CHAPTER
THREE

B-spline Curves and Surfaces

3.1 Introduction

In this chapter we define nonrational B-spline curves and surfaces, study their properties, and derive expressions for their derivatives. For brevity we drop the word nonrational for the remainder of this chapter. The primary goal is to acquire an intuitive understanding of B-spline curves and surfaces, and to that end the reader should carefully study the many examples and figures given in this chapter. We also give algorithms for computing points and derivatives on B-spline curves and surfaces. The use of B-splines to define curves and surfaces for computer-aided geometric design was first proposed by Gordon and Riesenfeld [Gord74b; Ries73]. B-spline techniques are now covered in many books on curves and surfaces – see [DeBo78; Mort85; Bart87; Fari93; Yama88; Hosc93; Su89; Roge90; Beac91].

3.2 The Definition and Properties of B-spline Curves

A *pth-degree B-spline curve* is defined by

$$\mathbf{C}(u) = \sum_{i=0}^{n} N_{i,p}(u)\,\mathbf{P}_i \qquad a \le u \le b \tag{3.1}$$

where the $\{\mathbf{P}_i\}$ are the *control points*, and the $\{N_{i,p}(u)\}$ are the pth-degree B-spline basis functions (Eq. [2.5]) defined on the nonperiodic (and nonuniform) knot vector

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

($m+1$ knots). Unless stated otherwise, we assume that $a = 0$ and $b = 1$. The polygon formed by the $\{\mathbf{P}_i\}$ is called the *control polygon*. Examples of B-spline curves (in some cases together with their corresponding basis functions) are shown in Figures 3.1–3.14.

Three steps are required to compute a point on a B-spline curve at a fixed u value:

1. find the knot span in which u lies (Algorithm A2.1);
2. compute the nonzero basis functions (Algorithm A2.2);
3. multiply the values of the nonzero basis functions with the corresponding control points.

Consider Example Ex2.3 of Section 2.5, with $U = \{0,0,0,1,2,3,4,4,5,5,5\}$, $u = 5/2$, and $p = 2$. Then $u \in [u_4, u_5)$, and

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

Multiplying with the control points yields

$$\mathbf{C}\left(\frac{5}{2}\right) = \frac{1}{8}\mathbf{P}_2 + \frac{6}{8}\mathbf{P}_3 + \frac{1}{8}\mathbf{P}_4$$

The algorithm follows.

```
ALGORITHM A3.1:
  CurvePoint(n,p,U,P,u,C)
    { /*  Compute curve point  */
      /*  Input:  n,p,U,P,u  */
      /*  Output: C  */
    span = FindSpan(n,p,u,U);
    BasisFuns(span,u,p,U,N);
    C = 0.0;
    for (i=0; i<=p; i++)
      C = C + N[i]*P[span-p+i];
    }
```

We now list a number of properties of B-spline curves. These properties follow from those given in Chapter 2 for the functions $N_{i,p}(u)$. Let $\mathbf{C}(u)$ be defined by Eq. (3.1).

P3.1 If $n = p$ and $U = \{0,\ldots,0,1,\ldots,1\}$, then $\mathbf{C}(u)$ is a Bézier curve (Figure 3.1);

P3.2 $\mathbf{C}(u)$ is a piecewise polynomial curve (since the $N_{i,p}(u)$ are piecewise polynomials); the degree, p, number of control points, $n+1$, and number of knots, $m+1$, are related by

$$m = n + p + 1 \tag{3.2}$$

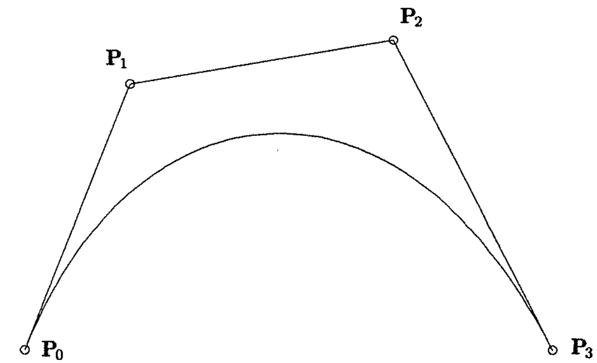


Figure 3.1. A cubic B-spline curve on $U = \{0,0,0,0,1,1,1,1\}$, i.e., a cubic Bézier curve.

(see Section 2.4). Figures 3.2 and 3.3 show basis functions and sections of the B-spline curves corresponding to the individual knot spans; in both figures the alternating solid/dashed segments correspond to the different polynomials (knot spans) defining the curve.

P3.3 Endpoint interpolation: $\mathbf{C}(0) = \mathbf{P}_0$ and $\mathbf{C}(1) = \mathbf{P}_n$;

P3.4 Affine invariance: an affine transformation is applied to the curve by applying it to the control points. Let $\mathbf{r}$ be a point in $\mathcal{E}^3$ (three-dimensional Euclidean space). An *affine transformation*, denoted by Φ, maps $\mathcal{E}^3$ into $\mathcal{E}^3$ and has the form

$$\Phi(\mathbf{r}) = A\mathbf{r} + \mathbf{v}$$

where A is a 3×3 matrix and $\mathbf{v}$ is a vector. Affine transformations include translations, rotations, scalings, and shears. The affine invariance property for B-spline curves follows from the partition of unity property of the $N_{i,p}(u)$. Thus, let $\mathbf{r} = \sum \alpha_i \mathbf{p}_i$, where $\mathbf{p}_i \in \mathcal{E}^3$ and $\sum \alpha_i = 1$. Then

$$\Phi(\mathbf{r}) = \Phi\left(\sum \alpha_i \mathbf{p}_i\right) = A\left(\sum \alpha_i \mathbf{p}_i\right) + \mathbf{v} = \sum \alpha_i A\mathbf{p}_i + \sum \alpha_i \mathbf{v}$$
$$= \sum \alpha_i (A\mathbf{p}_i + \mathbf{v}) = \sum \alpha_i \Phi(\mathbf{p}_i)$$

P3.5 Strong convex hull property: the curve is contained in the convex hull of its control polygon; in fact, if $u \in [u_i, u_{i+1})$, $p \le i < m-p-1$, then $\mathbf{C}(u)$ is in the convex hull of the control points $\mathbf{P}_{i-p}, \ldots, \mathbf{P}_i$ (Figures 3.4, 3.5, and 3.6). This follows from the nonnegativity and partition of unity properties of the $N_{i,p}(u)$ (Properties P2.3 and P2.4), and the property that $N_{j,p}(u) = 0$ for $j < i-p$ and $j > i$ when $u \in [u_i, u_{i+1})$ (Property P2.2). Figure 3.6 shows how to construct a quadratic curve containing a straight line segment. Since $\mathbf{P}_2$, $\mathbf{P}_3$, and $\mathbf{P}_4$ are colinear, the strong

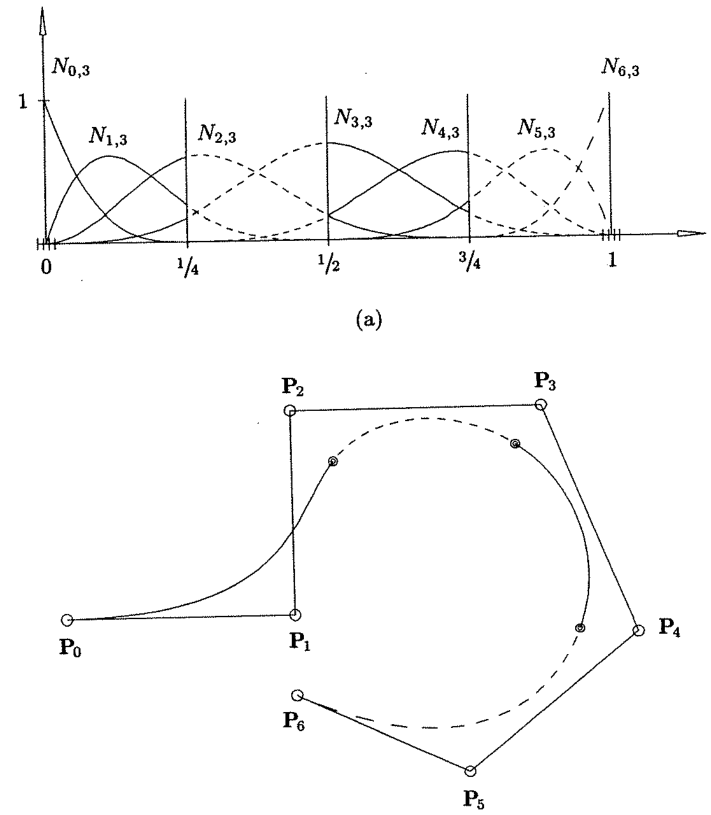


Figure 3.2. (a) Cubic basis functions $U = \{0, 0, 0, 0, 1/4, 1/2, 3/4, 1, 1, 1, 1\}$; (b) a cubic curve using the basis functions of Figure 3.2a.

convex hull property forces the curve to be a straight line segment from $\mathbf{C}(2/5)$ to $\mathbf{C}(3/5)$;

P3.6 Local modification scheme: moving $\mathbf{P}_i$ changes $\mathbf{C}(u)$ only in the interval $[u_i, u_{i+p+1})$ (Figure 3.7). This follows from the fact that $N_{i,p}(u) = 0$ for $u \notin [u_i, u_{i+p+1})$ (Property P2.1).

P3.7 The control polygon represents a piecewise linear approximation to the curve; this approximation is improved by knot insertion or degree elevation (see Chapter 5). As a general rule, the lower the degree, the closer a B-spline curve follows its control polygon (see Figures 3.8 and 3.9). The curves of Figure 3.9 are defined using the same six control points, and the knot vectors

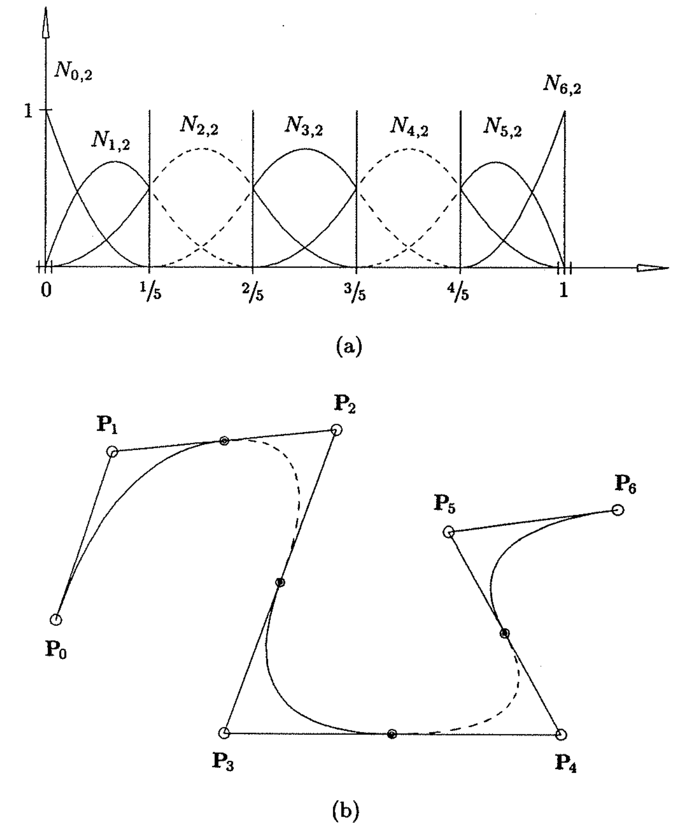


Figure 3.3. (a) Quadratic basis functions on $U = \{0, 0, 0, 1/5, 2/5, 3/5, 4/5, 1, 1, 1\}$; (b) a quadratic curve using the basis functions of Figure 3.3a.

$$p = 1 : U = \left\{0, 0, \frac{1}{5}, \frac{2}{5}, \frac{3}{5}, \frac{4}{5}, 1, 1\right\}$$

$$p = 2 : U = \left\{0, 0, 0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1, 1, 1\right\}$$

$$p = 3 : U = \left\{0, 0, 0, 0, \frac{1}{3}, \frac{2}{3}, 1, 1, 1, 1\right\}$$

$$p = 4 : U = \left\{0, 0, 0, 0, 0, \frac{1}{2}, 1, 1, 1, 1, 1\right\}$$

$$p = 5 : U = \{0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1\}$$

The reason for this phenomenon is intuitive: the lower the degree, the fewer the control points that are contributing to the computation of

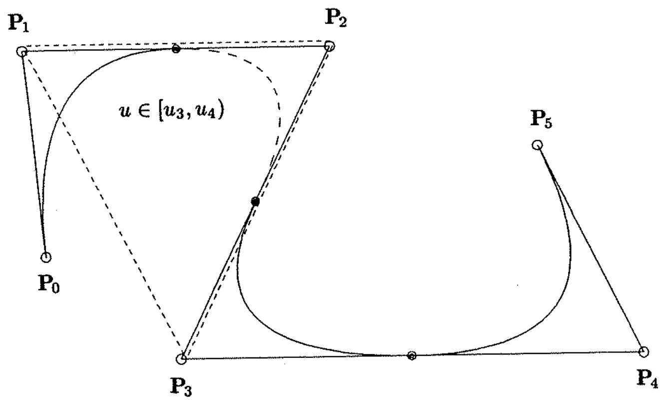


Figure 3.4. The strong convex hull property for a quadratic B-spline curve; for $u \in [u_i, u_{i+1})$, $\mathbf{C}(u)$ is in the triangle $\mathbf{P}_{i-2}\mathbf{P}_{i-1}\mathbf{P}_i$.

