1988

B-spline surface techniques for solid modeling an application to computer-aided geometric design

Chi-Ming Tang

Rochester Institute of Technology
School of Computer Science and Technology

B-Spline Surface Techniques for Solid Modeling
An Application to Computer-Aided Geometric Design

by

Chi - Ming Tang

A thesis submitted to
The Faculty of the School of Computer Science and Technology,
in partial fulfillment of the requirements for the degree of
Master of Science in Computer Science

Approved by: Professor Guy Johnson
Professor Andrew Kitchen
Professor Peter G. Anderson

January 21, 1988

Title of Thesis:

B-Spline Surface Techniques for Solid Modeling
An Application to Computer-Aided Geometric Design

I, Chi - Ming Tang, hereby grant permission to the Wallace Memorial Library, of RIT, to reproduce my thesis in whole or in part. Any reproduction will not be for commercial use or profit.

Date: _____

Abstract

One important area of Computer-Aided Geometric Design (**CAGD**) is concerned with the approximation and representation of the surfaces of solid objects. Accurately describing the shape of an object so that the description is useful to designers who must decide how to manipulate it is an important problem. B-spline techniques promise greater versatility in describing complex surfaces than other techniques, thus the B-spline surface is highlighted in the field of constructive solid geometric modeling. A method for drawing complex surfaces by using B-spline techniques is presented. The tensor product surface scheme is developed for constructing sculptured surfaces. Also, the basic principle of multivariate B-splines, i.e., nontensor product surfaces, the light of tomorrow in **CAGD**, is introduced.

Acknowledgments

The author wishes to express his deepest appreciation to his thesis advisor Professor Guy Johnson for his suggestion of the original topic, his valuable knowledge, guidance, and support throughout the period of this thesis work. Many thanks go to Professors Andrew Kichen and Peter Anderson for their most knowledgeable and fruitful suggestions and criticisms and kind correction of the manuscript.

Also, I would like to take this opportunity to thank Professor Donald Trasher, Chairman of the Department of Mathematics, SUNY Geneseo, for supporting my pursuit this research to expand the parameters of my career.

Most of all I would like to thank my wife Jasmine and my children Angeline, Audwin, and Austin for their support and understanding during the past few years.

CONTENTS

List of Figures

Chapter 1 Introduction and Background

1.1 Introduction

One important area of Computer Aided Geometric Design (**CAGD**) is the approximation and representation of the surfaces of solid objects.

Accurately describing the shape of an object so that the description is useful to designers who must decide how to manufacture it is an old problem. Describing complex shapes is particularly problematic, but computer methods are becoming increasingly useful for defining and communicating complex surface shapes.

Mortenson [1985] classified six major categories of methods of constructing solid models: instances or/and parametrized shapes, cell decomposition (including spatial occupancy enumeration), sweep representations, wireframe representations, constructive solid geometry (**CSG**), and boundary representations.

Of these six, the last two : **CSG** and boundary representations are the most important methods for constructive solid geometric modeling. These two methods are prominent today and will continue to be so in the furture.

The **CSG** representation is a modeling method that defines complex solids as compositions of simple solids (primitives). Examples from PADL-2 language include simple solids such as blocks, cylinders, spheres, wedges, cones and tori. 'PADL' is a

language that has come to designate a family of languages and geometric solid modelling systems developed by the Production Automation Project at the University of Rochester. A binary tree of sets or Boolean operations are used to represent the composition. The following two examples were shown in Mortenson [1985]. Figure 1.1 shows an example of the binary tree for D = (A **U** B) - C.

Figure 1.2 is another example of a Boolean model of a solid object. Let II denote a primitive object. Here, II_1 is a rectangular parallelpiped and II_2 is a right circular cylinder. T_i denotes a transformation that scales and positions the primitives.

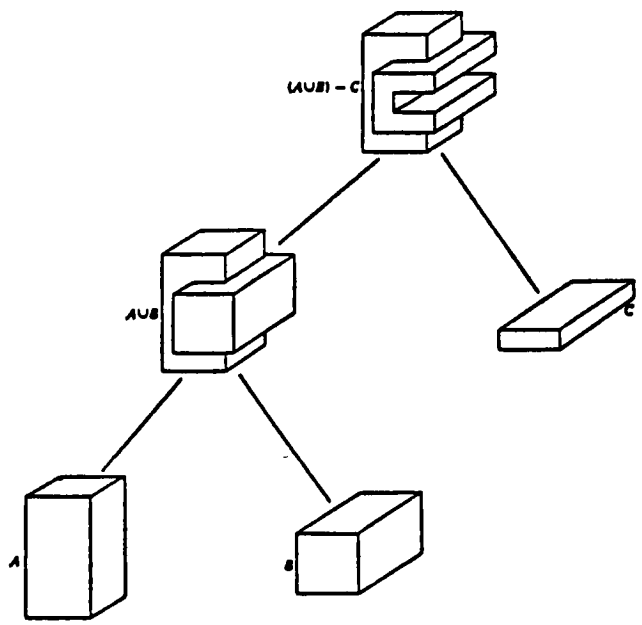


Figure 1.1 The binary tree for D = (A **U** B) - C

(see Mortenson [1985], page 440)

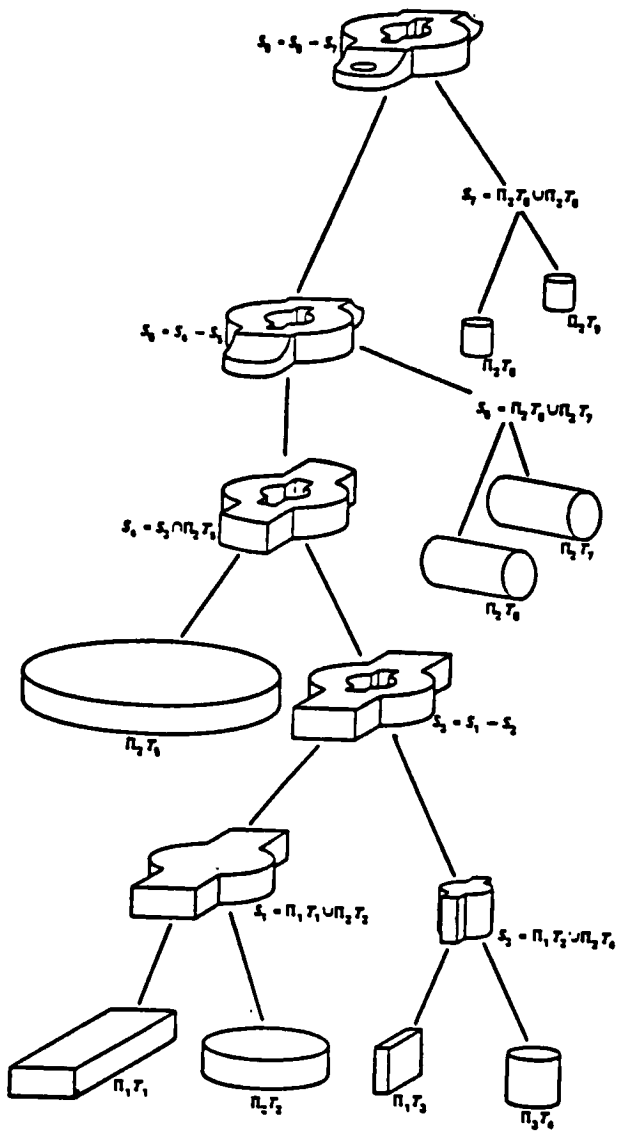


Figure 1.2 A Boolean Model of a Mechanical Part.

(see Mortenson [1985], page 411)

Certain complex shapes, however, particularly those with sculptured surfaces, cannot be represented adequately using CSG methods. The term sculptured is used to refer to parts' surfaces

that have irregular shapes and are not composed of simple primitive shapes. For examples, the figures 1.4, 1.5, and 1.6 are demonstrated the different cases of the sculptured surfaces.

The boundary representation method is used to model objects by defining a closed surface skin around an object that separates the inside from the outside (see figure 1.3). Difficulties arise in using boundary representation methods for defining sculptured surfaces, in particular, when pieces of defined surfaces must be joined to represent the entire surface accurately (see figure 5.1). Continuity may be lacking at the juncture of the two surfaces.

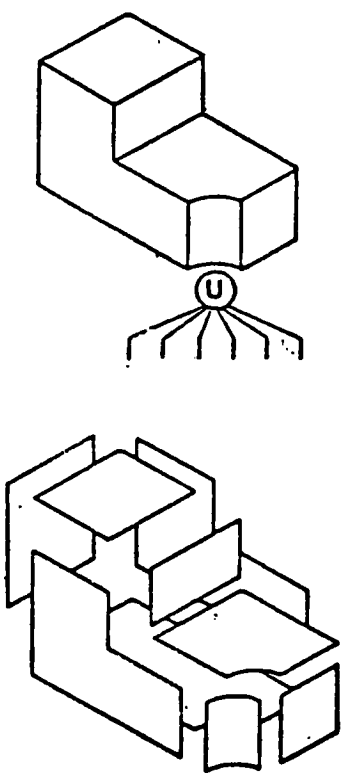


Figure 1.3 Boundary Representation of a Simple Part.

(see Casale and Stanton[1985], page 48)

Techniques for describing sculptured surfaces from which to create models have been explored for the past two decades. Although a large percentage of manufactured parts and components can usually be described with the methods named above, however, parts requiring complex geometry that meet functional constraints are not easily represented with commercially available systems. In other words, there is still a gap between the descriptive power of the geometric modeling scheme and the geometric properties of certain objects. Examples of objects that required specialized modeling techniquires include automobile bodies, internal aircraft components, and engine parts that are designed to maintain their correct shape under extreme operating temperatures and work loads. Some objects that have sculptured surfaces are airfoils, cast machine pieces, molded ornamental trim parts, equipment housings, and car bodies.

Sculptured surfaces can be characterized by their occurrence in one of three general categories of shapes (Knapp [1984]) :

(1) Surfaces that are easily and simply described but are not as simple to represent, such as fillets or rounded edges that are adjacent to cylindrical or elliptical surfaces. (See Figure 1.4)

(2) Blend surfaces that provide a transition between two or more other functional surfaces. Blend surface shapes usually are not critical; it is the surface shapes they transition between that are critical (e.g., a blend is the surface of a product package that smoothly changes from a round screw cap to a square or

octagonal body section, as shown in Figure 1.5).

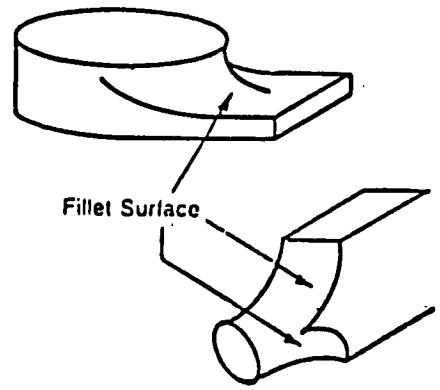


Figure 1.4 Fillet Surfaces between Circular and Rectangular Surfaces (see Knapp [1984], page 4)

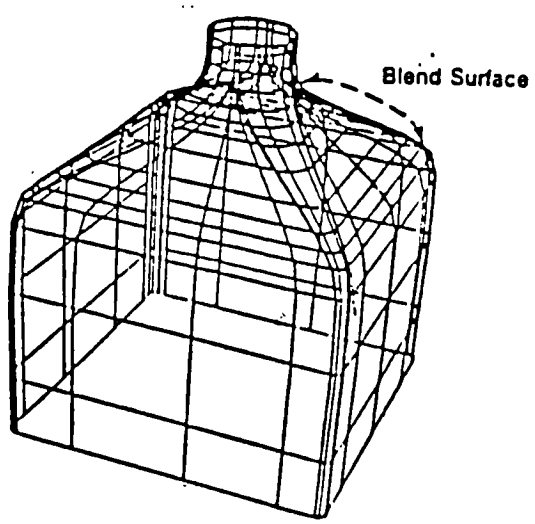


Figure 1.5 Blend Surface Transitioning from Round Top to Square Body (see Knapp [1984], page 5)

(3) Surfaces that are designed strictly according to functional criteria such as fluid or airflow or (e.g., the airfoil of the turbine

blade shown in Figure 1.6 is designed specifically to function with aerodynamic properties) or aesthetic criteria such as styling considerations.

