \frac{-1 - \frac{1}{2}}{4-3} = -\frac{3}{2}$$

$$a_{2,2} = -\frac{a_{1,1}}{u_7 - u_6} = \frac{1}{4-4} = \frac{1}{0}$$

$$N_{4,2}^{(1)} = 2\left[a_{1,0}N_{4,1}\left(\frac{5}{2}\right) + a_{1,1}N_{5,1}\left(\frac{5}{2}\right)\right]$$

and

$$N_{4,2}^{(2)} = 2\left[a_{2,0}N_{4,0}\left(\frac{5}{2}\right) + a_{2,1}N_{5,0}\left(\frac{5}{2}\right) + a_{2,2}N_{6,0}\left(\frac{5}{2}\right)\right]$$

Now $a_{1,1}$, $a_{2,1}$, and $a_{2,2}$ all use knot differences which are not in the array, but they are multiplied respectively by $N_{5,1}(5/2)$, $N_{5,0}(5/2)$, and

$N_{6,0}(5/2)$, which are also not in the array. These terms are defined to be zero, and we are left with

$$N_{4,2}^{(1)} = 2a_{1,0}N_{4,1}\left(\frac{5}{2}\right) = \frac{1}{2}$$

$$N_{4,2}^{(2)} = 2a_{2,0}N_{4,0}\left(\frac{5}{2}\right) = 1$$

To check these values, recall from Section 2.2 that $N_{4,2}(u) = 1/2(u-2)^2$ on $u \in [2,3)$. The computation of $N_{3,2}^{(1)}(5/2)$, $N_{3,2}^{(2)}(5/2)$, $N_{2,2}^{(1)}(5/2)$, and $N_{2,2}^{(2)}(5/2)$ is analogous.

Based on these observations (and **Ex2.4**), it is not difficult to develop Algorithm **A2.3**, which computes the nonzero basis functions and their derivatives, up to and including the nth derivative ($n \leq p$). Output is in the two-dimensional array, **ders**. **ders[k][j]** is the kth derivative of the function $N_{i-p+j,p}$, where $0 \leq k \leq n$ and $0 \leq j \leq p$. Two local arrays are used:

- **ndu[p+1][p+1]**, to store the basis functions and knot differences;
- **a[2][p+1]**, to store (in an alternating fashion) the two most recently computed rows $a_{k,j}$ and $a_{k-1,j}$.

The algorithm avoids division by zero and/or the use of terms not in the array **ndu[][]**.

```
ALGORITHM A2.3
  DersBasisFuns(i,u,p,n,U,ders)
    { /*  Compute nonzero basis functions and their  */
      /*  derivatives. First section is A2.2 modified  */
      /*  to store functions and knot differences.  */
      /*  Input:  i,u,p,n,U  */
      /*  Output: ders  */
    ndu[0][0]=1.0;
    for (j=1; j<=p; j++)
      {
      left[j] = u-U[i+1-j];
      right[j] = U[i+j]-u;
      saved = 0.0;
      for (r=0; r<j; r++)
        {                                  /* Lower triangle */
        ndu[j][r] = right[r+1]+left[j-r];
        temp = ndu[r][j-1]/ndu[j][r];
                                           /* Upper triangle */
        ndu[r][j] = saved+right[r+1]*temp;
        saved = left[j-r]*temp;
        }
      ndu[j][j] = saved;
```



```
      }
    for (j=0; j<=p; j++)    /* Load the basis functions */
      ders[0][j] = ndu[j][p];
    /* This section computes the derivatives (Eq. [2.9]) */
    for (r=0; r<=p; r++)    /* Loop over function index */
      {
      s1=0;   s2=1;   /* Alternate rows in array a */
      a[0][0] = 1.0;
      /* Loop to compute kth derivative */
      for (k=1; k<=n; k++)
        {
        d = 0.0;
        rk = r-k;    pk = p-k;
        if (r >= k)
          {
          a[s2][0] = a[s1][0]/ndu[pk+1][rk];
          d = a[s2][0]*ndu[rk][pk];
          }
        if (rk >= -1)   j1 = 1;
          else          j1 = -rk;
        if (r-1 <= pk)   j2 = k-1;
          else           j2 = p-r;
        for (j=j1; j<=j2; j++)
          {
          a[s2][j] = (a[s1][j]-a[s1][j-1])/ndu[pk+1][rk+j];
          d += a[s2][j]*ndu[rk+j][pk];
          }
        if (r <= pk)
          {
          a[s2][k] = -a[s1][k-1]/ndu[pk+1][r];
          d += a[s2][k]*ndu[r][pk];
          }
        ders[k][r] = d;
        j=s1;   s1=s2;   s2=j;   /* Switch rows */
        }
      }
    /* Multiply through by the correct factors */
    /* (Eq. [2.9]) */
    r = p;
    for (k=1; k<=n; k++)
      {
      for (j=0; j<=p; j++)   ders[k][j] *= r;
      r *= (p-k);
      }
    }
```