$\mathbf{C}(u_0)$ for any given u_0. The extreme case is $p = 1$, for which every point $\mathbf{C}(u)$ is just a linear interpolation between two control points. In this case, the curve *is* the control polygon;

P3.8 Moving along the curve from $u = 0$ to $u = 1$, the $N_{i,p}(u)$ functions act like switches; as u moves past a knot, one $N_{i,p}(u)$ (and hence the corresponding $\mathbf{P}_i$) switches off, and the next one switches on (Figures 3.2 and 3.3).

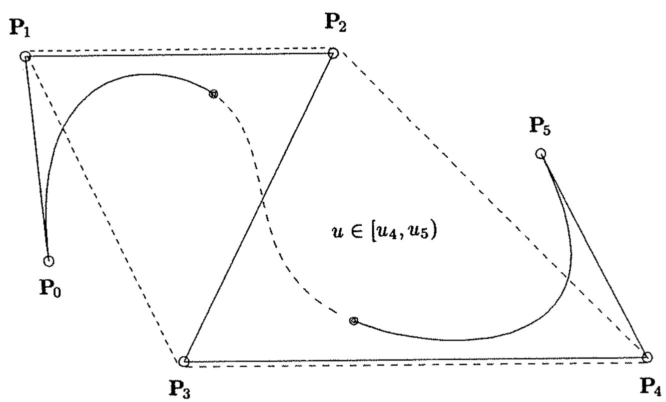


Figure 3.5. The strong convex hull property for a cubic B-spline curve; for $u \in [u_i, u_{i+1})$, $\mathbf{C}(u)$ is in the quadrilateral $\mathbf{P}_{i-3}\mathbf{P}_{i-2}\mathbf{P}_{i-1}\mathbf{P}_i$.

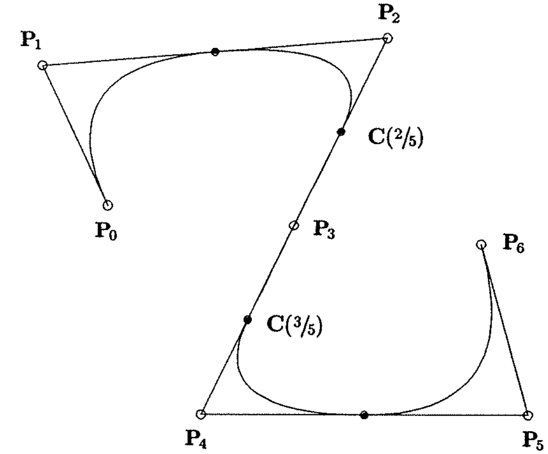


Figure 3.6. A quadratic B-spline curve on $U = \{0, 0, 0, 1/5, 2/5, 3/5, 4/5, 1, 1, 1\}$. The curve is a straight line between $\mathbf{C}(2/5)$ and $\mathbf{C}(3/5)$.

P3.9 Variation diminishing property: no plane has more intersections with the curve than with the control polygon (replace the word plane with line, for two-dimensional curves) – see [Lane83] for proof;

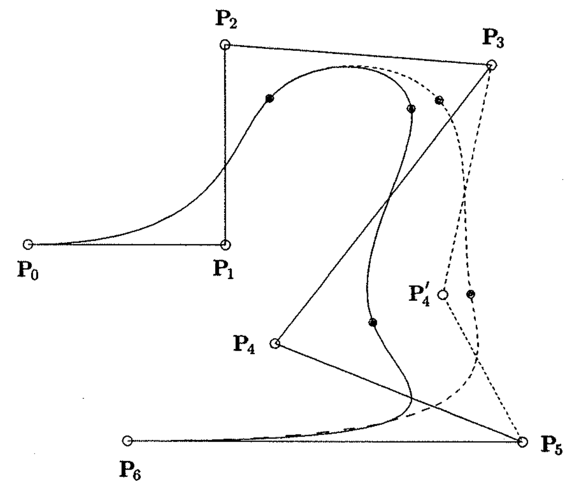


Figure 3.7. A cubic curve on $U = \{0, 0, 0, 0, 1/4, 1/2, 3/4, 1, 1, 1, 1\}$; moving $\mathbf{P}_4$ (to $\mathbf{P}'_4$) changes the curve in the interval $[1/4, 1)$.

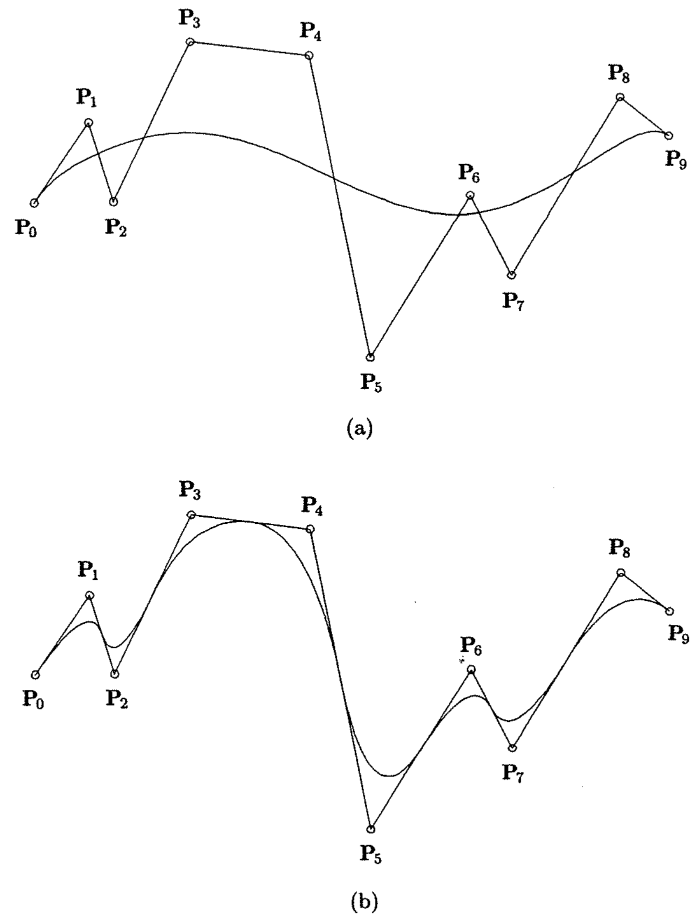


Figure 3.8. B-spline curves. (a) A ninth-degree Bézier curve on the knot vector $U = \{0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,1,1\}$; (b) a quadratic curve using the same control polygon defined on $U = \{0,0,0,{}^1/_8,{}^2/_8,{}^3/_8,{}^4/_8,{}^5/_8,{}^6/_8,{}^7/_8,1,1,1\}$.

P3.10 The continuity and differentiability of $\mathbf{C}(u)$ follow from that of the $N_{i,p}(u)$ (since $\mathbf{C}(u)$ is just a linear combination of the $N_{i,p}(u)$). Thus, $\mathbf{C}(u)$ is infinitely differentiable in the interior of knot intervals, and it is at least $p-k$ times continuously differentiable at a knot of multiplicity k. Figure 3.10 shows a quadratic curve ($p = 2$). The curve is $\mathbf{C}^1$ continuous (the first derivative is continuous but the second is not) at all interior knots of multiplicity 1. At the double knot, $u = {}^4/_5$, $\mathbf{C}(u)$ is only $\mathbf{C}^0$ continuous, thus there is a cusp (a visual discontinuity). Figure 3.11 shows a

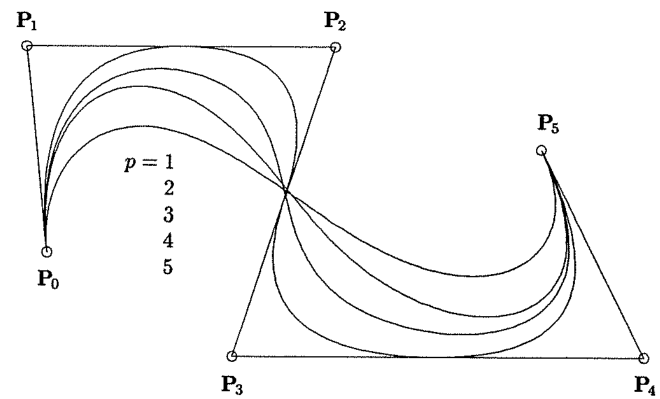


Figure 3.9. B-spline curves of different degree, using the same control polygon.

quadratic curve defined on the same knot vector. Hence, the two curves use the same basis functions, $N_{i,p}(u)$, for their definitions. But the curve of Figure 3.11 is $\mathbf{C}^1$ continuous at $u = {}^4/_5$; this is not obvious but can be seen using the derivative expression given in Section 3.3. This is simply a consequence of the fact that discontinuous functions can sometimes be combined in such a way that the result is continuous. Notice that $\mathbf{P}_4$, $\mathbf{P}_5$, and $\mathbf{P}_6$ are colinear, and length$(\mathbf{P}_4\mathbf{P}_5) =$ length$(\mathbf{P}_5\mathbf{P}_6)$. Figure 3.12 shows a cubic curve which is $\mathbf{C}^2$ continuous at $u = {}^1/_4$ and $u = {}^1/_2$, but only $\mathbf{C}^1$ continuous at the double knot $u = {}^3/_4$. The eye detects discontinuities in the second derivative but probably not in third and higher derivatives. Thus, cubics are generally adequate for visual purposes.

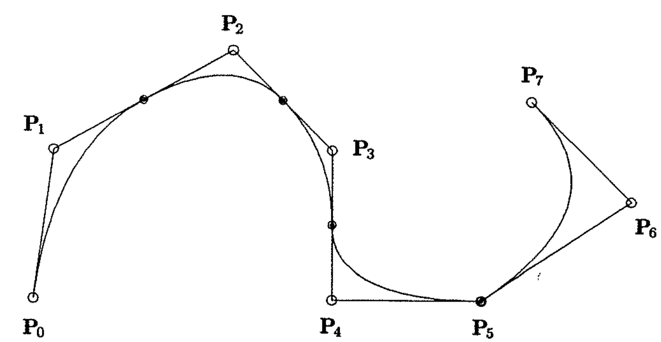


Figure 3.10. A quadratic curve on $U = \{0,0,0,{}^1/_5,{}^2/_5,{}^3/_5,{}^4/_5,{}^4/_5,1,1,1\}$ with a cusp at $u = {}^4/_5$.

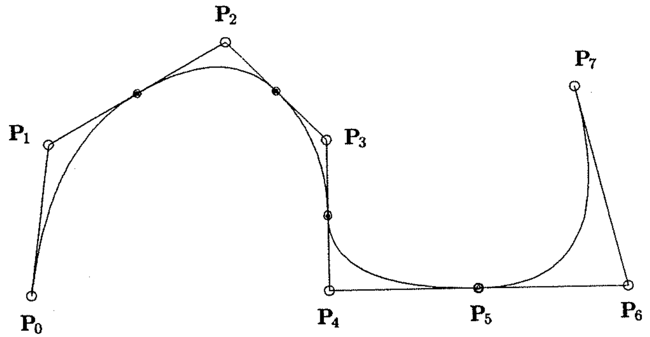


Figure 3.11. A quadratic curve on $U = \{0,0,0,{}^1/_5,{}^2/_5,{}^3/_5,{}^4/_5,{}^4/_5,1,1,1\}$; the first derivative is continuous at $u = {}^4/_5$.