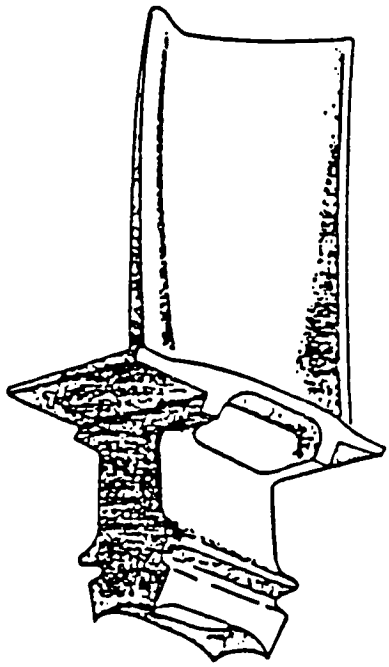


Figure 1.6 Designed Surface: Turbine Airfoil

(see Knapp [1984], page 5)

1.2 The Objective

The major subject this thesis addresses is developing the drawing of complex surfaces by using B-spline techniques. The subject involves the design and implementation of sculptured surfaces by using B-spline methods. The B-spline techniques promise greater versatility in describing complex surfaces than other techniques.

In order to paraphrase the aspect of the central problem, we first need to review the B-Spline approximation, and its (basic) application to B-Spline curves. Then, using the techniques of the B-spline curves, we develope the B-spline surfaces. The method we use here is called the tensor-product surface technique, a traditional approach to computer aided design. This technique can produce the most complex surfaces and is not difficult to understand or implement. However, there is a limitation in this method. For example, in figure 5.1 which depicts the wing-body fairing of an airplane. The region where the leading edge of the wing joins the fuselage cannot be satisfactorily modelled unless some special construction is attempted there such as coalescing knots. This attempt will add the complexity of the constrction. A possible way to overcome this problem is to use the multivariate splines.

Since the theory of multivariate splines was developed, researchers in computer-aided geometric design have hoped to obtain from that theory new and useful tools for the

representation and handling of sculptured surfaces. Therefore, we will introduce the basic principle of multivariate B-Splines, i.e., nontensor product surfaces, the light of tomorrow in **CAGD**.

Chapter 2 B-Spline Approximation

The purpose of this chapter is to describe the theory of **B splines**. The modern mathematical theory of spline approximation was introduced by I. J. Schoenberg in 1947 (Curry and Schoenberg [1966]). In that paper he developed splines for using in a new approach to statistical data smoothing. By the early 1960's, General Motors and United Aircraft had become known as institutional proponents of spline techniques. The first applications of splines in computer-aided design were for interpolation and approximation of existing drawings; that is, copying as opposed to design. In contrast, R. F. Riesenfeld [1973] developed a Bezier-type facility for controlling splines in an ab initio design situation. However, the same interactive techniques can be adapted to the fitting and smoothing of pre-existing data.

A spline function is a function consisting of polynomial pieces joined together with certain smoothness conditions. Thus, the general definition of a **spline** S of **degree** k (**order** k+1), (k > 0), on [a, b] can be defined as following:

(i) The domain of S is an interval [a, b].

(ii) S, S', S'', ..., $S^{(k-1)}$ are all continuous functions on [a, b].

(iii) There are points t_i (called the **knots** of S) such that

$$a = t_1 < t_2 < \cdots < t_n = b$$

and such that S is a polynomial of degree $\leq$ k on each

subinterval $[t_i, t_{i+1}]$.

2.1 B Splines

The B spline was originally introduced by Curry and Schoenberg in 1947. The B splines were so named because they formed a "basis" for the set of all splines. That is, every spline of degree k is a linear combination of the B-splines of degree k based on the same knot sequence. There are several ways to define the B-spline basis mathematically. Originally Schoenberg gave a divided difference formulation. However, for our purpose it is most convenient to adopt the recursive definition of de Boor [1978].

The B splines can be generally defined on a nondecreasing sequence of the knots, which may be finite, infinite or biinfinite. However, at the beginning, we define a B spline without multiple knots. In other words, we suppose that an infinite sequence of knots $\{ t_i \}$ has been prescribed in such a way that

(2.1) $$\cdots < t_{-2} < t_{-1} < t_0 < t_1 < t_2 < \cdots \quad \text{and}$$

$$\lim t_i = \infty = - \lim t_{-i}$$

The **B splines of degree 0** are defined by

(2.2) $$B_{i,0}(x) = \begin{cases} 1 & \text{if } t_i \le x < t_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

If the **support** of a function f(x) is defined as the set of points x where $f(x) \neq 0$, then we can say that the support of $B_{i,0}$ is the half-open interval $[t_i, t_{i+1})$. With the functions $B_{i,0}$ as a starting point, we now generate all the higher degree B **splines** by a simple recursive definition:

$$B_{i,k}(x) = \left(\frac{x - t_i}{t_{i+k} - t_i}\right) B_{i,k-1}(x) + \left(\frac{t_{i+k+1} - x}{t_{i+k+1} - t_{i+1}}\right) B_{i+1,k-1}(x) \qquad k \geq 1 \tag{2.3}$$

Here k = 1,2, ... and i = 0, $\pm 1, \pm 2, \ldots$.

To illustrate equation (2.3), let us determine the **linear B-spline**, i.e., the B-spline of degree 1, $B_{i,1}$ in an alternative form:

$$B_{i,1}(x) = \left(\frac{x - t_i}{t_{i+1} - t_i}\right) B_{i,0}(x) + \left(\frac{t_{i+2} - x}{t_{i+2} - t_{i+1}}\right) B_{i+1,0}(x) \tag{2.4}$$

or

$$B_{i,1}(x) = \begin{cases} \dfrac{x - t_i}{t_{i+1} - t_i} & \text{if } t_i \leq x < t_{i+1} \\ \dfrac{t_{i+2} - x}{t_{i+2} - t_{i+1}} & \text{if } t_{i+1} \leq x < t_{i+2} \\ 0 & \text{if } x \geq t_{i+2} \text{ or } x \leq t_i \end{cases}$$

Note that the support of $B_{i,1}(x)$ is the open interval (t_i, t_{i+2}).

Similarly, recurring the formula (2.3), we will obtain the **parabolic B-spline**, or the B spline of degree 2, $B_{i,2}$, as following:

$$(2.5) \qquad B_{i,2}(x) = \left(\frac{x - t_i}{t_{i+2} - t_i}\right) B_{i,1}(x) + \left(\frac{t_{i+3} - x}{t_{i+3} - t_{i+1}}\right) B_{i+1,1}(x)$$

or

$$B_{i,2}(x) = \begin{cases} \dfrac{(x - t_i)^2}{(t_{i+2} - t_i)(t_{i+1} - t_i)} & \text{if } t_i \le x \le t_{i+1} \\ \left(\dfrac{(x - t_i)(t_{i+2} - x)}{(t_{i+2} - t_i)(t_{i+2} - t_{i+1})}\right) + \left(\dfrac{(t_{i+3} - x)(x - t_{i+1})}{(t_{i+3} - t_{i+1})(t_{i+2} - t_{i+1})}\right) & \text{if } t_{i+1} \le x \le t_{i+2} \\ \dfrac{(t_{i+3} - x)^2}{(t_{i+3} - t_{i+1})(t_{i+3} - t_{i+2})} & \text{if } t_{i+2} \le x \le t_{i+3} \\ 0, \text{ elsewhere.} \end{cases}$$

It is obvious that the support of $B_{i,2}(x)$ is the open interval (t_i, t_{i+3}). In particular, the parabolic B-spline with equispaced knots t_i, t_i+h, t_i+2h, and t_i+3h ($h > 0$) is given explicitly by

$$(2.6)\qquad B_{i,2}(x) = \begin{cases} \frac{1}{2} W^2 & \text{if } t_i \le x < t_i+h \\ -\frac{3}{2} + 3W - W^2 & \text{if } t_i+h \le x < t_i+2h \\ \frac{9}{2} - 3W + \frac{1}{2} W^2 & \text{if } t_i+2h \le x < t_i+3h \\ 0, \text{elsewhere,} \end{cases}$$

where $W = (x - t_i)/h$.

Once more recurrence on the formula (2.3), we have the **cubic B-spline**, i.e., the B spline of degree 3, which is given by

$$(2.7)\qquad B_{i,3}(x) = \left(\frac{x - t_i}{t_{i+3} - t_i} \right) B_{i,2}(x) + \left(\frac{t_{i+4} - x}{t_{i+4} - t_{i+1}} \right) B_{i+1,2}(x)$$

We can trivially see that the support of $B_{i,3}(x)$ is the open interval (t_i, t_{i+4}). Same derivation as the parabolic case, we can show that the cubic B-spline with equispaced knots t_i, t_i+h, t_i+2h, t_i+3h, and t_i+4h ($h > 0$) is given explicitly by

$$(2.8)\qquad B_{i,3}(x) = \begin{cases} \frac{1}{6} W^3 & \text{if } t_i \le x < t_i+h \\ \frac{2}{3} - 2W + 2W^2 - \frac{1}{2} W^3 & \text{if } t_i+h \le x < t_i+2h \\ -\frac{22}{3} + 10W - 4W^2 + \frac{1}{2} W^3 & \text{if } t_i+2h \le x < t_i+3h \\ \frac{32}{3} - 8W + 2W^2 - \frac{1}{6} W^3 & \text{if } t_i+3h \le x < \\ 0, \text{elsewhere} \end{cases}$$

where $W = (x - t_i)/h$.

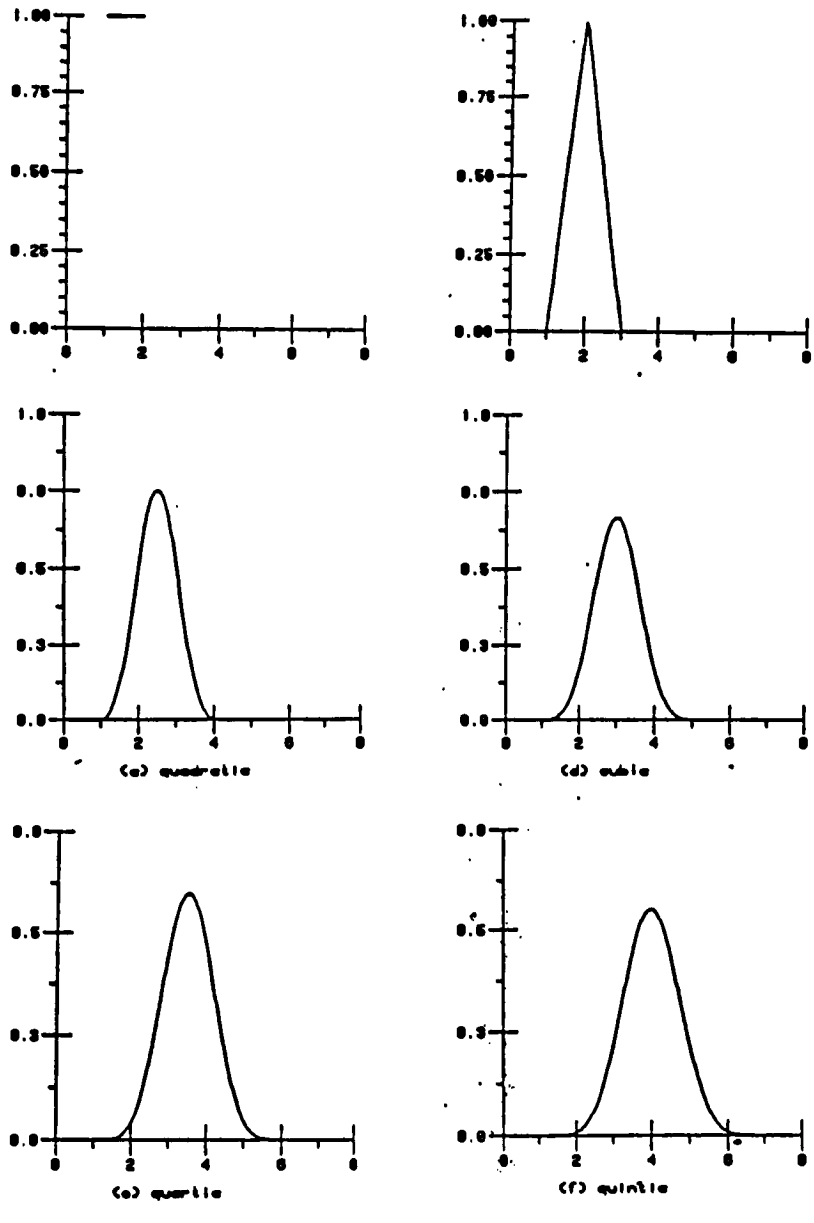


Figure 2.1 (a)-(f) B-spline basis functions

Figure 2.1 Shows plots of B-splines of degree 0, 1, 2, 3, 4, and

5 with equispaced integer knots $t_i = i$ over the interval [0, 8]. That is , $B_{1,k}$ for k = 0, ... , 5. They are continuous functions except the case k = 0. All graphs are symmetric with respect to its maximum point.