We turn our attention now to the last two algorithms, namely computing a single basis function, $N_{i,p}(u)$, or the derivatives, $N_{i,p}^{(k)}(u)$, of a single basis function. The solutions to these problems result in triangular tables of the form

$$\begin{array}{cccc} N_{i,0} & & & \\ & N_{i,1} & & \\ N_{i+1,0} & & N_{i,2} & \\ \vdots & & \cdots & N_{i,p} \\ N_{i+p-1,0} & & N_{i+p-2,2} & \\ & N_{i+p-1,1} & & \\ N_{i+p,0} & & & \end{array}$$

Example

Ex2.5 Let $p = 2$, $U = \{0,0,0,1,2,3,4,4,5,5,5\}$, and $u = 5/2$. The computation of $N_{3,2}(5/2)$ yields

$$\begin{array}{ccc} N_{3,0}(5/2) = 0 & & \\ & N_{3,1}(5/2) = \dfrac{1}{2} & \\ N_{4,0}(5/2) = 1 & & N_{3,2}(5/2) = \dfrac{6}{8} \\ & N_{4,1}(5/2) = \dfrac{1}{2} & \\ N_{5,0}(5/2) = 0 & & \end{array}$$

$N_{4,2}(5/2)$ is obtained from

$$\begin{array}{ccc} N_{4,0}(5/2) = 1 & & \\ & N_{4,1}(5/2) = \dfrac{1}{2} & \\ N_{5,0}(5/2) = 0 & & N_{4,2}(5/2) = \dfrac{1}{8} \\ & N_{5,1}(5/2) = 0 & \\ N_{6,0}(5/2) = 0 & & \end{array}$$

Notice that the position and relative number of nonzero entries in the table depend on p and on the position of the 1 in the first column. Algorithm A2.4 computes only the nonzero entries. The value $N_{i,p}(u)$ is returned in Nip; m is the high index of U ($m + 1$ knots). The algorithm is similar to Algorithm A2.2 in its use of the variables temp and saved.

```
ALGORITHM A2.4
  OneBasisFun(p,m,U,i,u,Nip)
    { /* Compute the basis function Nip */
      /* Input:  p,m,U,i,u  */
      /* Output: Nip  */
```



```
  if ((i == 0 && u == U[0]) ||        /* Special */
      (i == m-p-1 && u == U[m]))      /*  cases  */
    {
    Nip = 1.0;    return;
    }
  if (u < U[i] || u >= U[i+p+1])      /* Local property */
    {
    Nip = 0.0;    return;
    }
  for (j=0; j<=p; j++) /* Initialize zeroth-degree functs */
    if (u >= U[i+j] && u < U[i+j+1])  N[j] = 1.0;
      else          N[j] = 0.0;
  for (k=1; k<=p; k++)  /* Compute triangular table */
    {
    if (N[0] == 0.0)  saved = 0.0;
      else        saved = ((u-U[i])*N[0])/(U[i+k]-U[i]);
    for (j=0; j<p-k+1; j++)
      {
      Uleft  = U[i+j+1];
      Uright = U[i+j+k+1];
      if (N[j+1] == 0.0)
        {
        N[j] = saved;    saved = 0.0;
        }
        else
        {
        temp = N[j+1]/(Uright-Uleft);
        N[j] = saved+(Uright-u)*temp;
        saved = (u-Uleft)*temp;
        }
      }
    }
  Nip = N[0];
  }
```