P3.11 It is possible (and sometimes useful) to use multiple (coincident) control points. Figure 3.13 shows a quadratic curve with a double control point, $\mathbf{P}_2 = \mathbf{P}_3$. The interesting portion of this curve lies between $\mathbf{C}({}^1/_4)$ and $\mathbf{C}({}^3/_4)$. Indeed, $\mathbf{C}({}^1/_2) = \mathbf{P}_2 = \mathbf{P}_3$, and the curve segments between $\mathbf{C}({}^1/_4)$ and $\mathbf{C}({}^1/_2)$, and $\mathbf{C}({}^1/_2)$ and $\mathbf{C}({}^3/_4)$, are straight lines. This follows from Property P3.5, e.g., $\mathbf{C}(u)$ is in the convex hull of $\mathbf{P}_1\mathbf{P}_2\mathbf{P}_3$ (a line) if $u \in [{}^1/_4, {}^1/_2)$. Furthermore, since the knot $u = {}^1/_2$ has multiplicity $= 1$, the curve must be $\mathbf{C}^1$ continuous there, even though it has a cusp (visual discontinuity). This is a result of the magnitude of the first derivative vector going to zero (continuously) at $u = {}^1/_2$. In the next section we see that the derivative at $u = {}^1/_2$ is proportional to the difference, $\mathbf{P}_3 - \mathbf{P}_2$. Figures 3.14a and 3.14b are cubic examples using the same control polygon, including the double control point, $\mathbf{P}_2 = \mathbf{P}_3$, but with different knot vectors.

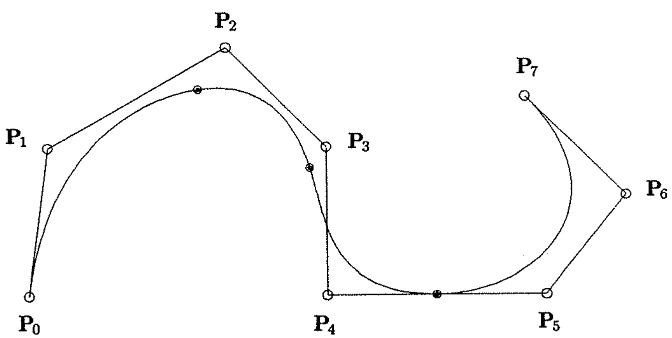


Figure 3.12. A cubic curve on $U = \{0,0,0,0,{}^1/_4,{}^1/_2,{}^3/_4,{}^3/_4,1,1,1,1\}$, $\mathbf{C}^2$ continuous at $u = {}^1/_4$ and $u = {}^1/_2$, and $\mathbf{C}^1$ continuous at $u = {}^3/_4$.

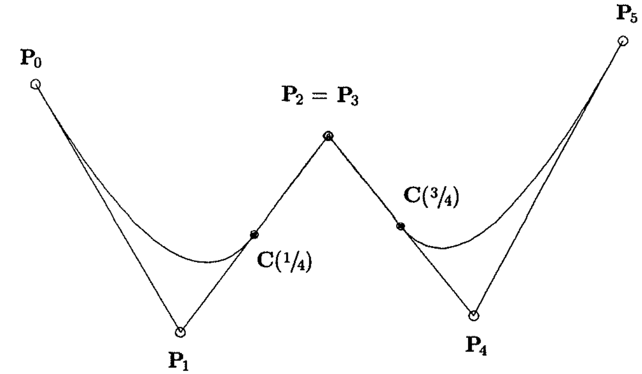


Figure 3.13. A quadratic curve on $U = \{0,0,0,{}^1/_4,{}^1/_2,{}^3/_4,1,1,1\}$; $\mathbf{P}_2 = \mathbf{P}_3$ is a double control point.

3.3 The Derivatives of a B-spline Curve

Let $\mathbf{C}^{(k)}(u)$ denote the kth derivative of $\mathbf{C}(u)$. If u is fixed, we can obtain $\mathbf{C}^{(k)}(u)$ by computing the kth derivatives of the basis functions (see Eqs. [2.7], [2.9], and [2.10] and Algorithm A2.3). In particular

$$\mathbf{C}^{(k)}(u) = \sum_{i=0}^{n} N_{i,p}^{(k)}(u)\,\mathbf{P}_i \tag{3.3}$$

Consider the example of Section 2.5, with $p = 2$, $U = \{0,0,0,1,2,3,4,4,5,5,5\}$, and $u = {}^5/_2$. From Eq. (2.7) we have

$$N'_{2,2}\left(\frac{5}{2}\right) = 0 - \frac{2}{3-1}\frac{1}{2} = -\frac{1}{2}$$

$$N'_{3,2}\left(\frac{5}{2}\right) = \frac{2}{3-1}\frac{1}{2} - \frac{2}{4-2}\frac{1}{2} = 0$$

$$N'_{4,2}\left(\frac{5}{2}\right) = \frac{2}{4-2}\frac{1}{2} - 0 = \frac{1}{2}$$

It follows that

$$\mathbf{C}'\left(\frac{5}{2}\right) = -\frac{1}{2}\mathbf{P}_2 + \frac{1}{2}\mathbf{P}_4$$

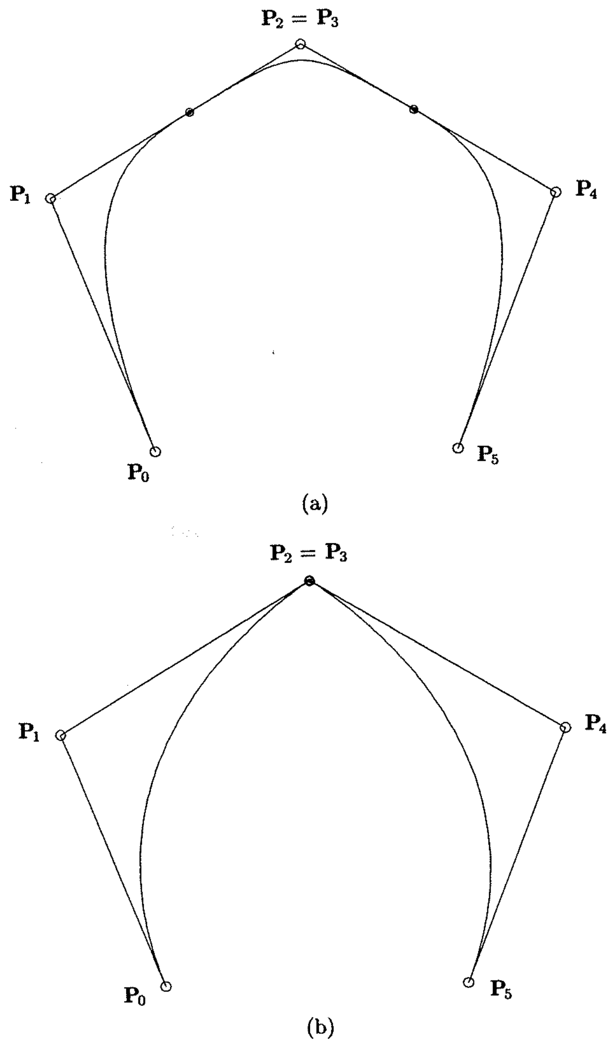


Figure 3.14. Cubic curves with double control point $\mathbf{P}_2 = \mathbf{P}_3$. (a) $U = \{0,0,0,0,{}^1\!/\!_4,{}^3\!/\!_4,1,1,1,1\}$; (b) $U = \{0,0,0,0,{}^1\!/\!_2,{}^1\!/\!_2,1,1,1,1\}$.

An algorithm to compute the point on a B-spline curve and all derivatives up to and including the dth at a fixed u value follows. We allow $d > p$, although the derivatives are 0 in this case (for nonrational curves); these derivatives are necessary for rational curves. Input to the algorithm is `u,d`, and the B-spline curve, defined (throughout the remainder of this book) by

- `n` : the number of control points is $n+1$;
- `p` : the degree of the curve;
- `U` : the knots;
- `P` : the control points.

Output is the array `CK[]`, where `CK[k]` is the kth derivative, $0 \le k \le d$. We use Algorithms `A2.1` and `A2.3`. A local array, `nders[][]`, is used to store the derivatives of the basis functions.

```
ALGORITHM A3.2:
  CurveDerivsAlg1(n,p,U,P,u,d,CK)
    { /*  Compute curve derivatives  */
      /*  Input:  n,p,U,P,u,d  */
      /*  Output: CK  */
    du = min(d,p);
    for (k=p+1; k<=d; k++)   CK[k] = 0.0;
    span = FindSpan(n,p,u,U);
    DersBasisFuns(span,u,p,du,U,nders);
    for (k=0; k<=du; k++)
      {
      CK[k] = 0.0;
      for (j=0; j<=p; j++)
        CK[k] = CK[k] + nders[k][j]*P[span-p+j];
      }
    }
```

Now instead of fixing u, we want to formally differentiate the pth-degree B-spline curve,

$$\mathbf{C}(u) = \sum_{i=0}^{n} N_{i,p}(u)\,\mathbf{P}_i$$

defined on the knot vector

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

From Eqs. (3.3) and (2.7) we obtain

$$\mathbf{C}'(u) = \sum_{i=0}^{n} N'_{i,p}(u)\,\mathbf{P}_i$$

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

$$= \left(p \sum_{i=-1}^{n-1} N_{i+1,p-1}(u) \frac{\mathbf{P}_{i+1}}{u_{i+p+1}-u_{i+1}} \right)$$

$$- \left(p \sum_{i=0}^{n} N_{i+1,p-1}(u) \frac{\mathbf{P}_i}{u_{i+p+1} - u_{i+1}} \right)$$

$$= p \frac{N_{0,p-1}(u)\mathbf{P}_0}{u_p - u_0} + p \sum_{i=0}^{n-1} N_{i+1,p-1}(u) \frac{(\mathbf{P}_{i+1} - \mathbf{P}_i)}{u_{i+p+1} - u_{i+1}} - p \frac{N_{n+1,p-1}(u)\mathbf{P}_n}{u_{n+p+1} - u_{n+1}}$$

The first and last terms evaluate to $0/0$, which is 0 by definition. Thus

$$\mathbf{C}'(u) = p \sum_{i=0}^{n-1} N_{i+1,p-1}(u) \frac{(\mathbf{P}_{i+1} - \mathbf{P}_i)}{u_{i+p+1} - u_{i+1}} = \sum_{i=0}^{n-1} N_{i+1,p-1}(u)\, \mathbf{Q}_i$$

where
$$\mathbf{Q}_i = p \frac{\mathbf{P}_{i+1} - \mathbf{P}_i}{u_{i+p+1} - u_{i+1}} \tag{3.4}$$

Now let U' be the knot vector obtained by dropping the first and last knots from U, i.e.

$$U' = \{\underbrace{0, \ldots, 0}_{p}, u_{p+1}, \ldots, u_{m-p-1}, \underbrace{1, \ldots, 1}_{p}\} \tag{3.5}$$

(U' has $m-1$ knots). Then it is easy to check that the function $N_{i+1,p-1}(u)$, computed on U, is equal to $N_{i,p-1}(u)$ computed on U'. Thus

$$\mathbf{C}'(u) = \sum_{i=0}^{n-1} N_{i,p-1}(u)\, \mathbf{Q}_i \tag{3.6}$$

where the $\mathbf{Q}_i$ are defined by Eq. (3.4), and the $N_{i,p-1}(u)$ are computed on U'. Hence, $\mathbf{C}'(u)$ is a $(p-1)$th-degree B-spline curve.

Examples

Ex3.1 Let $\mathbf{C}(u) = \sum_{i=0}^{4} N_{i,2}(u)\mathbf{P}_i$ be a quadratic curve defined on

$$U = \{0, 0, 0, 2/5, 3/5, 1, 1, 1\}$$

Then $U' = \{0, 0, 2/5, 3/5, 1, 1\}$ and $\mathbf{C}'(u) = \sum_{i=0}^{3} N_{i,1}(u)\, \mathbf{Q}_i$, where

$$\mathbf{Q}_0 = \frac{2(\mathbf{P}_1 - \mathbf{P}_0)}{\frac{2}{5} - 0} = 5(\mathbf{P}_1 - \mathbf{P}_0)$$

$$\mathbf{Q}_1 = \frac{2(\mathbf{P}_2 - \mathbf{P}_1)}{\frac{3}{5} - 0} = \frac{10}{3}(\mathbf{P}_2 - \mathbf{P}_1)$$

$$\mathbf{Q}_2 = \frac{2(\mathbf{P}_3 - \mathbf{P}_2)}{1 - \frac{2}{5}} = \frac{10}{3}(\mathbf{P}_3 - \mathbf{P}_2)$$

$$\mathbf{Q}_3 = \frac{2(\mathbf{P}_4 - \mathbf{P}_3)}{1 - \frac{3}{5}} = 5(\mathbf{P}_4 - \mathbf{P}_3)$$

$\mathbf{C}(u)$ and $\mathbf{C}'(u)$ are shown in Figures 3.15a and 3.15b, respectively.