From the above discussions, we can list a number of properties of B splines $B_{i,k}(x)$. For each $k \geq 0$,

(2.9) (i) **Positivity:**

$$B_{i,k}(x) \geq 0 \qquad \text{for all } x \text{ and for all } i,$$

(ii) **Partition of Unity:**

$$\sum_{i=-\infty}^{+\infty} B_{i,k}(x) = 1 \qquad \text{for all } x \text{ and for all } k \geq 0,$$

(iii) **Local Support:**

$$B_{i,k}(x) = 0 \qquad \text{if } x < t_i \text{ or } t_{i+k+1} \leq x$$

(iv) **Continuity:**

$B_{i,k}(x)$ is (k-1) times continuously differentiable.

Properties (i) and (iii) are clearly indicated on the figure 2.1. Property (iv) can be observed from the definitions (2.2) and (2.3). Property (ii) is obvious for k = 0 and is not so obvious for k > 0, however, can be shown by using the de Boor algorithm (see equation (2.18) on section 2.3).

2.2 B Splines with Multiple Knots

As we mentioned early, the B-splines can be generally defined on a nondecreasing sequence of the knots. In early history of spline theory, a spline function of degree k was defined to be a piecewise polynomial function of degree k on some (finite or infinite) interval with k-1 continous derivatives there. In other words, "spline function" meant a piecewise polynomial function which was as smooth as it could be without simply reducing to a polynomial. But it was soon found that piecewise polynomial functions of less than this maximum smoothness were also quite interesing and useful; examples will shown in the next chapter. Some people have called such piecewise polynomial functions with less than maximum (nontrival) smoothness **deficient splines**. However, we call such functions **splines with multiple knots** since we can think of them as having been obtained from a splire with simple knots (i.e., one of the original splines) by letting some knots coalesce. Now, we define B splines with multiple knots as follows:

Suppose in the knot sequence $t_i, t_{i+1}, t_{i+2}, \cdots, t_{i+k+1}$ we have $t_j = t_{j+1} = \cdots = t_{j+m}$, $0 < m \leq k$. Then the spline at t_j may have a jump in its (k-m)th derivative. That is a natural interpretation of multiple knots and is illustrated in figure 2.2. This figure shows a B spline of degree 1 with the knots $\{t_{j-i}, t_j, t_{j+1}\}$ evolved to a piecewise linear function. In other words, a discontinous

piecewise linear function is shown as a limit of a linear spline with knots as two knots coalesce ($t_j = t_{j+1}$). The limiting case is a break or gap at t_j, and one degree of smoothness is lost. This means the function is not continous any more. To extend this idea, we can see that, if (m+1) knots $t_j = \cdots = t_{j+m}$ coincide, the B-spline curve may become only C^{k-m-1} continuous at t_j [de Boor '78, p 111 and p 125; Bohm '84; Rice '83, p71].

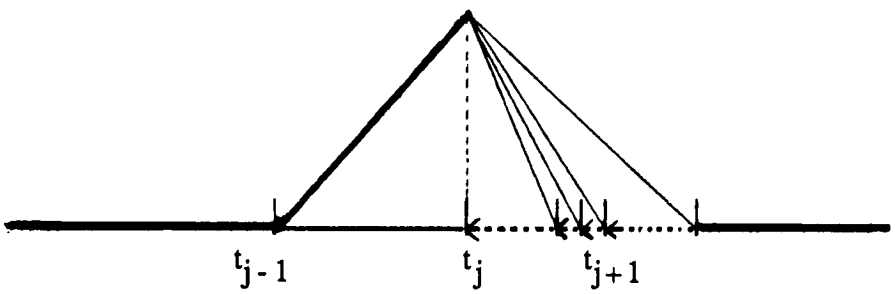


Figure 2.2 Formation of a jump by coalescing two knots.

If we adopt the idea of B splines with multiple knots, we may encounter zero denominators in equation (2.3). For most situations, one can define the limit 0/0 to be 1 to eliminate difficulty. However, to be consistent with the properties (2.9) listed above and the definitions (2.2) and (2.3), we shall apecify that the B-spline of degree 0 with multiple knots, $t_i = t_{i+1}$, is defined by

(2.10) $\quad B_{i,0}(x) = 0 \quad$ for all x.

Using these definitions we can define B-splines with multiple

knots just by repeating the knots in the set of knots. The algorithms using the B-spline representations (2.2) and (2.3) can all be extended in a numerically stable way to handle multiple knots.

Figure 2.3 shows all eight possible combinations of the cubic B-splines, $B_{i,3}$, with multiple knots. In figures 2.3(a) and 2.3(b), the graphs, with the knot sequences $(t_i, t_{i+1}, t_{i+2}, t_{i+3}, t_{i+4}) = (1,1,2,5,7)$ and $(t_i, t_{i+1}, t_{i+2}, t_{i+3}, t_{i+4}) = (1,2,2,5,7)$ appear to have a discontinuous second derivative at 1 and 2, respectively. In figure 2.3(c) , the graph with the knot sequence $(t_i, t_{i+1}, t_{i+2}, t_{i+3}, t_{i+4}) = (1,1,2,2,7)$ exhibits to have a discontinuous second derivative at 1 and 2. In figure 2.3(d) , the graph with the knot sequence $(t_i, t_{i+1}, t_{i+2}, t_{i+3}, t_{i+4}) = (1,1,2,3,3)$ exposes to have a discontinuous second derivative at 1 and 3. In figures 2.3(e) and 2.3(f), the graphs, with the knot sequences $(t_i, t_{i+1}, t_{i+2}, t_{i+3}, t_{i+4}) = (1,1,1,3,7)$ and $(t_i, t_{i+1}, t_{i+2}, t_{i+3}, t_{i+4}) = (1,4,4,4,7)$, display a discontinuous first derivative at 1 and 4, respectively. In figure 2.3(g) , the graph with the knot sequence $(t_i, t_{i+1}, t_{i+2}, t_{i+3}, t_{i+4}) = (1,1,1,7,7)$ demonstrates to have a discontinuous first derivative at 1 and a discoutinuous second derivative at 7. In figure 2.3(h), the graph with the knots $(t_i, t_{i+1}, t_{i+2}, t_{i+3}, t_{i+4}) = (1,1,1,1,7)$ shows a jump discontinuity at 1.

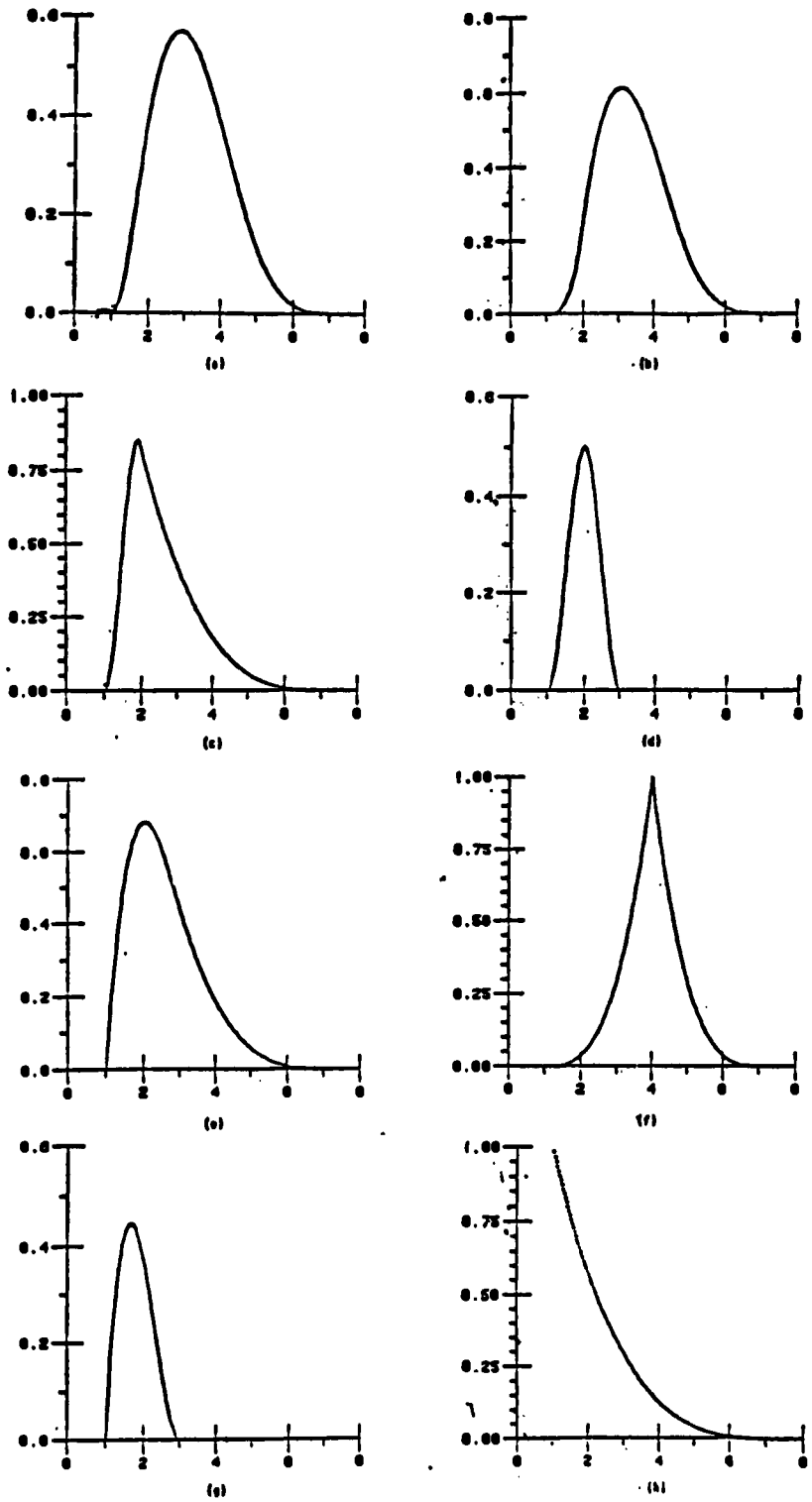


Figure 2.3 (a) - (h) The cubic B-splines with multiple knots

2.3 The de Boor Algorithm

The principal use of the B-splines $B_{i,k}$ ($i = 0, \pm 1, \pm 2, \ldots$) is as a basis for the set of all k th-degree splines having the same knot sequences. Thus, linear combinations

$$(2.11) \qquad \sum_{i=-\infty}^{+\infty} c_{i,k} B_{i,k}(x)$$

are important objects of study. We use the lower case letter $c_{i,k}$ for fixed k and the upper case letter $C_{i,k}$ to emphasize the degree k of the corresponding B splines. Our first task is to develop an efficient method to evaluate a function of the form

$$(2.12) \qquad f(x) = \sum_{i=-\infty}^{+\infty} C_{i,k} B_{i,k}(x)$$

given the coefficients $C_{i,k}$ and the knot sequence t_i. Using definition (2.3) and some simple series manipulations, we have

$$(2.13) \qquad f(x) = \sum_{i=-\infty}^{+\infty} C_{i,k} B_{i,k}(x)$$

$$= \sum_{i=-\infty}^{+\infty} C_{i,k-1} B_{i,k-1}(x)$$

where $C_{i,k-1}$ is determined by

$$(2.14) \qquad C_{i,k-1} = \frac{C_{i,k}\,(x - t_i) + C_{i-1,k}\,(t_{i+k} - x)}{t_{i+k} - t_i}$$

This algebraic manipulation shows how a linear combination of $B_{i,k}(x)$ can be expressed as a linear combination of $B_{i,k-1}(x)$. Repeating this process k-1 times, we eventually express f(x) in the form

$$(2.15) \qquad f(x) = \sum_{i=-\infty}^{+\infty} C_{i,0}\, B_{i,0}(x)$$

Although the coefficients $C_{i,k}$'s in equation (2.12) are not dependent of x, whereas the $C_{i,k-1}$'s calculated subsequently by equation (2.14) do depent on x.