Now for fixed i, the computation of the derivatives, $N_{i,p}^{(k)}(u)$, for $k = 0,\ldots,n$, $n \le p$, uses Eq. (2.9). For example, if $p = 3$ and $n = 3$, then

$$N_{i,3}^{(1)} = 3\left(\frac{N_{i,2}}{u_{i+3}-u_i} - \frac{N_{i+1,2}}{u_{i+4}-u_{i+1}}\right)$$

$$N_{i,3}^{(2)} = 3\left(\frac{N_{i,2}^{(1)}}{u_{i+3}-u_i} - \frac{N_{i+1,2}^{(1)}}{u_{i+4}-u_{i+1}}\right)$$

$$N_{i,3}^{(3)} = 3\left(\frac{N_{i,2}^{(2)}}{u_{i+3}-u_i} - \frac{N_{i+1,2}^{(2)}}{u_{i+4}-u_{i+1}}\right)$$

Using triangular tables, we must compute

$k = 0$:

$$\begin{array}{cccc}
N_{i,0} & & & \\
 & N_{i,1} & & \\
N_{i+1,0} & & N_{i,2} & \\
 & N_{i+1,1} & & N_{i,3} \\
N_{i+2,0} & & N_{i+1,2} & \\
 & N_{i+2,1} & & \\
N_{i+3,0} & & &
\end{array}$$

$k = 1$:

$$\begin{array}{cc}
N_{i,2} & \\
 & N_{i,3}^{(1)} \\
N_{i+1,2} &
\end{array}$$

$k = 2$:

$$\begin{array}{ccc}
N_{i,1} & & \\
 & N_{i,2}^{(1)} & \\
N_{i+1,1} & & N_{i,3}^{(2)} \\
 & N_{i+1,2}^{(1)} & \\
N_{i+2,1} & &
\end{array}$$

$k = 3$:

$$\begin{array}{cccc}
N_{i,0} & & & \\
 & N_{i,1}^{(1)} & & \\
N_{i+1,0} & & N_{i,2}^{(2)} & \\
 & N_{i+1,1}^{(1)} & & N_{i,3}^{(3)} \\
N_{i+2,0} & & N_{i+1,2}^{(2)} & \\
 & N_{i+2,1}^{(1)} & & \\
N_{i+3,0} & & &
\end{array}$$

In words, the algorithm is:

1. compute and store the entire triangular table corresponding to $k = 0$;
2. to get the kth derivative, load the column of the table which contains the functions of degree $p - k$, and compute the remaining portion of the triangle.

Algorithm A2.5 computes $N_{i,p}^{(k)}(u)$ for $k = 0, \ldots, n$, $n \le p$. The kth derivative is returned in ders[k].

```
ALGORITHM A2.5
  DersOneBasisFun(p,m,U,i,u,n,ders)
    { /*  Compute derivatives of basis function Nip  */
```



```
      /*  Input:  p,m,U,i,u,n  */
      /*  Output: ders  */
    if (u < U[i] || u >= U[i+p+1])   /* Local property */
      {
      for (k=0; k<=n; k++)  ders[k] = 0.0;
      return;
      }
    for (j=0; j<=p; j++)  /* Initialize zeroth-degree functs */
      if (u >= U[i+j] && u < U[i+j+1])  N[j][0] = 1.0;
        else                            N[j][0] = 0.0;
    for (k=1; k<=p; k++)   /* Compute full triangular table */
      {
      if (N[0][k-1] == 0.0)  saved = 0.0;
        else    saved = ((u-U[i])*N[0][k-1])/(U[i+k]-U[i]);
      for (j=0; j<p-k+1; j++)
        {
        Uleft = U[i+j+1];
        Uright = U[i+j+k+1];
        if (N[j+1][k-1] == 0.0)
          {
          N[j][k] = saved;   saved = 0.0;
          }
          else
          {
          temp = N[j+1][k-1]/(Uright-Uleft);
          N[j][k] = saved+(Uright-u)*temp;
          saved = (u-Uleft)*temp;
          }
        }
      }
    ders[0] = N[0][p];   /* The function value */
    for (k=1; k<=n; k++)   /* Compute the derivatives */
      {
      for (j=0; j<=k; j++)   /* Load appropriate column */
        ND[j] = N[j][p-k];
      for (jj=1; jj<=k; jj++)   /* Compute table of width k */
        {
        if (ND[0] == 0.0)  saved = 0.0;
          else     saved = ND[0]/(U[i+p-k+jj]-U[i]);
        for (j=0; j<k-jj+1; j++)
          {
          Uleft = U[i+j+1];
          Uright = U[i+j+p+jj+1];
          if (ND[j+1] == 0.0)
            {
```

```
            ND[j] = (p-k+jj)*saved;    saved = 0.0;
            }
            else
            {
            temp = ND[j+1]/(Uright-Uleft);
            ND[j] = (p-k+jj)*(saved-temp);
            saved = temp;
            }
          }
        }
    ders[k] = ND[0];    /* kth derivative */
    }
  }
```