Ex3.2 Let $\mathbf{C}(u) = \sum_{i=0}^{6} N_{i,3}(u)\mathbf{P}_i$ be a cubic curve defined on

$$U = \{0, 0, 0, 0, 2/5, 3/5, 3/5, 1, 1, 1, 1\}$$

Then $U' = \{0, 0, 0, 2/5, 3/5, 3/5, 1, 1, 1\}$ and $\mathbf{C}'(u) = \sum_{i=0}^{5} N_{i,2}(u)\, \mathbf{Q}_i$,

where
$$\mathbf{Q}_0 = \frac{3(\mathbf{P}_1 - \mathbf{P}_0)}{\frac{1}{3} - 0} = 9(\mathbf{P}_1 - \mathbf{P}_0)$$

$$\mathbf{Q}_1 = \frac{3(\mathbf{P}_2 - \mathbf{P}_1)}{\frac{2}{3} - 0} = \frac{9}{2}(\mathbf{P}_2 - \mathbf{P}_1)$$

$$\mathbf{Q}_2 = \frac{3(\mathbf{P}_3 - \mathbf{P}_2)}{\frac{2}{3} - 0} = \frac{9}{2}(\mathbf{P}_3 - \mathbf{P}_2)$$

$$\mathbf{Q}_3 = \frac{3(\mathbf{P}_4 - \mathbf{P}_3)}{1 - \frac{1}{3}} = \frac{9}{2}(\mathbf{P}_4 - \mathbf{P}_3)$$

$$\mathbf{Q}_4 = \frac{3(\mathbf{P}_5 - \mathbf{P}_4)}{1 - \frac{2}{3}} = 9(\mathbf{P}_5 - \mathbf{P}_4)$$

$$\mathbf{Q}_5 = \frac{3(\mathbf{P}_6 - \mathbf{P}_5)}{1 - \frac{2}{3}} = 9(\mathbf{P}_6 - \mathbf{P}_5)$$

$\mathbf{C}(u)$ and $\mathbf{C}'(u)$ are shown in Figures 3.16a and 3.16b, respectively. Notice that $\mathbf{C}'(u)$ is a quadratic curve with a cusp at the double knot $u = 3/5$.

Ex3.3 Recalling that a pth-degree Bézier curve is a B-spline curve on $U = \{0, \ldots, 0, 1, \ldots, 1\}$ (no interior knots), Eq. (3.4) reduces to $\mathbf{Q}_i = p(\mathbf{P}_{i+1} - \mathbf{P}_i)$ for $0 \le i \le n-1$. Since $n = p$ and $N_{i,p-1}(u) = B_{i,n-1}(u)$, the Bernstein polynomials, Eq. (3.6) is equivalent to Eq. (1.9).

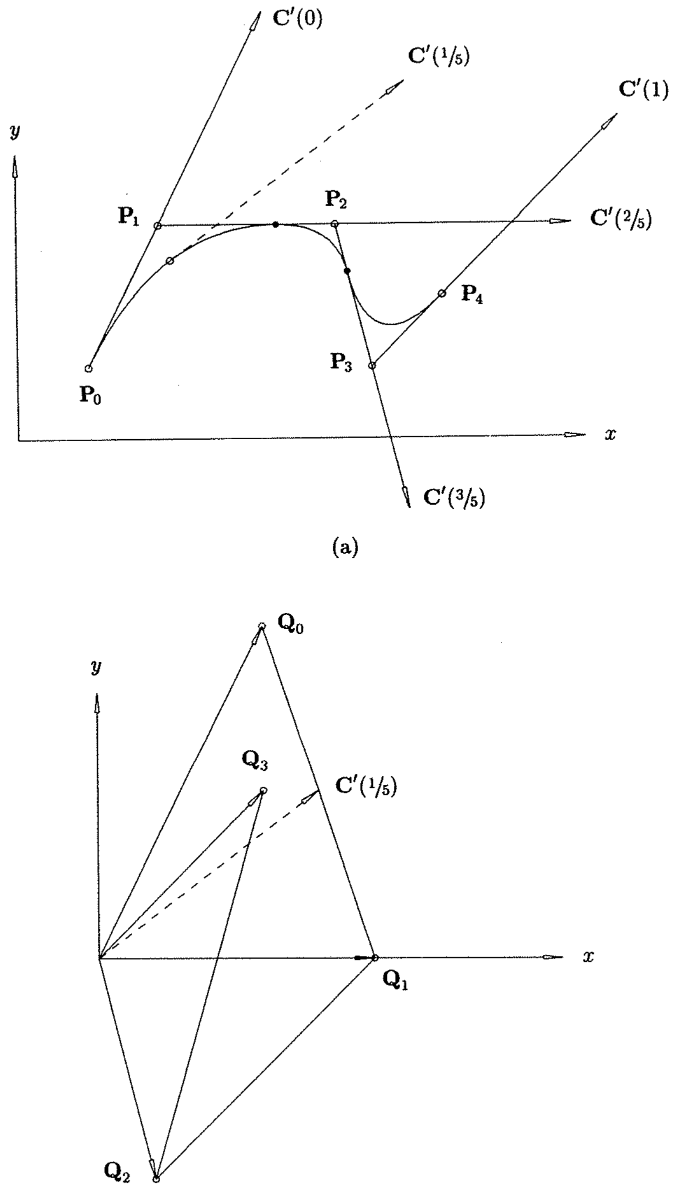
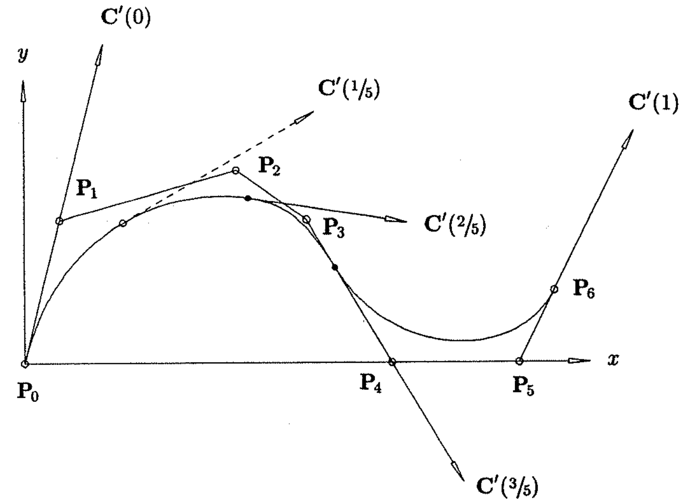


Figure 3.15. (a) A quadratic curve on $U = \{0, 0, 0, {}^2\!/_5, {}^3\!/_5, 1, 1, 1\}$; (b) the derivative of the curve is a first-degree B-spline curve on $U' = \{0, 0, {}^2\!/_5, {}^3\!/_5, 1, 1\}$.



(a)

Figure 3.16. (a) A cubic curve on $U = \{0, 0, 0, 0, {}^2\!/_5, {}^3\!/_5, {}^3\!/_5, 1, 1, 1, 1\}$; (b) the quadratic derivative curve on $U' = \{0, 0, 0, {}^2\!/_5, {}^3\!/_5, {}^3\!/_5, 1, 1, 1\}$.

The first derivatives at the endpoints of a B-spline curve are given by

$$\mathbf{C}'(0) = \mathbf{Q}_0 = \frac{p}{u_{p+1}}(\mathbf{P}_1 - \mathbf{P}_0)$$

$$\mathbf{C}'(1) = \mathbf{Q}_{n-1} = \frac{p}{1 - u_{m-p-1}}(\mathbf{P}_n - \mathbf{P}_{n-1}) \tag{3.7}$$

(see Examples Ex3.1 and Ex3.2, and Figures 3.15(a) and (b) and Figures 3.16 (a) and (b)). Note that in Figures 3.15b and 3.16b the derivative vectors and control point differences are scaled down for better visualization, by 1/2 and by 1/3, respectively.

Since $\mathbf{C}'(u)$ is a B-spline curve, we apply Eqs. (3.4) through (3.6) recursively to obtain higher derivatives. Letting $\mathbf{P}_i^{(0)} = \mathbf{P}_i$, we write

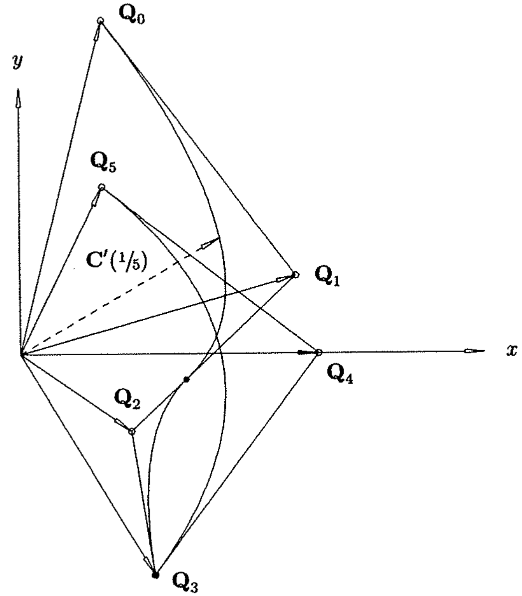
$$\mathbf{C}(u) = \mathbf{C}^{(0)}(u) = \sum_{i=0}^{n} N_{i,p}(u)\mathbf{P}_i^{(0)}$$

Then

$$\mathbf{C}^{(k)}(u) = \sum_{i=0}^{n-k} N_{i,p-k}(u)\mathbf{P}_i^{(k)} \tag{3.8}$$

with

$$\mathbf{P}_i^{(k)} = \begin{cases} \mathbf{P}_i & k = 0 \\ \dfrac{p-k+1}{u_{i+p+1} - u_{i+k}}\left(\mathbf{P}_{i+1}^{(k-1)} - \mathbf{P}_i^{(k-1)}\right) & k > 0 \end{cases}$$



(b)

Figure 3.16. (*Continued.*)

and $$U^{(k)} = \{\underbrace{0,\ldots,0}_{p-k+1}, u_{p+1},\ldots,u_{m-p-1},\underbrace{1,\ldots,1}_{p-k+1}\}$$

Algorithm A3.3 is a nonrecursive implementation of Eq. (3.8). It computes the control points of all derivative curves up to and including the dth derivative ($d \le p$). On output, PK[k][i] is the ith control point of the kth derivative curve, where $0 \le k \le d$ and $r_1 \le i \le r_2 - k$. If $r_1 = 0$ and $r_2 = n$, all control points are computed.

```
ALGORITHM A3.3
  CurveDerivCpts(n,p,U,P,d,r1,r2,PK)
    { /*  Compute control points of curve derivatives  */
      /*  Input:  n,p,U,P,d,r1,r2  */
      /*  Output: PK  */
    r = r2-r1;
    for (i=0; i<=r; i++)
      PK[0][i] = P[r1+i];
    for (k=1; k<=d; k++)
      {
      tmp = p-k+1;
```



```
      for (i=0; i<=r-k; i++)
        PK[k][i] = tmp*(PK[k-1][i+1]-PK[k-1][i])/
              (U[r1+i+p+1]-U[r1+i+k]);
      }
    }
```

Using Eq. (3.8), we compute the second derivative at the endpoint, $u = 0$, of a B-spline curve ($p > 1$)

$$\mathbf{C}^{(2)}(0) = \mathbf{P}_0^{(2)} = \frac{p-2+1}{u_{p+1}-u_2}\left(\mathbf{P}_1^{(1)} - \mathbf{P}_0^{(1)}\right)$$

$$= \frac{p-1}{u_{p+1}-u_2}\left[\frac{p}{u_{p+2}-u_2}\left(\mathbf{P}_2^{(0)} - \mathbf{P}_1^{(0)}\right) - \frac{p}{u_{p+1}-u_1}\left(\mathbf{P}_1^{(0)} - \mathbf{P}_0^{(0)}\right)\right]$$

From $u_1 = u_2 = 0$ it follows that

$$\mathbf{C}^{(2)}(0) = \frac{p(p-1)}{u_{p+1}}\left[\frac{\mathbf{P}_0}{u_{p+1}} - \frac{(u_{p+1}+u_{p+2})\mathbf{P}_1}{u_{p+1}u_{p+2}} + \frac{\mathbf{P}_2}{u_{p+2}}\right] \tag{3.9}$$

Analogously,

$$\mathbf{C}^{(2)}(1) = \frac{p(p-1)}{1-u_{m-p-1}} \times$$

$$\left[\frac{\mathbf{P}_n}{1-u_{m-p-1}} - \frac{(2-u_{m-p-1}-u_{m-p-2})\mathbf{P}_{n-1}}{(1-u_{m-p-1})(1-u_{m-p-2})} + \frac{\mathbf{P}_{n-2}}{1-u_{m-p-2}}\right] \tag{3.10}$$

Notice that for Bézier curves these equations reduce to the corresponding expressions of Eq. (1.10). Figure 3.17 shows the quadratic curve of Figure 3.15a with the vectors $\mathbf{C}^{(2)}(0)$ and $\mathbf{C}^{(2)}(1)$. $\mathbf{C}^{(2)}(u)$ is a piecewise zeroth-degree curve, i.e., it is a constant (but different) vector on each of the three intervals $[0, 2/5)$, $[2/5, 3/5)$, and $[3/5, 1]$.