A nice feature of equation (2.12) or (2.15) is that

$$(2.16) \qquad f(x) = C_{m,0} \qquad \text{if } t_m \le x < t_{m+1}.$$

In other words, if f is defined by equation (2.12), then it can be simplified to

$$(2.17) \qquad \sum_{i=-\infty}^{+\infty} C_{i,k}\, B_{i,k}(x)$$

$$= \sum_{i=m-k}^{m} C_{i,k}\, B_{i,k}(x) \qquad \text{if } t_m \le x < t_{m+1}$$

In a word, only the k+1 coefficients $C_{m,k}, C_{m-1,k}, \cdots, C_{m-k,k}$ are

needed to compute f(x) if $t_m \leq x < t_{m+1}$.

Summarily, if f is defined by equation (2.12) and we want to evaluate f(x), we use equation (2.14) to calculate the entries in the following triangular array:

$$
\begin{array}{ccccccc}
C_{m,k} & \rightarrow & C_{m,k-1} & \rightarrow & \cdots & \rightarrow & C_{m,0} \\
 & \nearrow & & \nearrow & & & \\
C_{m-1,k} & \rightarrow & C_{m-1,k-1} & \rightarrow & \cdots & & \\
\vdots & & \vdots & & & & \\
C_{m-k+1,k} & \rightarrow & C_{m-k+1,k-1} & & & & \\
 & \nearrow & & & & & \\
C_{m-k,k} & & & & & &
\end{array}
$$

The above algorithm is equivalent to the so-called **de Boor algorithm**. The de Boor algorithm is a generalization of the de Casteljan algorithm which evaluates a Bernstein polynomials at a given point; i.e., a Bezier curve at a given point. (Bohm, Farin, and Kahmann [1984])

It is now a simple matter back to establish that

$$(2.18) \qquad \sum_{i=-\infty}^{+\infty} B_{i,k}(x) = 1 \qquad \text{for all } x \text{ and all } k \geq 0.$$

It is obvious for k = 0. If k > 0, we use equation (2.12) with $C_{i,k} = 1$ for all i. By equation (2.14), all subsequent coefficients

$C_{i,k}$, $C_{i,k-1}$, $C_{i,k-2}$, ... , $C_{i,0}$ are also equal to 1. Thus at the end equation (2.15) or (2.16) is true with $C_{i,0} = 1$ and so $f(x) = 1$. Therefore, from (2.12) the sum of all B splines of degree k is unity.

2.4 Derivatives

A basic result about the derivatives of B splines is

$$D_x B_{i,k}(x) = \left(\frac{k}{t_{i+k} - t_i}\right) B_{i,k-1}(x) - \left(\frac{k}{t_{i+k+1} - t_{i+1}}\right) B_{i+1,k-1}(x) \tag{2.19}$$

The symbol D_x indicates the operation of taking derivative of whatever follows with respect to the variable x, that is, $D_x = d/dx$. The equation (2.19) can be proved by induction, using the recursive formula (2.3). Thus, the first derivative of a B spline of degree k is a linear combination of two B splines of degree k-1. Once equation (2.19) is established, we get the useful formula

$$D_x \sum_{i=-\infty}^{+\infty} c_i B_{i,k}(x) = \sum_{i=-\infty}^{+\infty} d_i B_{i,k-1}(x) \tag{2.20}$$

where

$$d_i = k\left(\frac{c_i - c_{i-1}}{t_{i+k} - t_i}\right)$$

Equation (2.20) explains that the first derivative of a B-spline curve of degree k is a B-spline curve of degree k-1. The verification is as follows: By equation (2.19),

$$\mathbf{D}_x \sum_{i=-\infty}^{+\infty} c_i B_{i,k}(x)$$

$$= \sum_{i=-\infty}^{+\infty} c_i \mathbf{D}_x B_{i,k}(x)$$

$$= \sum_{i=-\infty}^{+\infty} c_i \left[\left(\frac{k}{t_{i+k} - t_i} \right) B_{i,k-1}(x) - \left(\frac{k}{t_{i+k+1} - t_{i+1}} \right) B_{i+1,k-1}(x) \right]$$

$$= \sum_{i=-\infty}^{+\infty} \left[\frac{c_i k}{t_{i+k} - t_i} - \frac{c_{i-1} k}{t_{i+k} - t_i} \right] B_{i,k-1}(x)$$

$$= \sum_{i=-\infty}^{+\infty} d_i B_{i,k-1}(x)$$

Chapter 3 B-Spline Curves - An Application of B-Spline Approximation

3.1 Parametric Equations

We can use either parametric or nonparametric equations to represent geometric elements mathematically. However, the shapes of most objects we want to model are intrinsically independent of any coordinate system. In fact, if we must fit a curve or surface through a set of points, we are interested in seeing the relationship between the (control) points themselves and their determining resulting shape, not the relationship between these points and some arbitrary coordinate system. In addition, any closed object, solid, will have vertical tangent lines or planes with respect to any choice of coordinate system. This results in infinite slopes or other ill-defined mathematical properties. Thus, the curves and surfaces of geometric modeling may not be represented by an ordinary nonparametric function at all. For these reasons and many others related to ease of programming and computability, the dominant means of representing shapes in geometric modeling is with parametric equations. In other words, the parametric representation is the most adequate description of the curves or surfaces that are drawn by a plotter or on a computer graphics display screen. Therefore, we will use parametric equations to express curves or surfaces because parametric equations have many advantages

over direct or explicit forms. Also, it is convienent for us to normalize the parametric variable, which means limiting its value to the closed interval between 0 and 1, inclusive. This restriction gives rise to curve and surface boundaries, which we will investigate later.

Throughout this thesis, we shall use lowercase boldface letters to denote the points or vectors in 2D or 3D; e.g.,

$$\mathbf{p} = [u, v] \text{ or } \mathbf{p} = [x, y, z]$$

A curve $\mathbf{p}(u)$ can be interpreted as the image of a part of the real u-axis (the domain) by a map into 2D or 3D. A surface $\mathbf{p}(u,v)$ can be interpreted as the image of a region of the real uv-plane (the domain) by a map into 2D or 3D.

As we mentioned above, the dominant form used to model curves and surfaces is the parameteric or vector-valued function. For example, a two-dimensional curve is expressed by a set of two functions $x = x(u)$, $y = y(u)$ of a parameter u. A point on such a curve is represented by the vector

$$\mathbf{p}(u) = [\ x(u),\ y(u)\ ].$$

Similary, a point on a space curve is given by the vector

$$\mathbf{p}(u) = [\ x(u),\ y(u),\ z(u)\ ]$$

and a point on a surface is represented by the vector

$$\mathbf{p}(u,v) = [\ x(u,v),\ y(u,v),\ z(u,v)],$$

where each component of $\mathbf{p}$ is a function of two parameters u and v.

Again, recall that the parameters u and v take on values in a

specified range, usually 0 to 1, the parametric functions x, y, and z trace out the location of the curve or surface. In the subsequence, these parametric functions will serve as the so called blending functions in the geometric form in B-spline curve or B-spline surface.

One of the key principles in **CAGD** is to segment curves and surfaces into smaller pieces and then to treat each piece individually. A segmentation of a curve into several segments corresponds to a partition of its domain by "knots" $u_0 < u_1 < \ldots < u_p$. A segmentation of a surface corresponds to a partition of the domain by "knotlines" $u_0 < u_1 < \ldots < u_p$, and $v_0 < v_1 < \ldots < v_q$.

3.2 B-Spline Curves

A B-spline curve is a piecewise polynomial curve

$$(3.1) \qquad s(u) = \sum_{i=0}^{n} \mathbf{p}_i B_{i,k}(u)$$

The vector coefficents $\mathbf{p}_i$ of the B-spline $B_{i,k}(u)$ are called control points or de Boor points; they form the vertices of its so-called de Boor polygon. Comparing with Bezier curves, the most important difference is the way the blending functions $B_{i,k}(u)$ are formulated. For example, the Bezier curves, the degree of the blending function polynomial is determined by the number of control points. For B-spline curves, the degree of these

polynomials is specially controlled by a parameter k and usually independent of the number of control points, except as limited by (3.4). The B-spline blending functions $B_{i,k}(t)$'s are defined recursively by the expressions (2.2) and (2.3).

Throughtout this chapter: we will generate both open (periodic and nonperiodic) and closed (periodic) curves by appropriately specifying the knots and their corresponding B-spline functions. The knot values relate the parametric variable u to the $\mathbf{p}_i$ control points. Second, we apply these resulting blending functions to any set of (n+1) control points by means of eq. (3.1) and, then, find the expression for the required B-spline curve. We now canvass open curves and canvass closed curves later.

3.2.1 Open Non-periodic Curves

For an **open curve**, the knot values t's are defined as follows:

$$t_i = \begin{cases} 0 & \text{if } i \le k \\ i - k & \text{if } k < i \le n \\ n - k + 1 & \text{if } n < i \end{cases} \tag{3.2}$$

with

$$0 \le i \le n + k + 1. \tag{3.3}$$

The range of the parametric variable u is

(3.4) $0 \leq u \leq n - k + 1.$

Example (A) - Quadratic Curves

For the $B_{i,2}(u)$ blending functions with 6 control points $\mathbf{p}_0$, $\mathbf{p}_1, \ldots, \mathbf{p}_5$, (or n = 5), we find that

$$0 \leq u \leq 4$$

and the knot values

$$(t_0, t_1, t_2, \ldots, t_8) = (0, 0, 0, 1, 2, 3, 4, 4, 4).$$

If we now apply to any set of six control points using eq.(3.1), we find that $\mathbf{p}(u)$ is given by a different equation for each unit interval in u,

(3.5) $\mathbf{p}(u) =$

$$\mathbf{p}_1(u) = (1-u)^2\mathbf{p}_0+0.5u(4-3u)\mathbf{p}_1+0.5u^2\mathbf{p}_2 \quad \text{for } 0\leq u<1$$

$$\mathbf{p}_2(u)=0.5(2-u)^2\mathbf{p}_1+0.5(-2u^2+6u-3)\mathbf{p}_2+0.5(u-1)^2\mathbf{p}_3 \quad \text{for } 1\leq u<2$$

$$\mathbf{p}_3(u)=0.5(3-u)^2\mathbf{p}_2+0.5(-2u^2+10u-11)\mathbf{p}_3+0.5(u-2)^2\mathbf{p}_4 \quad \text{for } 2\leq u<3$$

$$\mathbf{p}_4(u)= 0.5(4-u)^2\mathbf{p}_3+0.5(-3u^2+20u-32)\mathbf{p}_4+(u-3)^2\mathbf{p}_5 \quad \text{for } 3\leq u\leq 4$$

where $\mathbf{p}_i(u)$, (i = 1, ... , 4), is the i $\underline{\text{th}}$ curve segment. We shall note the following results:

(i) By evaluating the eqs. (3.5), we have

$$\mathbf{p}(0) = \mathbf{p}_1(0) = \mathbf{p}_1 \quad \text{and} \quad \mathbf{p}(4) = \mathbf{p}_4(4) = \mathbf{p}_5.$$

and

(ii) By differentiating the eqs. (3.5) before the evaluation, we have

$$\mathbf{p}'(0) = \mathbf{p}_1'(0) = 2(\mathbf{p}_1 - \mathbf{p}_0),$$

$$\mathbf{p}'(1) = \mathbf{p}_1'(1) = \mathbf{p}_2'(1) = (\mathbf{p}_2 - \mathbf{p}_1),$$

$$\mathbf{p}'(2) = \mathbf{p}_2'(2) = \mathbf{p}_3'(2) = (\mathbf{p}_3 - \mathbf{p}_2),$$

$$\mathbf{p}'(3) = \mathbf{p}_3'(3) = \mathbf{p}_4'(3) = (\mathbf{p}_4 - \mathbf{p}_3), \quad \text{and}$$

$$\mathbf{p}'(4) = \mathbf{p}_4'(4) = 2(\mathbf{p}_5 - \mathbf{p}_4).$$

Therefore, the resulting curve is a composite sequence of four curve segments connected with c^1 continuity; an example is shown in figure 3.1. From (i), we note that the curve passes through only the first and last points, $\mathbf{p}_0$ and $\mathbf{p}_5$, and it is tangent to $\mathbf{p}_1 - \mathbf{p}_0$ and $\mathbf{p}_5 - \mathbf{p}_4$ at these same points. From (ii), we can see that this curve is tangent to each successive side of the characteristic polygon (only for quadratic curve). This tangency occurs at the joints between curve segments (that is, at integral values of u). These joints are indicated by tick marks.