Finally, note that Algorithms A2.3 and A2.5 compute derivatives from the right if u is a knot. However, Eqs. (2.5), (2.9), (2.10), and others in this chapter could have been defined using intervals of the form $u \in (u_i, u_{i+1}]$. This would not change Algorithms A2.2 through A2.5. In other words, derivatives from the left can be found by simply having the span-finding algorithm use intervals of the form $(u_i, u_{i+1}]$, instead of $[u_i, u_{i+1})$. In the preceding example, with $p = 2$ and $U = \{0,0,0,1,2,3,4,4,5,5,5\}$, if $u = 2$ then span $i = 3$ yields derivatives from the left, and $i = 4$ yields derivatives from the right.

Exercises

2.1. Consider the linear and quadratic functions computed earlier and shown in Figures 2.5 and 2.6. Substitute $u = {}^5\!/_2$ into the polynomial equations to obtain $N_{3,1}(5/2)$, $N_{4,1}(5/2)$, $N_{2,2}(5/2)$, $N_{3,2}(5/2)$, and $N_{4,2}(5/2)$. What do you notice about the sum of the two linear, and the sum of the three quadratic functions?

2.2. Consider the quadratic functions of Figure 2.6. Using the polynomial expressions for $N_{3,2}(u)$, evaluate the function and its first and second derivatives at $u = 2$ from both the left and right. Observe the continuity. Does Property P2.5 hold? Do the same with $N_{4,2}(u)$ at $u = 4$.

2.3. Let $U = \{0,0,0,0,1,2,3,4,4,5,5,5,5\}$. How does this change the degree 0, 1, and 2 functions of Figures 2.4–2.6? Compute and sketch the nine cubic basis functions associated with U.

2.4. Consider the function $N_{2,2}(u)$ of Figure 2.5, $N_{2,2}(u) = {}^1\!/_2 u^2$ on $[0,1)$, $-{}^3\!/_2 + 3u - u^2$ on $[1,2)$ and ${}^1\!/_2(3-u)^2$ on $[2,3)$. Use Eq. (2.10) to obtain the expressions for the first and second derivatives of $N_{2,2}(u)$.

2.5. Again consider $N_{2,2}(u)$ of Figure 2.5. Obtain the first derivatives of $N_{2,1}$ and $N_{3,1}$ by differentiating the polynomial expressions directly. Then use these, together with Eq. (2.11), to obtain $N'_{2,2}$.

2.6. Again let $p = 2$, $u = {}^5\!/_2$, and $U = \{0,0,0,1,2,3,4,4,5,5,5\}$. Trace through Algorithm A2.2 by hand to find the values of the three nonzero basis functions. Trace through Algorithm A2.3 to find the first and second derivatives of the basis functions.

2.7. Use the same p and U as in Exercise 2.6, with $u = 2$. Trace through Algorithm A2.3 with $n = 1$, once with $i = 3$, and once with $i = 4$. Then differentiate the appropriate polynomial expressions for the $N_{j,2}$ given in Section 2.2, and evaluate the derivatives from the left and right at $u = 2$. Compare the results with what you obtained from Algorithm A2.3.

2.8. Using the same p and U as in Exercise 2.6, let $u = 4$. Trace through Algorithms A2.2 and A2.3 to convince yourself there are no problems with double knots.

2.9. With the same p and U as in Exercise 2.6, let $u = {}^5\!/_2$. Trace through Algorithm A2.5 and compute the derivatives $N^{(k)}_{4,2}(5/2)$ for $k = 0,1,2$.