We close this section with another algorithm to compute the point on a B-spline curve and all derivatives up to and including the dth derivative at a fixed u value (compare with Algorithm A3.2). The algorithm is based on Eq. (3.8) and Algorithm A3.3. We assume a routine, AllBasisFuns, which is a simple modification of BasisFuns (Algorithm A2.2), to return all nonzero basis functions of all degrees from 0 up to p. In particular, N[j][i] is the value of the ith-degree basis function, $N_{\mathrm{span}-i+j,i}(u)$, where $0 \le i \le p$ and $0 \le j \le i$.

```
ALGORITHM A3.4
  CurveDerivsAlg2(n,p,U,P,u,d,CK)
    { /*  Compute curve derivatives  */
      /*  Input:  n,p,U,P,u,d  */
      /*  Output: CK  */
    du = min(d,p);
```

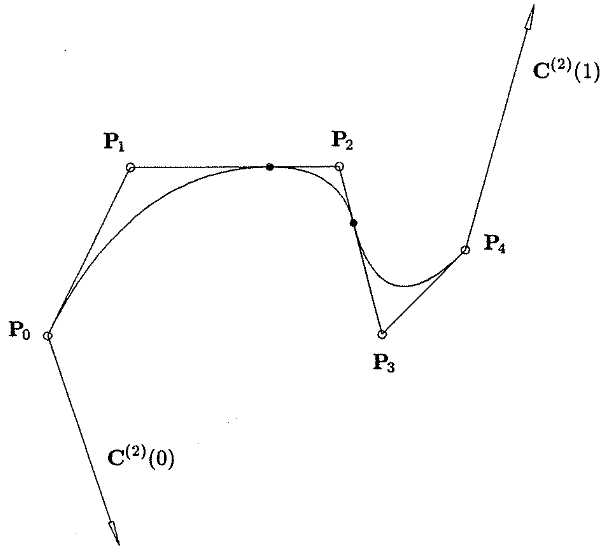


Figure 3.17. The second derivatives at the endpoints of the curve of Figure 3.15a.

```
for (k=p+1; k<=d; k++)    CK[k] = 0.0;
span = FindSpan(n,p,u,U);
AllBasisFuns(span,u,p,U,N);
CurveDerivCpts(n,p,U,P,du,span-p,span,PK);
for (k=0; k<=du; k++)
  {
  CK[k] = 0.0;
  for (j=0; j<=p-k; j++)
    CK[k] = CK[k] + N[j][p-k]*PK[k][j];
  }
}
```

Figure 3.18 shows a cubic curve with first, second, and third derivatives computed at $u = 2/5$. (The derivatives are scaled down by $2/5$.)

3.4 Definition and Properties of B-spline Surfaces

A B-spline surface is obtained by taking a bidirectional net of control points, two knot vectors, and the products of the univariate B-spline functions

$$\mathbf{S}(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u)N_{j,q}(v)\mathbf{P}_{i,j} \tag{3.11}$$

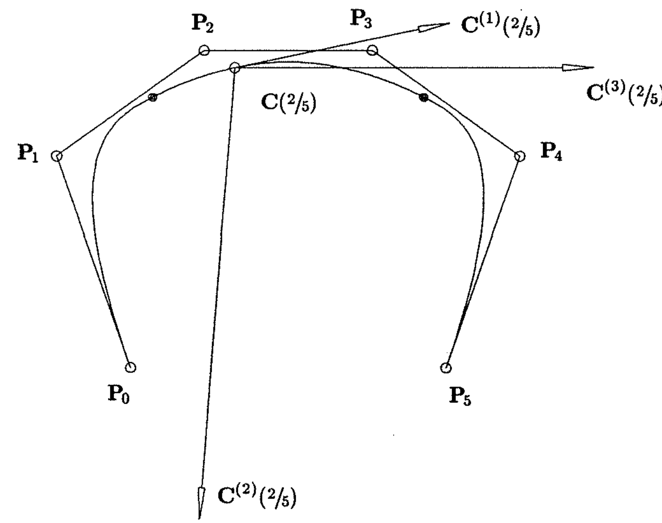


Figure 3.18. A cubic curve on $U = \{0,0,0,0,1/4,3/4,1,1,1,1\}$ with first, second, and third derivatives computed at $u = 2/5$.

with

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

$$V = \{\underbrace{0,\ldots,0}_{q+1}, v_{q+1},\ldots,v_{s-q-1},\underbrace{1,\ldots,1}_{q+1}\}$$

U has $r+1$ knots, and V has $s+1$. Equation (3.2) takes the form

$$r = n+p+1 \quad \text{and} \quad s = m+q+1 \tag{3.12}$$

Let U and $\{N_{i,3}(u)\}$ be the knot vector and cubic basis functions of Figure 3.2a, and $\{N_{j,2}(v)\}$ the quadratic basis functions defined on $V = \{0,0,0,1/5,2/5,3/5,3/5,4/5,1,1,1\}$. Figures 3.19a and 3.19b show the tensor product basis functions $N_{4,3}(u)N_{4,2}(v)$ and $N_{4,3}(u)N_{2,2}(v)$, respectively. Figures 3.20–3.25 show examples of B-spline surfaces.

Five steps are required to compute a point on a B-spline surface at fixed (u,v) parameter values:

1. find the knot span in which u lies, say $u \in [u_i, u_{i+1})$ (Algorithm `A2.1`);
2. compute the nonzero basis functions $N_{i-p,p}(u),\ldots,N_{i,p}(u)$ (`A2.2`);
3. find the knot span in which v lies, say $v \in [v_j, v_{j+1})$ (`A2.1`);
4. compute the nonzero basis functions $N_{j-q,q}(v),\ldots,N_{j,q}(v)$ (`A2.2`);

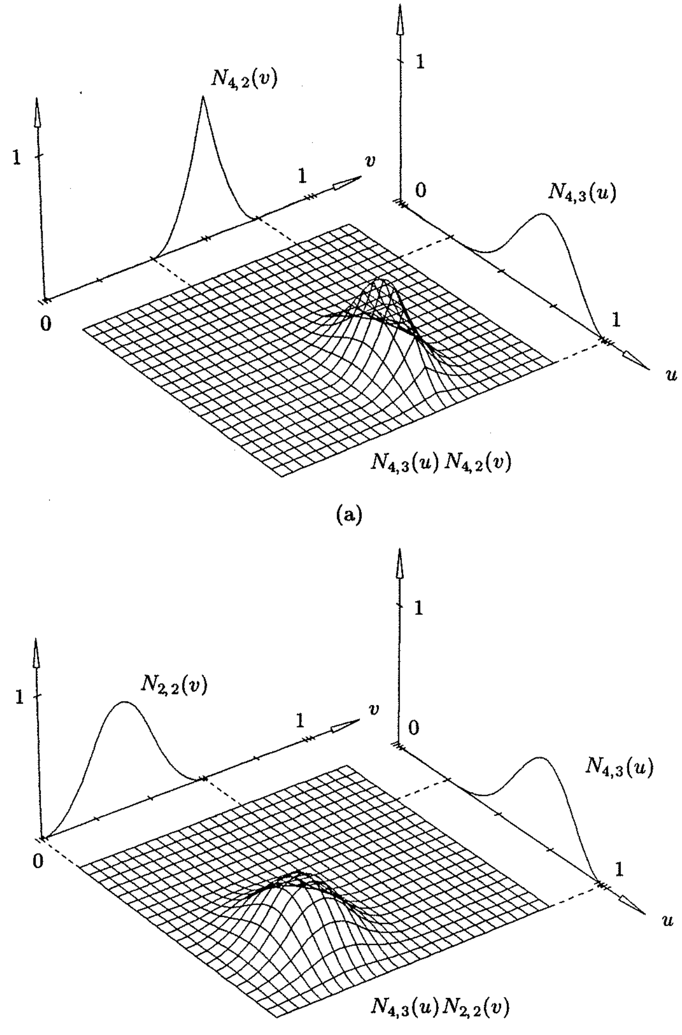


Figure 3.19. Cubic $\times$ quadratic basis functions. (a) $N_{4,3}(u)N_{4,2}(v)$; (b) $N_{4,3}(u)N_{2,2}(v)$; $U = \{0,0,0,0,\,{}^1/_4,\,{}^1/_2,\,{}^3/_4,1,1,1,1\}$ and $V = \{0,0,0,\,{}^1/_5,\,{}^2/_5,\,{}^3/_5,\,{}^3/_5,\,{}^4/_5,1,1,1\}$.

5. multiply the values of the nonzero basis functions with the corresponding control points.

The last step takes the form

$$\mathbf{S}(u,v) = [\,N_{k,p}(u)\,]^T\,[\,\mathbf{P}_{k,l}\,]\,[\,N_{l,q}(v)\,] \qquad i-p \le k \le i,\;\; j-q \le l \le j \tag{3.13}$$

Note that $[\,N_{k,p}(u)\,]^T$ is a $1\times(p+1)$ row vector of scalars, $[\,\mathbf{P}_{k,l}\,]$ is a $(p+1)\times(q+1)$ matrix of control points, and $[\,N_{l,q}(v)\,]$ is a $(q+1)\times 1$ column vector of scalars.

Example

Ex3.4 Let $p = q = 2$ and $\sum_{i=0}^{4}\,\sum_{j=0}^{5} N_{i,2}(u)N_{j,2}(v)\,\mathbf{P}_{i,j}$, with

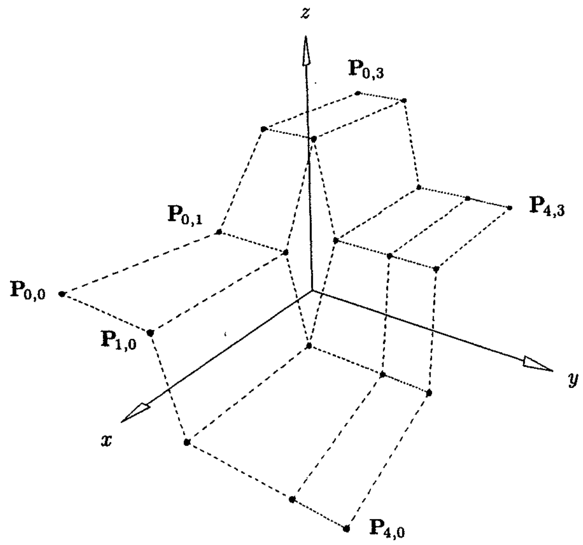
$$U = \{0,0,0,\,{}^2/_5,\,{}^3/_5,1,1,1\}$$
$$V = \{0,0,0,\,{}^1/_5,\,{}^1/_2,\,{}^4/_5,1,1,1\}$$