We shall note that there are computational advantages to reparametrizing the interval so that $0 \leq u < 1$ and then identifying the interval in same way. Thus, the equations (3.5) can be

reparametrized as follows:

$$
\begin{aligned}
(3.6)\quad \mathbf{p}(u) &= \\
\mathbf{p}_1(u) &= (1-u)^2\mathbf{p}_0+0.5u(4-3u)\mathbf{p}_1+0.5u^2\mathbf{p}_2 && \text{for } 0\le u<1\\
\mathbf{p}_2(u) &= 0.5(1-u)^2\mathbf{p}_1+0.5(-2u^2+2u+1)\mathbf{p}_2+0.5u^2\mathbf{p}_3 && \text{for } 0\le u<1\\
\mathbf{p}_3(u) &= 0.5(1-u)^2\mathbf{p}_2+0.5(-2u^2+2u+1)\mathbf{p}_3+0.5u^2\mathbf{p}_4 && \text{for } 0\le u<1\\
\mathbf{p}_4(u) &= 0.5(1-u)^2\mathbf{p}_3+0.5(-3u^2+2u+1)\mathbf{p}_4+u^2\mathbf{p}_5 && \text{for } 0\le u\le 1
\end{aligned}
$$

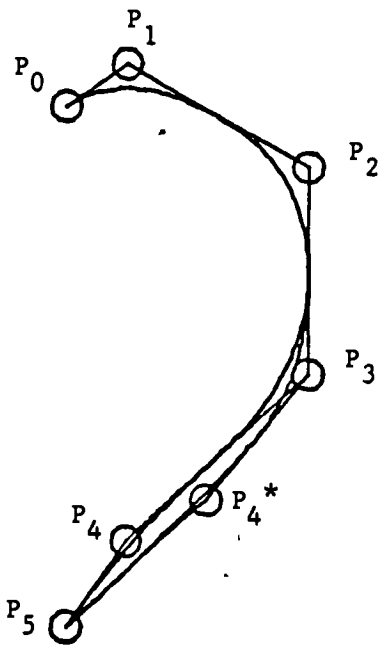


Figure 3.1 Non-periodic quadric B-spline curves

From eqs. (3.5) and (3.6), it is evident that only two and three control points influence each curve segment, respectively. Conversely, a control point can influence the shape of at most two

and three curve segments, respectively. For example, in figure 3.1, the point $\mathbf{p}_1$ is moved to $\mathbf{p}_1$*, and the effect on the curve is plotted. This is visual evidence that a local change affects only two segments of the curve. It is this local control of the curve shape, possible with B-splines, that gives the technique an advantage over other techniques because most curve-defining techniques do not provide for local control of shape. For example, a small (local) change in the position of a point on a spline curve or of a vertex of a characteristic polygon of a Bezier curve, tends to be strongly propagated throughout the entire curve. (This is sometimes described as a global propagation of change.)

Example (B) - Cubic Curves

For the $B_{i,3}(u)$ blending functions with 8 control points $\mathbf{p}_0$, $\mathbf{p}_1, \ldots, \mathbf{p}_7$, (or n = 7), we find that

$$0 \leq u \leq 5$$

and the knot values

$$(t_0, t_1, t_2, \ldots, t_{11}) = (0, 0, 0, 0,1, 2, 3, 4, 5, 5, 5, 5).$$

If we now apply to any set of eight control points using Eq. (3.1) and let a = 1/4, b = 1/12, and c = 1/6, we find that $\mathbf{p}(u)$ is given by a different equation for each unit interval in u,

$$
\begin{aligned}
(3.7)\quad \mathbf{p}(u) &= \\
\mathbf{p}_1(u) &= (1-u)^3\mathbf{p}_0+a(7u^3-18u^2+12u)\mathbf{p}_1 \\
&\quad +b(-11u^3+18u^2)\mathbf{p}_2+cu^3\mathbf{p}_3 && \text{for } 0\le u<1 \\
\mathbf{p}_2(u) &= a(2-u)^3\mathbf{p}_1+b(7u^3-36u^2+54u-18)\mathbf{p}_2 \\
&\quad +c(-3u^3+12u^2-12u+4)\mathbf{p}_3+c(u-1)^3\mathbf{p}_4 && \text{for } 1\le u<2 \\
\mathbf{p}_3(u) &= c(3-u)^3\mathbf{p}_2+c(3u^3-24u^2+60u-44)\mathbf{p}_3 \\
&\quad +c(-3u^3+21u^2-45u+31)\mathbf{p}_4+c(u-2)^3\mathbf{p}_5 && \text{for } 2\le u<3 \\
\mathbf{p}_4(u) &= c(4-u)^3\mathbf{p}_3+c(3u^3-33u^2+117u-131)\mathbf{p}_4 \\
&\quad +b(-7u^3+69u^2-219u+227)\mathbf{p}_5+a(u-3)^3\mathbf{p}_6 && \text{for } 3\le u<4 \\
\mathbf{p}_5(u) &= c(5-u)^3\mathbf{p}_4+b(11u^3-147u^2+645u-925)\mathbf{p}_5 \\
&\quad +a(-7u^3+87u^2-357u+485)\mathbf{p}_6+(u-4)^3\mathbf{p}_7 && \text{for } 4\le u\le 5
\end{aligned}
$$

where $\mathbf{p}_i(u)$, $(i = 1, \ldots, 5)$, is the $i^{\underline{th}}$ curve segment. Therefore, the resulting curve is a composite sequence of five curve segments connected with c^2 continuity; an example is shown in figure 3.2. Again, the curve passes through only the first and last points $\mathbf{p}_0$ and $\mathbf{p}_7$ because

$$\mathbf{p}(0) = \mathbf{p}_1(0) = \mathbf{p}_1 \quad \text{and} \quad \mathbf{p}(5) = \mathbf{p}_5(5) = \mathbf{p}_7.$$

Also, the equations (3.7) can be reparametrized as follows:

$$
\begin{aligned}
p_1(u) &= (1-u)^3 p_0 + a(7u^3-18u^2+12u)p_1 \\
&\quad + b(-11u^3+18u^2)p_2 + cu^3 p_3 && \text{for } 0 \le u < 1 \\
p_2(u) &= a(1-u)^3 p_1 + b(7u^3-15u^2+3u+7)p_2 \\
&\quad + c(-3u^3+3u^2+3u+1)p_3 + cu^3 p_4 && \text{for } 0 \le u < 1 \\
p_3(u) &= c(1-u)^3 p_2 + c(3u^3-6u^2+4)p_3 \\
&\quad + c(-3u^3+3u^2+3u+1)p_4 + cu^3 p_5 && \text{for } 0 \le u < 1 \\
p_4(u) &= c(1-u)^3 p_3 + c(3u^3-6u^2+4)p_4 \\
&\quad + b(-7u^3+6u^2+6u+2)p_5 + au^3 p_6 && \text{for } 0 \le u < 1 \\
p_5(u) &= c(1-u)^3 p_4 + b(11u^3-15u^2-3u+7)p_5 \\
&\quad + a(-7u^3+3u^2+3u+1)p_6 + u^3 p_7 && \text{for } 0 \le u \le 1
\end{aligned}
\tag{3.8}
$$

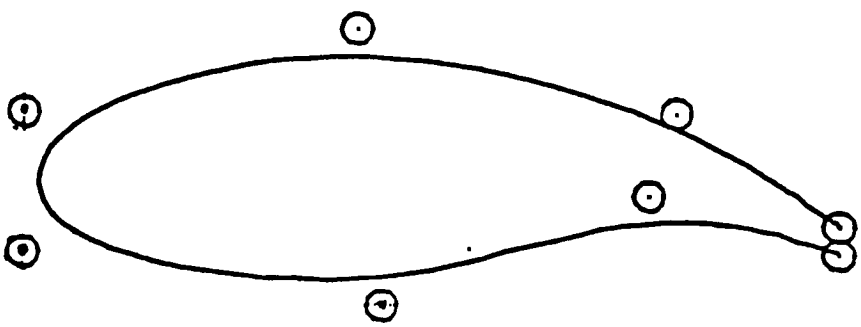


Figure 3.2 Non-periodic cubic B-spline curve

Once again, we note that only four control points influence each curve segment. Conversely, a control point can influence the

shape of at most only four curve segments. In fact, we can generalize this result: Each segment of a B-spline curve is influenced by only k+1 control points; conversely each control point influences only k+1 curve segments.

3.2.2 Periodic Curves

If we increase the number of control points, we can correctly infer that, except for the blending functions influenced by end points or those very near them (depending on k), the "interior" blending functions are independent of n. For examples,

(A) if n = 7 and k = 2, we would find that

$$B_{2,2}(u) = B_{3,2}(u) = B_{4,2}(u) = B_{5,3}(u)$$

and, hence, the coefficient (blending) functions of the i^{th} curve segment $\mathbf{p}_i(u)$, (i = 2, ... , 5), are the same, respectively.

(B) if n = 9 and k = 3, we would find that

$$B_{3,3}(u) = B_{4,3}(u) = B_{5,3}(u) = B_{6,3}(u)$$

and therefore it comes to the same coefficient (blending) functions in the i^{th} curve segment $\mathbf{p}_i(u)$, i = 2, ... , 5.

3.2.2.1 Open Curves

If we pursue this idea of blending-function independence of n, we can develop a more convenient and more familiar matrix notation to the open (periodic) B-spline curve and closed (periodic) B-spline curve.

Now, we discuss the case of open (periodic) curves. Later, we will discuss the case of closed curves. Let us choose an interval in i so that $k+1 \leq i \leq n$. This assumption simplifies the calculation of the t_i knot values so that $i-k \leq t_i \leq n-k$. After applying the recursive formulas in (2.2) and (2.3), we have an expression for $\mathbf{p}(u)$ over an arbitrary segment of the curve, say, for the interval $j \leq u < j+1$, that is,

$$(3.9) \qquad \mathbf{p}(u) = 0.5 \{ (j+1-u)^2 \mathbf{p}_j + [(u-j+1)(j+1-u) + (j+2-u)(u-j)] \mathbf{p}_{j+1} + (u-j)^2 \mathbf{p}_{j+2} \}$$

Here, $0 \leq j \leq n-k$, or more precisely, $0 \leq u \leq n-k+1$.

To reparametrize Eq.(3.9), replace u by u+j, so that, for the $j^{\underline{th}}$ interval, we have

$$(3.10) \qquad \mathbf{p}_j(u) = 0.5[(1-u)^2\mathbf{p}_j+(-2u^2+2u+1)\mathbf{p}_{i+1}+u^2\mathbf{p}_{j+2}]$$

where $0 \leq u < 1$. We can now easily rewrite Eq.(3.10), replacing j by i-1 (i now denotes the curve segment number) and obtain the $i^{\underline{th}}$ curve segment with degree 2 and $0 \leq u < 1$ as follows:

$$(3.11)\quad \mathbf{p}_i(u) = 0.5[(1-u)^2\mathbf{p}_{i-1}+(-2u^2+2u+1)\mathbf{p}_i+u^2\mathbf{p}_{i+1}]$$

for i = 1, ... , n-1

$$= \mathbf{U}_2\mathbf{M}_2\mathbf{P}_2$$

where $\mathbf{U}_2 = [u^2 \;\; u \;\; 1]$,

$$\mathbf{M}_2 = \frac{1}{2}\begin{bmatrix} 1 & -2 & 1 \\ -2 & 1 & 0 \\ 1 & 1 & 0 \end{bmatrix}$$

and

$$\mathbf{P}_2 = \begin{bmatrix} \mathbf{p}_{i-1} \\ \mathbf{p}_i \\ \mathbf{p}_{i+1} \end{bmatrix}$$

Similarly, the $i^{\underline{th}}$ curve segment with degree 3 is, (letting c = 1/6), for i = 1, ... , n-2,

$$(3.12)\quad \mathbf{p}_i(u) = c(1-u)^3\,\mathbf{p}_{i-1} + c(3u^3-6u^2+4)\,\mathbf{p}_i + c(-3u^3+3u^2+3u+1)\,\mathbf{p}_{i+1} + cu^3\,\mathbf{p}_{i+2}$$

$$= \mathbf{U}_3\mathbf{M}_3\mathbf{P}_3$$

where $\mathbf{U}_3 = [u^3 \;\; u^2 \;\; u \;\; 1]$,

$$\mathbf{M}_3 = c\begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix}$$

and

$$\mathbf{P}_3 = \begin{bmatrix} \mathbf{p}_{i-1} \\ \mathbf{p}_i \\ \mathbf{p}_{i+1} \\ \mathbf{p}_{i+2} \end{bmatrix}$$

Figure 3.3 presents the open periodic B-spline curves for k = 2 (figure 3.3(a)) and k = 3 (figure 3.3(b)). Notice that neither the k = 2 curve nor k = 3 curve passes through any of the control points.

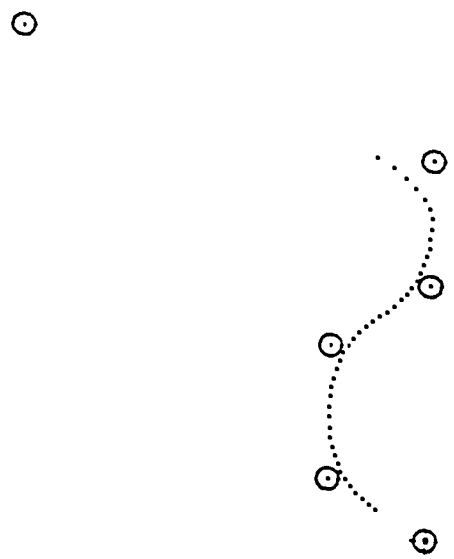


Figure 3.3(a) Open periodic quadric B-spline curve

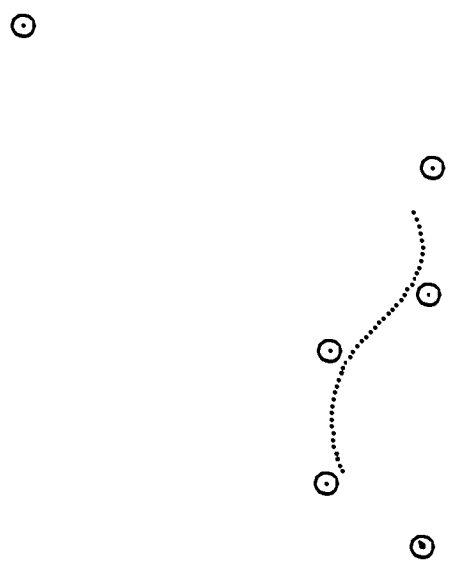


Figure 3.3(b) Open periodic cubic B-spline curve

Equations (3.11) and (3.12) are for specific k values. The general formulation, $k \geq 0$, is

$$\mathbf{p}_i(u) = \mathbf{U}_k \mathbf{M}_k \mathbf{P}_k \quad \text{for } i = 1, \ldots, n-k+1 \tag{3.13}$$

where

$$\mathbf{U}_k = [\, u^k \;\; u^{k-1} \;\; \ldots \;\; u \;\; 1 \,]$$

and

$$\mathbf{P}_k = [\, \mathbf{p}_j \,] \quad \text{for } j = i-1, \ldots, i+k-1$$

The matrix $\mathbf{M}_k$ is determined in the same way as $\mathbf{M}_2$ or $\mathbf{M}_3$ was earlier above. The matrix $\mathbf{M}_k$ is called the **M transformation**

matrix for B-spline curve. Notice that $\mathbf{M}_0 = [\ 1\ ]$ and $\mathbf{M}_1 = [\ \mathbf{a}\ \mathbf{b}\ ]$, where $\mathbf{a} = [\ -1\ \ 1\ ]^t$ and $\mathbf{b} = [\ 1\ \ 0\ ]^t$.

Reminder: The number of segment i is determined here for open curves. The closed B-spline curves will be investigated later in this section. Notice that the curve does not pass through any of the control points if we simple apply a formulation as in eq. (3.11) or (3.12) to the set of control points. This method produces a **periodic** B-spline curve. It is **periodic** because the blending function repeats itself identically over successive intervals of the parametric variable. On the other hand, the curves in figures (3.1) and (3.2) are **nonperiodic** B-spline curves.

3.2.2.2 Closed Curves

The periodic B-spline curves are particularly well suited to produce closed curves. Equation (3.13) is easily adapted by simple modifications of the segment number range and the subscripts on the control points. Let us nearly investigate closed B-spline curves now.

For a **closed** curve: Once more, the knot values relate the parametric variable u to the $\mathbf{p}_i$ control points. Thus, they will be defined as follows:

$$t_i = (\ i - k\ ) \bmod (\ n + 1\ ) \qquad \text{for } i = 0, 1, \ldots, (n+k+2).$$

where mod(n+1) is the remaindering operator (that is, 3 mod 4 = 3, 8 mod 6 = 2, 5 mod 5 = 0, and so on). The range of the

parametric variable u is

$$0 \leq u \leq n.$$

It can be shown that the expression for a closed periodic B-spline curve is same as eq. (3.13). In particular,

for k = 2

$$p_i(u) = U_2 M_2 \begin{bmatrix} p_{(i-1) \bmod (n+1)} \\ p_{(i) \bmod (n+1)} \\ p_{(i+1) \bmod (n+1)} \end{bmatrix} \qquad i = 1, \ldots, n+1 \tag{3.14}$$

and

for k = 3

$$p_i(u) = U_3 M_3 \begin{bmatrix} p_{(i-1) \bmod (n+1)} \\ p_{(i) \bmod (n+1)} \\ p_{(i+1) \bmod (n+1)} \\ p_{(i+2) \bmod (n+1)} \end{bmatrix} \qquad i = 1, \ldots, n+1 \tag{3.15}$$

The curve in figure (3.4) is an example of a closed periodic B-spline curve. Here, n = 5 and k = 3; using these values in Eq.(3.15), we have

$$p_i(u) = U_3 M_3 \begin{bmatrix} p_{(i-1) \bmod 6} \\ p_{(i) \bmod 6} \\ p_{(i+1) \bmod 6} \\ p_{(i+2) \bmod 6} \end{bmatrix} \qquad \text{for } i = 1, \ldots, 6$$

Expand this equation to obtain

$$p_1(u) = \mathbf{U}_3\mathbf{M}_3 [\, p_0 \;\; p_1 \;\; p_2 \;\; p_3 \,]^t$$

$$p_2(u) = \mathbf{U}_3\mathbf{M}_3 [\, p_1 \;\; p_2 \;\; p_3 \;\; p_4 \,]^t$$

$$p_3(u) = \mathbf{U}_3\mathbf{M}_3 [\, p_2 \;\; p_3 \;\; p_4 \;\; p_5 \,]^t$$

$$p_4(u) = \mathbf{U}_3\mathbf{M}_3 [\, p_3 \;\; p_4 \;\; p_5 \;\; p_0 \,]^t$$

$$p_5(u) = \mathbf{U}_3\mathbf{M}_3 [\, p_4 \;\; p_5 \;\; p_0 \;\; p_1 \,]^t$$

$$p_6(u) = \mathbf{U}_3\mathbf{M}_3 [\, p_5 \;\; p_0 \;\; p_1 \;\; p_2 \,]^t$$

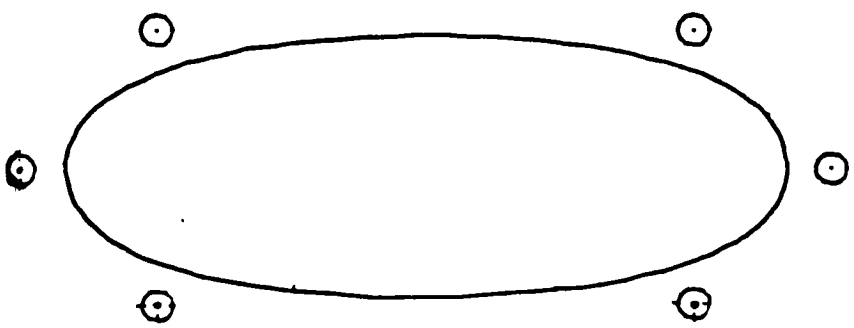


Figure 3.4 Closed, periodic cubic B-spline curve with 6 control points

We can define B-spline curves with multiply-coincident control points. The examples in figures (3.6) and (3.7) show the effect of having one, two, or three control points at the same location. First, we should look at the example in figure (3.5) which

shows a closed periodic B-spline curve with five control points. These points are indicated by circle marks. The B-spline curve in figure (3.6a) is obtained when we insert five(5) additional inner control points. The closed curves in figures (3.6b) and (3.6c) are constructed with the five(5) inner control points doubled and tripled, respectively. From these pictures, we can see the materialization of the sharp corners. The picture in figure (3.6b) does look like a "star". If we insert a point on either side of each segment, we can see a pulling effect , which produces a petal: Again, each picture in figure (3.7) corresponds to the picture in figure (3.6), that is, with the five inner control points single, doubled and tripled, respectively. Now, the picture in figure (3.7c) does look like a "flower" on a plum tree. Certainly, we will get a different shape of petal if we relocate the pulling points.