Compute $\mathbf{S}({}^1/_5,{}^3/_5)$. Then ${}^1/_5 \in [u_2,u_3)$ and ${}^3/_5 \in [v_4,v_5)$, and

$$\mathbf{S}\left(\frac{1}{5},\frac{3}{5}\right) = \left[\,N_{0,2}\left(\frac{1}{5}\right)\quad N_{1,2}\left(\frac{1}{5}\right)\quad N_{2,2}\left(\frac{1}{5}\right)\right] \times$$
$$\begin{bmatrix} \mathbf{P}_{0,2} & \mathbf{P}_{0,3} & \mathbf{P}_{0,4} \\ \mathbf{P}_{1,2} & \mathbf{P}_{1,3} & \mathbf{P}_{1,4} \\ \mathbf{P}_{2,2} & \mathbf{P}_{2,3} & \mathbf{P}_{2,4} \end{bmatrix} \begin{bmatrix} N_{2,2}\left(\frac{3}{5}\right) \\ N_{3,2}\left(\frac{3}{5}\right) \\ N_{4,2}\left(\frac{3}{5}\right) \end{bmatrix}$$

Algorithm A3.5 computes the point on a B-spline surface at fixed (u,v) values. For efficiency, it uses a local array, `temp[]`, to store the vector/matrix product, $[\,N_{k,p}(u)\,]^T\,[\,\mathbf{P}_{k,l}\,]$. The resulting vector of points (in `temp[]`) is then multiplied with the vector $[\,N_{l,q}(v)\,]$.

```
ALGORITHM A3.5
  SurfacePoint(n,p,U,m,q,V,P,u,v,S)
    { /*  Compute surface point  */
      /*  Input:  n,p,U,m,q,V,P,u,v  */
      /*  Output: S  */
    uspan = FindSpan(n,p,u,U);
    BasisFuns(uspan,u,p,U,Nu);
    vspan = FindSpan(m,q,v,V);
    BasisFuns(vspan,v,q,V,Nv);
    uind = uspan-p;
    S = 0.0;
    for (l=0; l<=q; l++)
      {
      temp = 0.0;
      vind = vspan-q+l;
      for (k=0; k<=p; k++)
        temp = temp + Nu[k]*P[uind+k][vind];
      S = S + Nv[l]*temp;
      }
    }
```



(a)

Figure 3.20. A B-spline surface. (a) The control net; (b) the surface.

The properties of the tensor product basis functions follow from the corresponding properties of the univariate basis functions listed in Chapter 2.

P3.12 Nonnegativity: $N_{i,p}(u)N_{j,q}(v) \geq 0$ for all i, j, p, q, u, v;

P3.13 Partition of unity: $\sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u)N_{j,q}(v) = 1$ for all $(u,v) \in [0,1] \times [0,1]$;

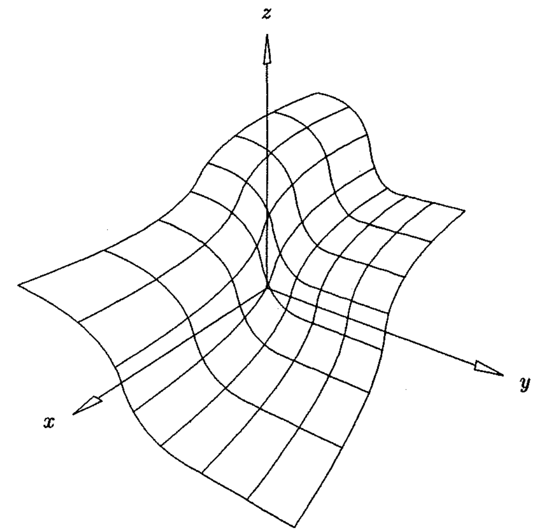
P3.14 If $n = p$, $m = q$, $U = \{0,\ldots,0,1,\ldots,1\}$, and $V = \{0,\ldots,0,1,\ldots,1\}$, then $N_{i,p}(u)N_{j,q}(v) = B_{i,n}(u)B_{j,m}(v)$ for all i, j; that is, products of B-spline functions degenerate to products of Bernstein polynomials;

P3.15 $N_{i,p}(u)N_{j,q}(v) = 0$ if (u,v) is outside the rectangle $[u_i, u_{i+p+1}) \times [v_j, v_{j+q+1})$ (see Figures 3.19a and 3.19b);

P3.16 In any given rectangle, $[u_{i_0}, u_{i_0+1}) \times [v_{j_0}, v_{j_0+1})$, at most $(p+1)(q+1)$ basis functions are nonzero, in particular the $N_{i,p}(u)N_{j,q}(v)$ for $i_0 - p \leq i \leq i_0$ and $j_0 - q \leq j \leq j_0$;

P3.17 If $p > 0$ and $q > 0$, then $N_{i,p}(u)N_{j,q}(v)$ attains exactly one maximum value (see Figures 3.19a and 3.19b);

P3.18 Interior to the rectangles formed by the u and v knot lines, where the function is a bivariate polynomial, all partial derivatives of $N_{i,p}(u)$ $N_{j,q}(v)$ exist; at a u knot (v knot) it is $p-k$ ($q-k$) times differentiable in the u (v) direction, where k is the multiplicity of the knot. In Figure 3.19a the first partial derivative of $N_{4,3}(u)N_{4,2}(v)$ with respect to v is discontinuous along the knot line $v = 3/5$ where $N_{4,2}(v)$ has a cusp. The second partial derivative with respect to u is everywhere continuous, because $N_{4,3}(u)$ is $\mathbf{C}^2$ continuous.



(b)

Figure 3.20. (*Continued.*)

B-spline surfaces have the following properties:

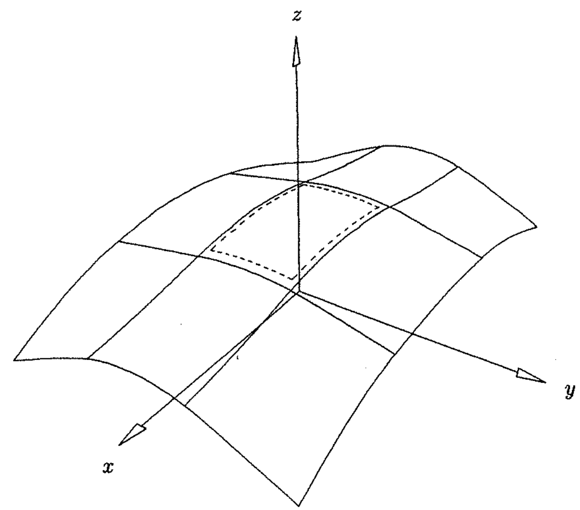
P3.19 If $n = p$, $m = q$, $U = \{0,\ldots,0,1,\ldots,1\}$, and $V = \{0,\ldots,0,1,\ldots,1\}$, then $\mathbf{S}(u,v)$ is a Bézier surface; this follows from P3.14;

P3.20 The surface interpolates the four corner control points: $\mathbf{S}(0,0) = \mathbf{P}_{0,0}$, $\mathbf{S}(1,0) = \mathbf{P}_{n,0}$, $\mathbf{S}(0,1) = \mathbf{P}_{0,m}$, and $\mathbf{S}(1,1) = \mathbf{P}_{n,m}$ (see Figures 3.20 through 3.25); this follows from P3.13 and the identity

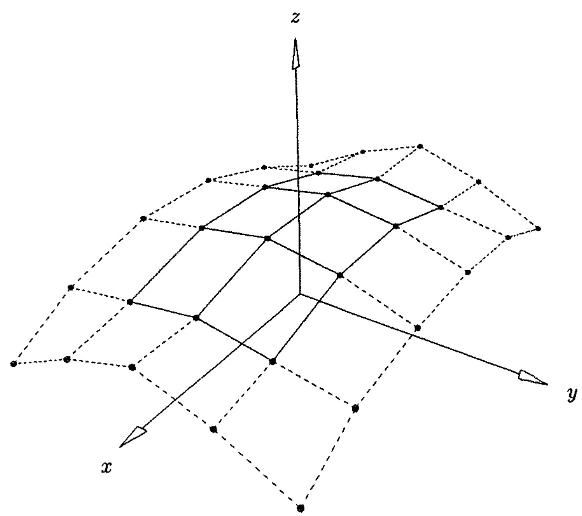
$$N_{0,p}(0)N_{0,q}(0) = N_{n,p}(1)N_{0,q}(0) = N_{0,p}(0)N_{m,q}(1)$$
$$= N_{n,p}(1)N_{m,q}(1) = 1$$

P3.21 Affine invariance: an affine transformation is applied to the surface by applying it to the control points; this follows from P3.13;

P3.22 Strong convex hull property: if $(u,v) \in [u_{i_0}, u_{i_0+1}) \times [v_{j_0}, v_{j_0+1})$, then $\mathbf{S}(u,v)$ is in the convex hull of the control points $\mathbf{P}_{i,j}$, $i_0 - p \leq i \leq i_0$ and $j_0 - q \leq j \leq j_0$ (see Figures 3.21); this follows from P3.12, P3.13, and P3.16;



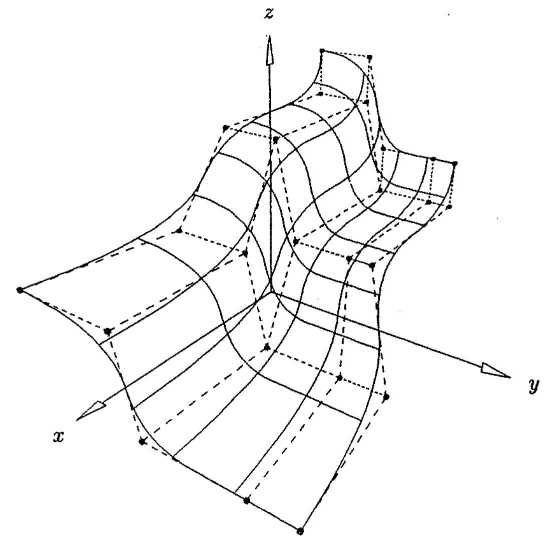
(a)



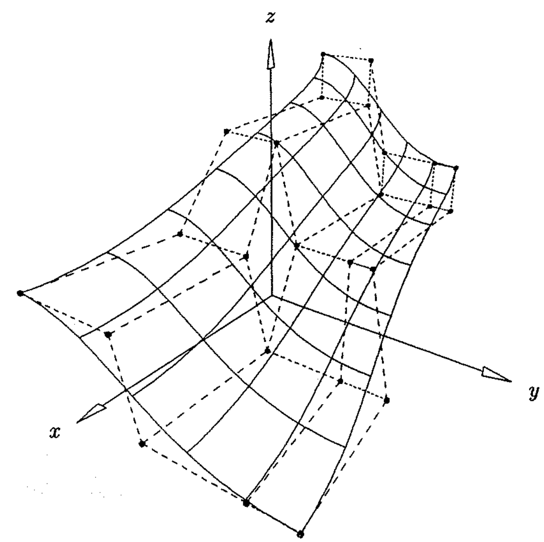
(b)

Figure 3.21. (a) A cubic×quadratic B-spline surface; (b) the strong convex hull property.

P3.23 If triangulated, the control net forms a piecewise planar approximation to the surface; as is the case for curves, the lower the degree the better the approximation (see Figures 3.22a and 3.22b);

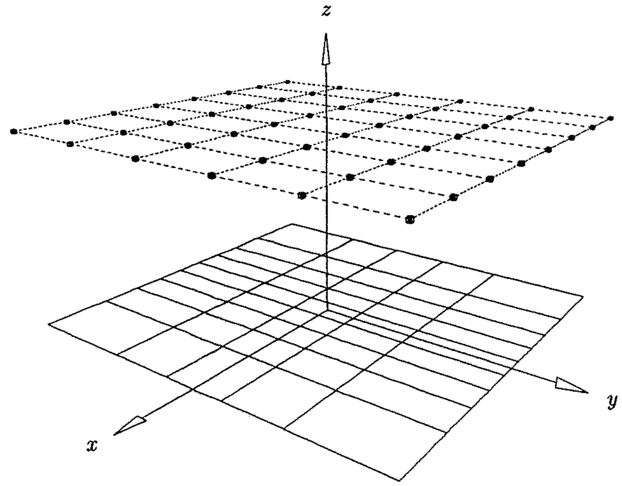


(a)



(b)

Figure 3.22. (a) A biquadratic surface; (b) a biquartic surface ($p = q = 4$) using the same control points as in Figure 3.22a.