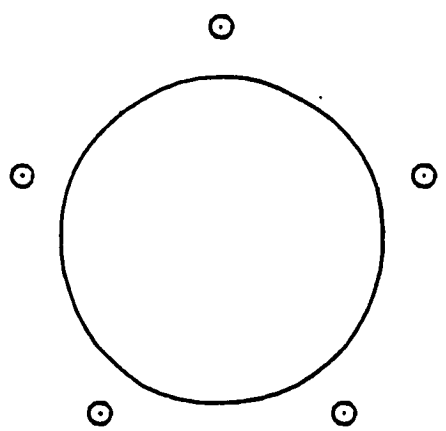


Figure 3.5 Closed, periodic cubic B-spline curve
with 5 control points

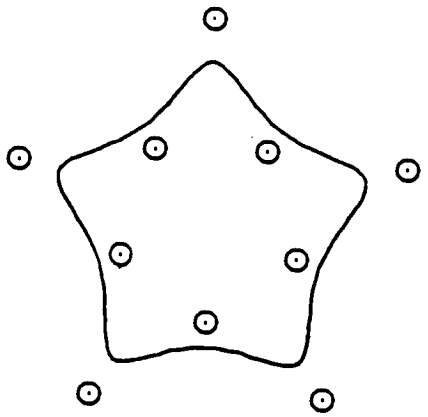


Figure 3.6 Closed, periodic cubic B-spline curve

with 10 control points

(a) Without any coincident inner control point

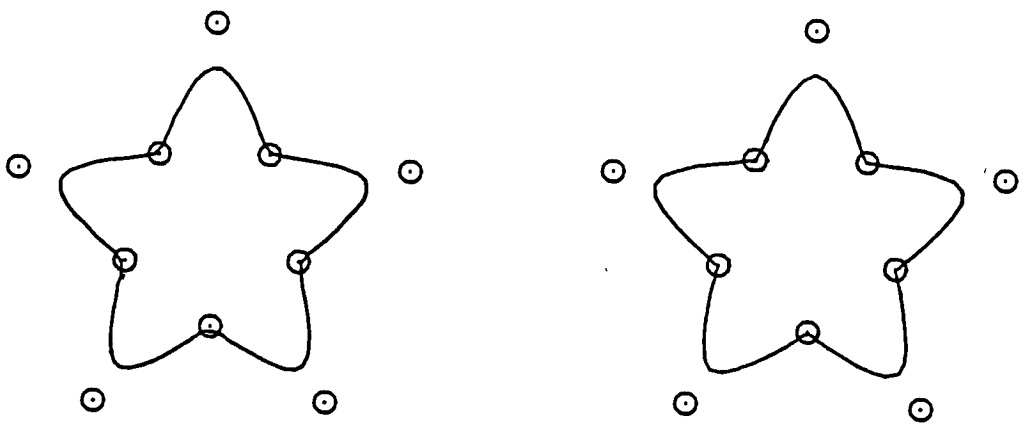


Figure 3.6 (b) With doubled-coincident inner control points

(c) With tripled-coincident inner control points

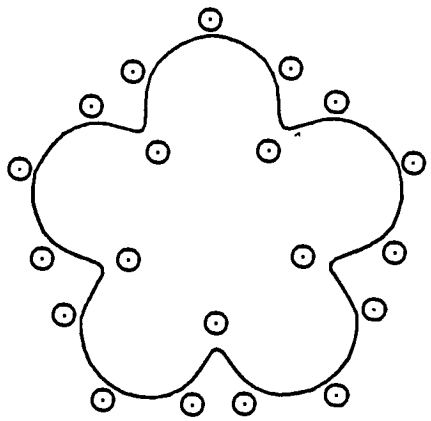


Figure 3.7 Closed, periodic cubic B-spline curve

with 20 control points,

(a) Without any coincident inner controal point and

with multiple-pulling sided control points

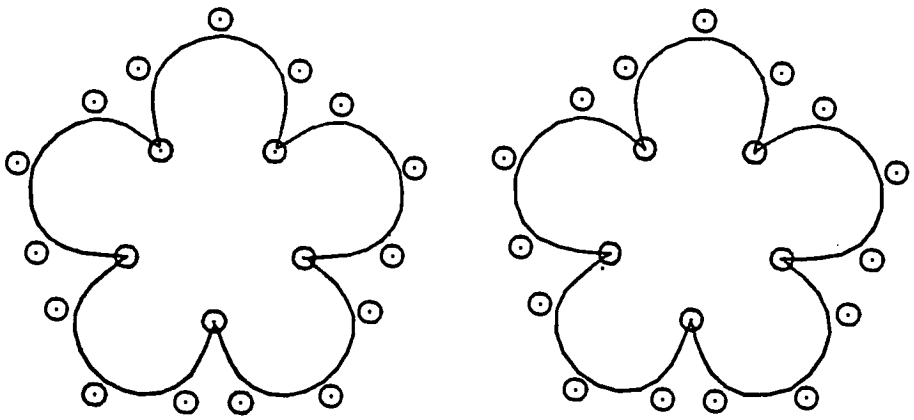


Figure 3.7(b) With multiple-pulling sided control points and

with doubled-coincident inner control points

(c) With multiple-pulling sided control points and

with tripled-coincident inner control points

Chapter 4 B-Spline Surfaces - Tensor Product Surfaces

The simplest mathematical element we use to model a surface is a **patch**. A patch is a curve-bounded collection of points whose coordinates are given by continuous, two-parameter, single-valued mathematical functions of the form

$$x = x(u,v) \qquad y = y(u,v) \qquad z = z(u,v).$$

The parametric variables u and v are constrained to the limits,

$$0 \leq u \leq 1 \text{ and } 0 \leq v \leq 1.$$

A patch can be defined as following:

$$\mathbf{p}(\mathbf{x}) = \mathbf{p}(u,\ v)$$

where

$$\mathbf{x} = [\ x(u,\ v),\ y(u,\ v),\ z(u,\ v)\ ],$$

and

$$0 \leq u \leq 1,\ \ 0 \leq v \leq 1.$$

Varying both parameters from 0 to 1 defines all points on a surface patch. If one parameter is assigned a constant value and the other parameter is varied from 0 to 1, the result is a curve on the patch. Thus, we can construct a patch by fixing the value of one of the parametric variables to result in a curve on the patch in terms of the other variable, which remains free. By continuing this process first for one variable and then the other for any number of arbitray values in the allowed interval, we form a parametric net of two one-parameter families of the curves on the patch so that just one curve of each family passes through

each point **p**(u, v). Figure 4.1 depicts a single surface patch that is drawn by a number of curves on the patch which are defined by constant u and constant v.

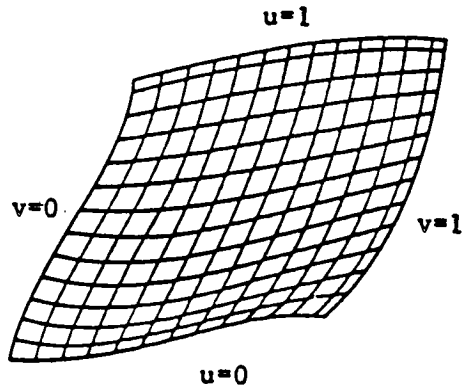


Figure 4.1 Single surface patch depicted by curves of the constant u and constant v

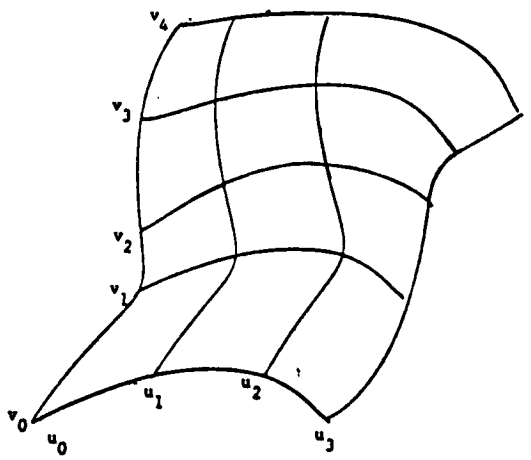


Figure 4.2 Composite surface

In analogy to curves it is also possible to build up complex surfaces from a number of patches. Associated with every patch is a set of boundary conditions, for example, C^r- continuity of adjacent surface patches ($r \geq 0$). Thus, a segmentation of a surface into several segments corresponds to a partition of the domain by "knotlines" $u_0 \leq u_1 \leq \cdots \leq u_p$, and $v_0 \leq v_1 \leq \cdots \leq v_q$. Figure 4.2 shows a composite surface defined by a 3 by 4 array of 12 patches.

4.1 Tensor Product Surfaces

One of the most popular schemes for surface description is called the tensor product surface. The reasons are that the properties of the tensor product surfaces can easily be deduced from properties of the underlying curve scheme; a general (and rather theoretical) treatment is given by de Boor [de Boor, '78, ch. XVII]. We give a more intuitive derivation, see also Bohm [1984]. Let

$$\mathbf{p}(u) = \sum_{i=0}^{m} \mathbf{c}_i F_i(u)$$

be a 3D (or 2D) curve expressed linearly in terms of basis functions F_i. First let this curve sweep out a surface by moving it through space, thereby possibly deforming it. Such a sweep can be described by letting each $\mathbf{c}_i$ trace out a curve $\mathbf{c}_i(v)$. If all these

curves $c_i(v)$ are likewise linear combinations of basis functions $G_j(v)$, i.e.

$$c_i(v) = \sum_{j=0}^{n} p_{i,j} G_j(v),$$

the resulting surface $p(u,v)$ is called a **tensor product surface**:

$$\begin{aligned} p(u, v) &= \sum_{i=0}^{m} c_i(v) F_i(u) \\ &= \sum_{i=0}^{m} \left(\sum_{j=0}^{n} p_{i,j} G_j(v) \right) F_i(u) \\ &= \sum_{i=0}^{m} \sum_{j=0}^{n} p_{i,j} F_i(u) G_j(v). \end{aligned}$$

The products $F_i(u)\ G_j(v)$ are basis functions for these surfaces. The $p_{i,j}$ are called the control points; in their natural ordering they are the vertices of the de Boor net of the surface.

4.2 Tensor Product B-Spline Surfaces

The formulation of a B-spline surface follows directly from our formulation of B-spline curves. Furthermore, the B-spline surface is defined in terms of a characteristic polyhedron. The shape of the surface approximates the polyhedron. The approximation is weaker the higher the value of k and l. Thus,

$$(4.1) \qquad \mathbf{p}(u, v) = \sum_{i=0}^{m} \sum_{j=0}^{n} \mathbf{p}_{i,j} \, B_{i,k}(u) \, B_{j,l}(v).$$

The $\mathbf{p}_{ij}$ are the vertices of the defining polyhedron. They are again called de Boor points. The $B_{i,k}(u)$ and $B_{j,l}(v)$ are the blending functions of the same form as those for B-spline curves; see equation (3.1). The degree of each of the blending-function polynomials $B_{i,k}(u)$ and $B_{j,l}(v)$ is controlled by k and l, respectively. The surface defined above, is called the **tensor product B-spline surface** which has remained on the most popular schemes in the **CAGD**.

Many properties of B-spline surfaces are easily carried over:

(i) Lines of constant v, so-called u-lines, are B-spline curves of degree k with de Boor points

$$\mathbf{c}_i(v) = \sum_{j=0}^{n} \mathbf{p}_{i,j} \, B_{j,l}(v).$$

(ii) A change of only one de Boor point $\mathbf{p}_{i,j}$ will only affect the surface for $u_i \le u \le u_{i+k+1}$, $v_j \le v \le v_{j+l+1}$. The patch $u_i \le u \le u_{i+1}$, $v_j \le v \le v_{j+1}$ is only influenced by the de Boor points $\mathbf{p}_{i-k, j-l}, \cdots, \mathbf{p}_{i,j}$.

(iii) For any point u, v the point $\mathbf{p}(u, v)$ can be computed by appling the de Boor algorithm several times (referring the

section 2.3) : first one determines the (m+1) de Boor points

$$p_j(u) = \sum_{j=0}^{m} p_{i,j} B_{i,k}(u) \qquad j = 0, 1, 2, \ldots, n$$

and then the point p(u, v) on the v-line

$$p(u, v) = \sum_{j=0}^{m} p_j(u) B_{j,l}(v).$$

4.3 Open B-Spline Surfaces

For a nonperiodic B-spline, the values of m, n, k, l, and the t_i and t_j knot values are selected and computed recursively using equations (3.2) and (3.3). Notice that two sets of the knot values are required and that the control points form an (m+1) by (n+1) array.

There is a matrix form for B-spline surface: it is similar to the matrix form we developed for B-spline curve. Recall that the B-spline curve is computed in segments of a unit interval on the parametric variable u; see equations (3.11), (3.12), and (3.13). A unit square on the parametric variables u and v is used to compute patches on the B-spline surface. The general matrix form of an open, periodic B-spline surface that approximates an (m+1) by (n+1) rectangular array of points is

$$p_{st}(u, v) = \mathbf{U}_k \mathbf{M}_k \mathbf{P}_{kl} \mathbf{M}_l^t \mathbf{V}_l^t \qquad \begin{matrix} 1 \le s \le m+1-k \\ 1 \le t \le n+1-l \\ 0 \le u, v \le 1 \end{matrix} \tag{4.2}$$

where k and l denote the parameters that control the continuity of the surface and the degree of the blending-function polynomials; s and t identify a particular patch in the surface. The range on s and t is a function of the parameters k and l and the dimensions of the rectangular array of control points. The matrix $\mathbf{U}_k$ is

$$\mathbf{U}_k = [\, u^k, u^{k-1}, \ \ldots\ , u, 1 \,] \tag{4.3}$$

and $\mathbf{V}_l$ is

$$\mathbf{V}_l = [\, v^l, v^{l-1}, \ \ldots\ , v, 1 \,] \tag{4.4}$$

Elements of the (k+1) by (l+1) matrix of the control points $\mathbf{P}_{kl}$ depend on the particular patch to be evaluated. Let p_{ij} denote these matrix elements; then

$$\mathbf{P}_{kl} = [\, p_{ij} \,] \qquad \begin{matrix} s-1 \le i \le s+k-1 \\ t-1 \le j \le t+l-1 \end{matrix} \tag{4.5}$$

The matrices $\mathbf{M}_k$ and $\mathbf{M}_l$ are identical to the $\mathbf{M}$ transformation matrix for B-spline curves described on chapter 3.