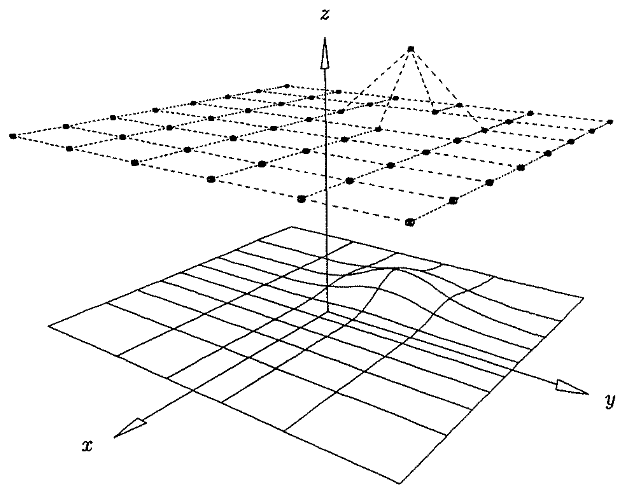


Figure 3.23. (a) A planar quadratic × cubic surface, $U = \{0,0,0,1/4,1/2,3/4,1,1,1\}$ and $V = \{0,0,0,0,1/5,2/5,3/5,4/5,1,1,1,1\}$; (b) $\mathbf{P}_{3,5}$ is moved, affecting surface shape only in the rectangle $[1/4,1) \times [2/5,1)$.

P3.24 Local modification scheme: if $\mathbf{P}_{i,j}$ is moved it affects the surface only in the rectangle $[u_i, u_{i+p+1}) \times [v_j, v_{j+q+1})$; this follows from P3.15. Now consider Figures 3.23a and 3.23b: the initial surface is flat because all the control points lie in a common plane (P3.22); the control net is offset from the surface for better visualization. When $\mathbf{P}_{3,5}$ is moved it affects the surface shape only in the rectangle $[1/4,1) \times [2/5,1)$;

P3.25 The continuity and differentiability of $\mathbf{S}(u,v)$ follows from that of the basis functions. In particular, $\mathbf{S}(u,v)$ is $p-k$ $(q-k)$ times differentiable in the u (v) direction at a u (v) knot of multiplicity k. Figure 3.24 shows a quadratic × cubic surface defined on the knot vectors $U = \{0,0,0,1/2,1/2,1,1,1\}$ and $V = \{0,0,0,0,1/2,1,1,1,1\}$. Notice the crease in the surface, corresponding to the knot line $u = 1/2$. Of course, as is the case for curves, it is possible to position the control points in such a way that they cancel the discontinuities in the basis functions. By using multiply coincident control points, visual discontinuities can be created where there are no corresponding discontinuities in the basis functions; Figure 3.25 shows such a surface, which is bicubic with no multiple knots. Hence, the second partial derivatives are everywhere continuous. The crease is due to the multiple control points.

We remark here that there is no known variation diminishing property for B-spline surfaces (see [Prau92]).

Isoparametric curves on $\mathbf{S}(u,v)$ are obtained in a manner analogous to that for Bézier surfaces. Fix $u = u_0$

$$\mathbf{C}_{u_0}(v) = \mathbf{S}(u_0,v) = \sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u_0)N_{j,q}(v)\,\mathbf{P}_{i,j}$$

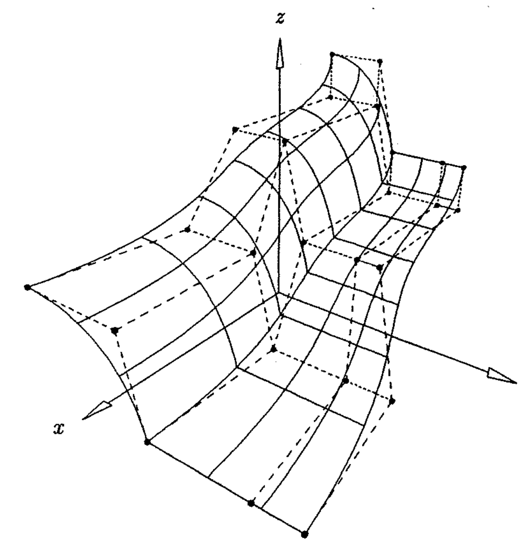


Figure 3.24. A quadratic × cubic surface with crease, $U = \{0,0,0,1/2,1/2,1,1,1\}$ and $V = \{0,0,0,0,1/2,1,1,1,1\}$.

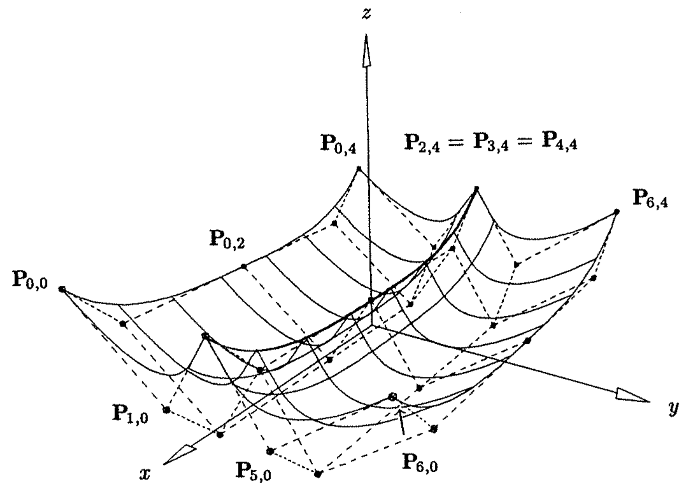


Figure 3.25. A bicubic surface with crease, $U = \{0,0,0,0,1/4,1/2,3/4,1,1,1,1\}$ and $V = \{0,0,0,0,1/2,1,1,1,1\}$; $\mathbf{P}_{2,j} = \mathbf{P}_{3,j} = \mathbf{P}_{4,j}$ for $0 \le j \le 4$.

$$= \sum_{j=0}^{m} N_{j,q}(v) \left(\sum_{i=0}^{n} N_{i,p}(u_0)\mathbf{P}_{i,j} \right) = \sum_{j=0}^{m} N_{j,q}(v)\,\mathbf{Q}_j(u_0) \tag{3.14}$$

where $$\mathbf{Q}_j(u_0) = \sum_{i=0}^{n} N_{i,p}(u_0)\mathbf{P}_{i,j}$$

Analogously $$\mathbf{C}_{v_0}(u) = \sum_{i=0}^{n} N_{i,p}(u)\,\mathbf{Q}_i(v_0)$$

where $$\mathbf{Q}_i(v_0) = \sum_{j=0}^{m} N_{j,q}(v_0)\mathbf{P}_{i,j} \tag{3.15}$$

is a u isocurve on $\mathbf{S}(u,v)$. $\mathbf{C}_{u_0}(v)$ is a qth-degree B-spline curve on V, and $\mathbf{C}_{v_0}(u)$ is a pth-degree B-spline curve on U. The point $\mathbf{S}(u_0,v_0)$ is at the intersection of $\mathbf{C}_{u_0}(v)$ and $\mathbf{C}_{v_0}(u)$. All lines shown on the surfaces of Figures 3.20–3.25 are isolines.

3.5 Derivatives of a B-spline Surface

Let (u,v) be fixed. Generally, one is interested in computing all partial derivatives of $\mathbf{S}(u,v)$ up to and including order d, that is

$$\frac{\partial^{k+l}}{\partial^k u\, \partial^l v}\mathbf{S}(u,v) \qquad 0 \le k+l \le d \tag{3.16}$$

As for curves, we obtain these derivatives by computing derivatives of the basis functions (see Eqs. [2.9] and[2.10], and Algorithm A2.3). In particular

$$\frac{\partial^{k+l}}{\partial^k u\, \partial^l v}\mathbf{S}(u,v) = \sum_{i=0}^{n} \sum_{j=0}^{m} N_{i,p}^{(k)}\, N_{j,q}^{(l)}\,\mathbf{P}_{i,j} \tag{3.17}$$

Algorithm A3.6 computes the point on a B-spline surface and all partial derivatives up to and including order d ($d > p, q$ is allowed). Analogous to Algorithm A3.5, this is a five-step process, with the last step being vector/matrix/vector multiplications of the form

$$\frac{\partial^{k+l}}{\partial^k u\, \partial^l v}\mathbf{S}(u,v) = \left[N_{r,p}^{(k)}(u)\right]^T [\mathbf{P}_{r,s}]\left[N_{s,q}^{(l)}(v)\right]$$
$$0 \le k+l \le d \quad uspan - p \le r \le uspan$$
$$vspan - q \le s \le vspan \tag{3.18}$$

Output is the array SKL[][], where SKL[k][l] is the derivative of $\mathbf{S}(u,v)$ with respect to u k times, and v l times. For fixed k, $0 \le k \le d$, local array temp[] stores the vector/matrix product, $\left[N_{r,p}{}^{(k)}(u)\right]^T [\mathbf{P}_{r,s}]$, while it is being multiplied with the $\left[N_{s,q}{}^{(l)}(v)\right]$, for $0 \le l \le d-k$. Arrays Nu[][] and Nv[][] are used to store the derivatives of the basis functions.

```
ALGORITHM A3.6
  SurfaceDerivsAlg1(n,p,U,m,q,V,P,u,v,d,SKL)
    { /*  Compute surface derivatives  */
      /*  Input:  n,p,U,m,q,V,P,u,v,d  */
      /*  Output: SKL  */
    du = min(d,p);
    for (k=p+1; k<=d; k++)
      for (l=0; l<=d-k; l++)    SKL[k][l] = 0.0;
    dv = min(d,q);
    for (l=q+1; l<=d; l++)
      for (k=0; k<=d-l; k++)    SKL[k][l] = 0.0;
    uspan = FindSpan(n,p,u,U);
    DersBasisFuns(uspan,u,p,du,U,Nu);
    vspan = FindSpan(m,q,v,V);
    DersBasisFuns(vspan,v,q,dv,V,Nv);
    for (k=0; k<=du; k++)
      {
      for (s=0; s<=q; s++)
        {
        temp[s] = 0.0;
        for (r=0; r<=p; r++)
          temp[s] = temp[s] + Nu[k][r]*P[uspan-p+r][vspan-q+s];
        }
```

```
      dd = min(d-k,dv);
      for (l=0; l<=dd; l++)
        {
        SKL[k][l] = 0.0;
        for (s=0; s<=q; s++)
          SKL[k][l] = SKL[k][l] + Nv[l][s]*temp[s];
        }
      }
    }
```

Figure 3.26 shows a bicubic surface and its first and second partial derivatives. Note that the derivatives are scaled down by 1/2 for better visualization.

Let us formally differentiate $\mathbf{S}(u,v)$. With respect to u we have

$$\mathbf{S}_u(u,v) = \frac{\partial}{\partial u}\mathbf{S}(u,v) = \sum_{j=0}^{m} N_{j,q}(v)\left(\frac{\partial}{\partial u}\sum_{i=0}^{n} N_{i,p}(u)\mathbf{P}_{i,j}\right) \tag{3.19}$$

$$= \sum_{j=0}^{m} N_{j,q}(v)\left(\frac{\partial}{\partial u}\mathbf{C}_j(u)\right)$$

where $\qquad \mathbf{C}_j(u) = \sum_{i=0}^{n} N_{i,p}(u)\mathbf{P}_{i,j} \qquad j = 0,\ldots,m$

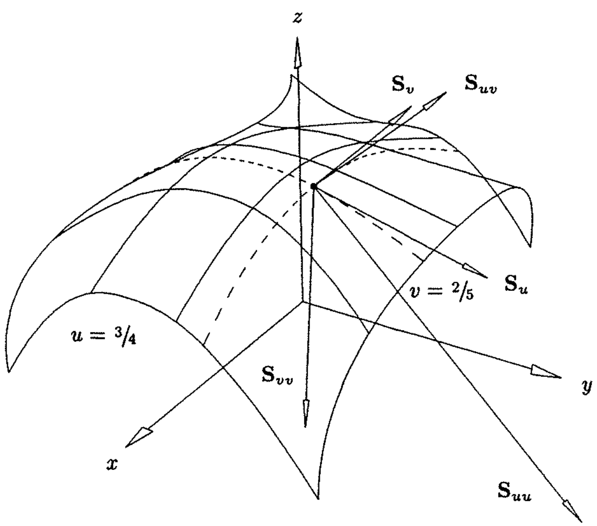


Figure 3.26. A bicubic surface defined on $U = V = \{0,0,0,0,1/2,1,1,1,1\}$ and its first and second partial derivatives computed at $u = 3/4$ and $v = 2/5$.

are B-spline curves. Applying Eq. (3.6) to each of the $\mathbf{C}_j(u)$ and substituting into Eq. (3.19), we obtain

$$\mathbf{S}_u(u,v) = \sum_{i=0}^{n-1}\sum_{j=0}^{m} N_{i,p-1}(u)N_{j,q}(v)\mathbf{P}_{i,j}^{(1,0)} \tag{3.20}$$

where $\qquad \mathbf{P}_{i,j}^{(1,0)} = p\,\dfrac{\mathbf{P}_{i+1,j} - \mathbf{P}_{i,j}}{u_{i+p+1} - u_{i+1}} \qquad$ (see Eq. [3.4])

$$U^{(1)} = \{\underbrace{0,\ldots,0}_{p}, u_{p+1},\ldots,u_{r-p-1},\underbrace{1,\ldots,1}_{p}\}$$

$$V^{(0)} = V$$

Analogously $\qquad \mathbf{S}_v(u,v) = \displaystyle\sum_{i=0}^{n}\sum_{j=0}^{m-1} N_{i,p}(u)N_{j,q-1}(v)\mathbf{P}_{i,j}^{(0,1)} \tag{3.21}$

where $\qquad \mathbf{P}_{i,j}^{(0,1)} = q\,\dfrac{\mathbf{P}_{i,j+1} - \mathbf{P}_{i,j}}{v_{j+q+1} - v_{j+1}}$

$$U^{(0)} = U$$

$$V^{(1)} = \{\underbrace{0,\ldots,0}_{q}, v_{q+1},\ldots,v_{s-q-1},\underbrace{1,\ldots,1}_{q}\}$$

Applying first Eq. (3.20), then Eq. (3.21) yields

$$\mathbf{S}_{uv}(u,v) = \sum_{i=0}^{n-1}\sum_{j=0}^{m-1} N_{i,p-1}(u)N_{j,q-1}(v)\mathbf{P}_{i,j}^{(1,1)} \tag{3.22}$$

where $\qquad \mathbf{P}_{i,j}^{(1,1)} = q\,\dfrac{\mathbf{P}_{i,j+1}^{(1,0)} - \mathbf{P}_{i,j}^{(1,0)}}{v_{j+q+1} - v_{j+1}}$

and $U^{(1)}$ and $V^{(1)}$ are as defined previously.

In general

$$\frac{\partial^{k+l}}{\partial^k u\,\partial^l v}\mathbf{S}(u,v) = \sum_{i=0}^{n-k}\sum_{j=0}^{m-l} N_{i,p-k}(u)N_{j,q-l}(v)\mathbf{P}_{i,j}^{(k,l)} \tag{3.23}$$

where $\qquad \mathbf{P}_{i,j}^{(k,l)} = (q-l+1)\,\dfrac{\mathbf{P}_{i,j+1}^{(k,l-1)} - \mathbf{P}_{i,j}^{(k,l-1)}}{v_{j+q+1} - v_{j+l}}$

Using Eqs. (3.20)–(3.23), we derive useful formulas for corner derivatives. For example, at the corner $(u,v)=(0,0)$, we have

$$
\begin{aligned}
\mathbf{S}_u(0,0) &= \mathbf{P}_{0,0}^{(1,0)} = \frac{p}{u_{p+1}}(\mathbf{P}_{1,0}-\mathbf{P}_{0,0}) \\
\mathbf{S}_v(0,0) &= \mathbf{P}_{0,0}^{(0,1)} = \frac{q}{v_{q+1}}(\mathbf{P}_{0,1}-\mathbf{P}_{0,0}) \\
\mathbf{S}_{uv}(0,0) &= \mathbf{P}_{0,0}^{(1,1)} = \frac{q}{v_{q+1}}\left(\mathbf{P}_{0,1}^{(1,0)}-\mathbf{P}_{0,0}^{(1,0)}\right) \\
&= \frac{pq}{u_{p+1}v_{q+1}}(\mathbf{P}_{1,1}-\mathbf{P}_{0,1}-\mathbf{P}_{1,0}+\mathbf{P}_{0,0})
\end{aligned}
\tag{3.24}
$$

Now let $u_0=0$ and $v_0=0$. From the properties of the basis functions, it is easy to see that the isocurves $\mathbf{C}_{u_0}(v)$ and $\mathbf{C}_{v_0}(u)$ are given by

$$
\mathbf{C}_{u_0}(v)=\sum_{j=0}^{m}N_{j,q}(v)\mathbf{P}_{0,j} \qquad \mathbf{C}_{v_0}(u)=\sum_{i=0}^{n}N_{i,p}(u)\mathbf{P}_{i,0}
$$

From Eq. (3.7) it follows that

$$
\mathbf{S}_u(0,0)=\mathbf{C}'_{v_0}(0) \qquad \mathbf{S}_v(0,0)=\mathbf{C}'_{u_0}(0)
$$

Algorithm A3.7 computes all (or optionally some) of the control points, $\mathbf{P}_{i,j}^{(k,l)}$, of the derivative surfaces up to order d $(0\le k+l\le d)$. The algorithm is based on Eq. (3.23) and Algorithm A3.3. Output is the array, PKL[][][][], where PKL[k][l][i][j] is the i,jth control point of the surface, differentiated k times with respect to u and l times with respect to v.

```
ALGORITHM A3.7
  SurfaceDerivCpts(n,p,U,m,q,V,P,d,r1,r2,s1,s2,PKL)
    { /* Compute control points of derivative surfaces */
      /* Input: n,p,U,m,q,V,P,d,r1,r2,s1,s2 */
      /* Output: PKL */
    du = min(d,p);    dv = min(d,q);
    r = r2-r1;    s = s2-s1;
    for (j=s1; j<=s2; j++)
      {
      CurveDerivCpts(n,p,U,&P[][j],du,r1,r2,temp);
      for (k=0; k<=du; k++)
        for (i=0; i<=r-k; i++)
          PKL[k][0][i][j-s1] = temp[k][i];
      }
    for (k=0; k<du; k++)
      for (i=0; i<=r-k; i++)
        {
        dd = min(d-k,dv);
        CurveDerivCpts(m,q,&V[s1],&PKL[k][0][i][],dd,0,s,temp);
        for (l=1; l<=dd; l++)
          for (j=0; j<=s-l; j++)
            PKL[k][l][i][j] = temp[l][j];
        }
    }
```


Algorithm A3.8 computes the point on a B-spline surface and all partial derivatives up to and including order d, at fixed parameters (u,v) (compare with Algorithm A3.6). $d>p,q$ is allowed. On output, SKL[k][l] is the derivative of $\mathbf{S}(u,v)$ k times with respect to u and l times with respect to v.

```
ALGORITHM A3.8:
  SurfaceDerivsAlg2(n,p,U,m,q,V,P,u,v,d,SKL)
    { /* Compute surface derivatives */
      /* Input: n,p,U,m,q,V,P,u,v,d */
      /* Output: SKL */
    du = min(d,p);
    for (k=p+1; k<=d; k++)
      for (l=0; l<=d-k; l++)   SKL[k][l] = 0.0;
    dv = min(d,q);
    for (l=q+1; l<=d; l++)
      for (k=0; k<=d-l; k++)   SKL[k][l] = 0.0;
    uspan = FindSpan(n,p,u,U);
    AllBasisFuns(uspan,u,p,U,Nu);
    vspan = FindSpan(m,q,v,V);
    AllBasisFuns(vspan,v,q,V,Nv);
    SurfaceDerivCpts(n,p,U,m,q,V,P,d,uspan-p,uspan,
                                        vspan-q,vspan,PKL);
    for (k=0; k<=du; k++)
      {
      dd = min(d-k,dv);
      for (l=0; l<=dd; l++)
        {
        SKL[k][l] = 0.0;
        for (i=0; i<=q-l; i++)
          {
          tmp = 0.0;
          for (j=0; j<=p-k; j++)
            tmp = tmp + Nu[j][p-k]*PKL[k][l][j][i];
          SKL[k][l] = SKL[k][l] + Nv[i][q-l]*tmp;
          }
        }
      }
    }
```

EXERCISES

3.1. Why do quadratic curves touch their control polygons at knots?

3.2. If a quadratic curve has an inflection point, it must be at a knot (see the figures). Why?

3.3. Construct a C^2 continuous cubic curve with a cusp.

3.4. Let a cubic curve be defined by $\mathbf{C}(u) = \sum_{i=0}^{7} N_{i,3}(u)\mathbf{P}_i$ and the knot vector $U = \{0,0,0,0,1/4,1/4,2/3,3/4,1,1,1,1\}$.

a. Assume some arbitrary locations for the $\mathbf{P}_i$ and sketch the curve.

b. Where is the point $\mathbf{C}(1/4)$?

c. If $\mathbf{P}_2$ is moved, on what subinterval of $[0,1]$ is $\mathbf{C}(u)$ affected? If $\mathbf{P}_5$ is moved, what subinterval is affected?

d. Which control points are affecting curve shape on the interval $u \in [1/4, 2/3)$? On the interval $u \in [2/3, 3/4)$?

3.5. Let $\mathbf{C}(u) = \sum_{i=0}^{3} N_{i,2}(u)\mathbf{P}_i$, where $U = \{0,0,0,1/2,1,1,1\}$ and $\mathbf{P}_0 = (-1,0)$, $\mathbf{P}_1 = (-1,1)$, $\mathbf{P}_2 = (1,1)$, and $\mathbf{P}_3 = (1,0)$. Sketch $\mathbf{C}(u)$. Compute $\mathbf{C}'(u)$, i.e., its control points and knot vector. Sketch $\mathbf{C}'(u)$.

3.6. For the cubic curve of Figure 3.16a, assume the control points $\{\mathbf{P}_i\} = \{(0,0),(1,2),(3,4),(5,2),(5,0),(8,0),(9,3)\}$. Sketch the curve. Using Eq. (3.8), compute the second derivative curve, $\mathbf{C}^{(2)}(u)$. Sketch $\mathbf{C}^{(2)}(u)$. Let $\mathbf{P}_0^{(2)}$ and $\mathbf{P}_4^{(2)}$ be the first and last control points of $\mathbf{C}^{(2)}(u)$, i.e., $\mathbf{P}_0^{(2)} = \mathbf{C}^{(2)}(0)$ and $\mathbf{P}_4^{(2)} = \mathbf{C}^{(2)}(1)$. Compute $\mathbf{C}^{(2)}(0)$ and $\mathbf{C}^{(2)}(1)$ using Eqs. (3.9) and (3.10).

3.7. A crease in a surface can be created using either multiple knots or multiple control points (see Figures 3.24 and 3.25). Which would you use if you wanted a crease running less than the full length of the surface? Construct and sketch such an example.

3.8. Consider the B-spline surface $\mathbf{S}(u,v) = \sum_{i=0}^{3}\sum_{j=0}^{2} N_{i,2}(u)N_{j,2}(v)\mathbf{P}_{i,j}$

where
$$U = \{0,0,0,1/2,1,1,1\}$$
$$V = \{0,0,0,1,1,1\}$$

and
$$\mathbf{P}_{0,0} = (0,0,0) \quad \mathbf{P}_{1,0} = (3,0,3) \quad \mathbf{P}_{2,0} = (6,0,3) \quad \mathbf{P}_{3,0} = (9,0,0)$$
$$\mathbf{P}_{0,1} = (0,2,2) \quad \mathbf{P}_{1,1} = (3,2,5) \quad \mathbf{P}_{2,1} = (6,2,5) \quad \mathbf{P}_{3,1} = (9,2,2)$$
$$\mathbf{P}_{0,2} = (0,4,0) \quad \mathbf{P}_{1,2} = (3,4,3) \quad \mathbf{P}_{2,2} = (6,4,3) \quad \mathbf{P}_{3,2} = (9,4,0)$$

Compute $\mathbf{S}(3/10, 6/10)$ by evaluating the nonzero B-spline basis functions and multiplying these by the appropriate control points.

3.9. Derive the expressions for $\mathbf{S}_u(u,v)$, $\mathbf{S}_v(u,v)$, and $\mathbf{S}_{uv}(u,v)$ at the three corners, $(u,v) = (0,1)$, $(1,0)$, and $(1,1)$ (see Eq. [3.24]).

3.10. Let $\mathbf{S}(u,v)$ be as in Exercise 3.8. Sketch this surface. Using Eqs. (3.20) and (3.21), compute the surfaces $\mathbf{S}_u(u,v)$ and $\mathbf{S}_v(u,v)$. Sketch these two surfaces. Using Eq. (3.22) and the expressions derived in Exercise 3.9, compute the mixed partial derivative, $\mathbf{S}_{uv}(u,v)$, at each of the four corners of the surface. What is the geometric significance of these four vectors?