If the B-spline surface is partially closed (that is, rolled into

an open-ended tube), then the ranges on s, t and i, j must reflect this closure. For example, if the v = constant curves are closed, then

$$
\begin{aligned}
& 1 \le s \le m+1 \\
& 1 \le t \le n + 1 - l \\
& (s - 1) \bmod (m + 1) \le i \le (s + k - 1) \bmod (m + 1) \\
& t-1 \le j \le t + l - 1
\end{aligned} \tag{4.6}
$$

Similar expressions apply when the u = constant curves are closed.

In order to understand the relationship between the control points and patches and simulate to design of the B-spline surfaces, we idealize an array of control points and the patches they produce with a two-dimensional diagram, arranging the points on a rectangular grid. Solid lines represent the edges of the faces of the characteristic polyhedron. Dashed lines represent the patch bounaries. For some idealization, these lines overlap: when this happens, the dashed lines are shown. The patches are lightly shaded to highlight them. Let us look at some examples. The first one is a bicubic patch. A bicubic B-spline patch is produced when $k = l = 3$. This surface exhibits that the second-derivative continuity exists at all points on this surface. Here, a minimum 4 x 4 array of control points is required (see figure 4.3(a)). Figure 4.3(b) gives an example of a bicubic B-spline surface with 16 control points. Note that it is often more efficient to define a B-spline surface with $k = l$.

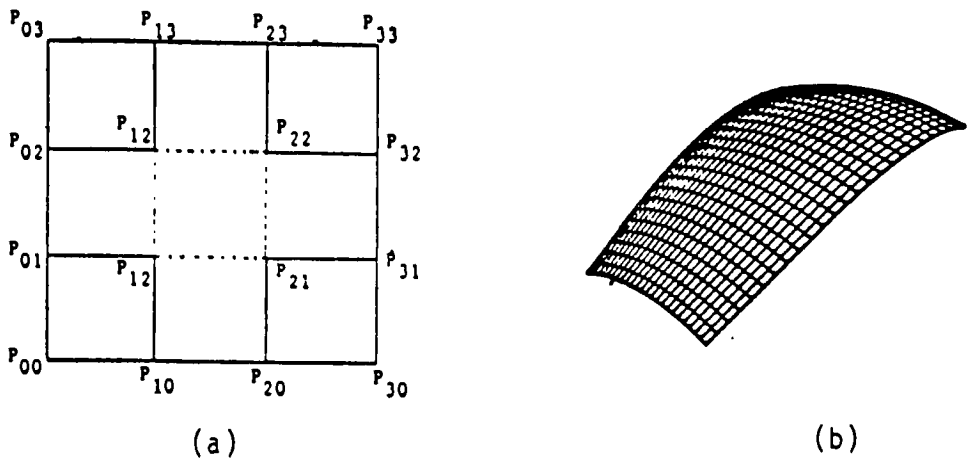
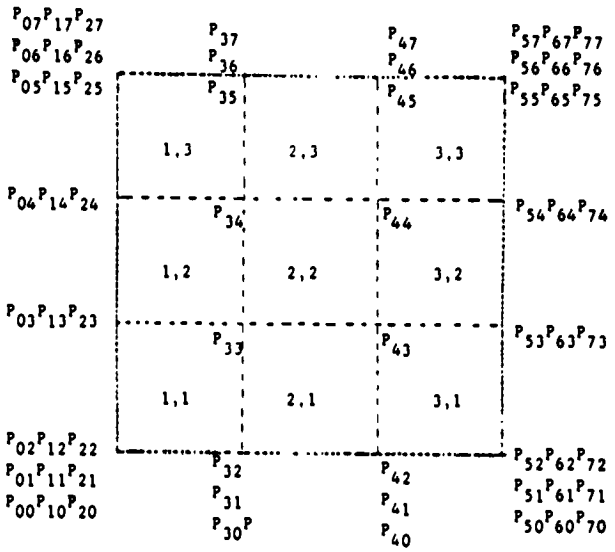


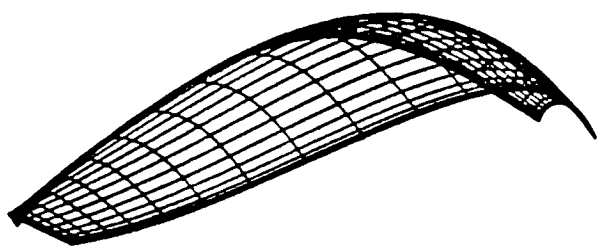
Figure 4.3 Open bicubic B-spline surface idealization and patch

The effect of B-spline surfaces with multiply-coincident control points is analogous to the B-spline curves with multiply-coincident control points. For example, the cubic B-spline curve will pass through the point with tripled-coincident control point. A bicubic B-spline surfaces with 8 x 8 control points is shown in figure 4.4(b). That is, m = n = 7. A two-dimensional idealization is shown in figure 4.4(a). In either direction of u or v, there are tripled-coincident control points at each endpoint. We shall note that there are only 16 distinct control points in this case. The resulting B-spline surface is shown in figure 4.4(b). There are 9 distinct B-spline surface patches are joined together.

The picture in figure 4.4(b) dispicts a bowl of a spoon. The complete picture of a spoon, bowl and handle, is shown on figure 4.5.



(a)



(b)

Figure 4.4 Open bicubic B-spline surface idealization and surface with tripled-coincident boundary control points

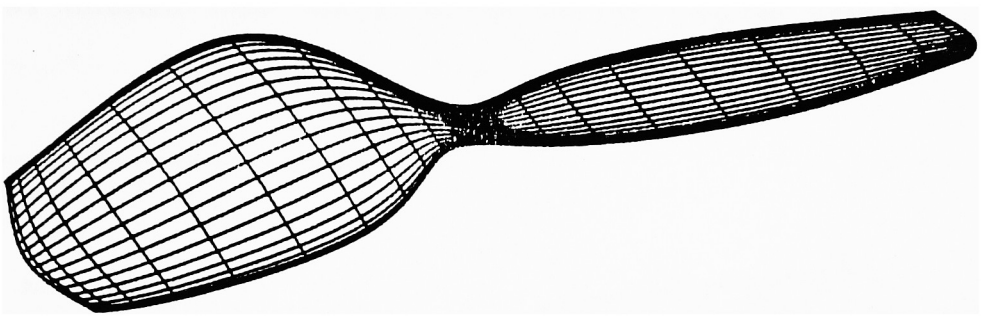


Figure 4.5 B- spline spoon

4.4 Partially Closed B-Spline Surfaces

A partial closed B-spline surface results when we impose the conditions in equations (4.6). We will show this effect with two examples. Figure 4.6(c) is the first example. Here, k = 3 = l, with m = 5 = n , to produce a 6 x 6 patch array. The v = constant curves are closed; the u = constant curves are open. A two-dimensional idealization is shown in figure 4.6(a) and a three-dimensional idealization is shown in figure 4.6(b). Only 36 control points produce the complex surfaces in figure 4.6(c).

The second example is the frame of a bottle, shown in figure 4.7. We use 9 x 4 or 36 distinct control points to produce this picture. The transition from a round screw cap to a round body section is very smoothly. However, we should remenber that one advantage of the B-spline formulation is its ability to preserve

arbitrarily high degrees of continuity over complex surfaces. This follows directed from the properties of the B-spline curves.

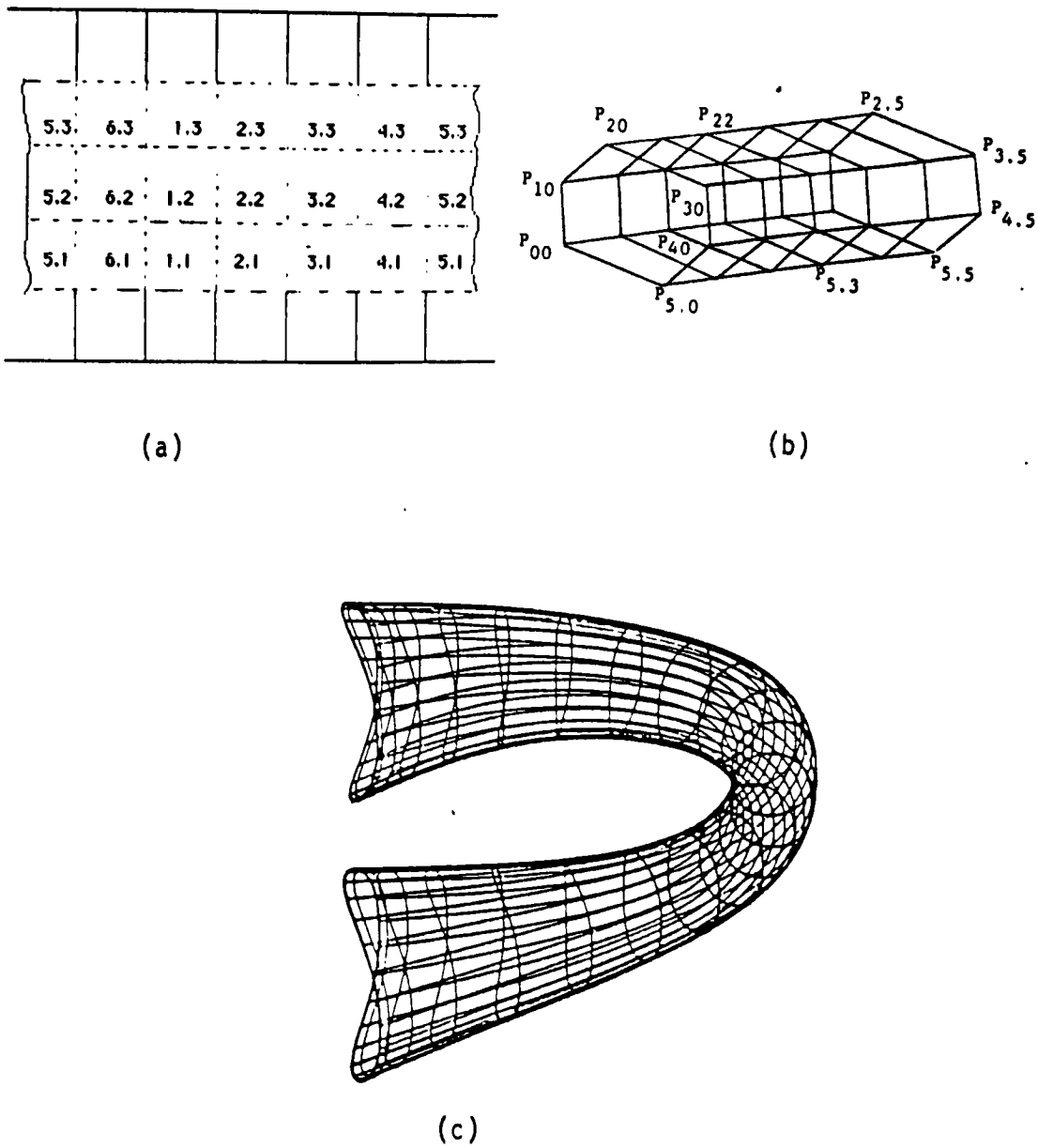


Figure 4.6 Partially closed bicubic B-spline surface

Also, any change in the local shape of a B-spline surface is not propagated throughout the entire surface. The figure 4.8 is constructed with multiple control at the desired locations. This advantage of the B-spline surface is shown in effect. The transition between any two adjacent parts: round screw cap, blend-surface upper section and round body section, is now critical.

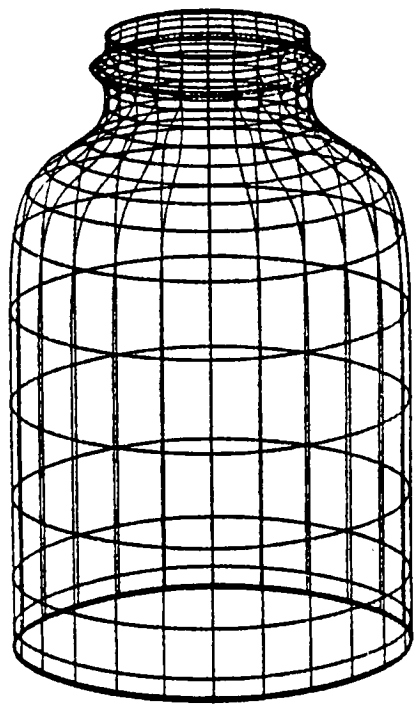


Figure 4.7 Smooth B-spline bottle

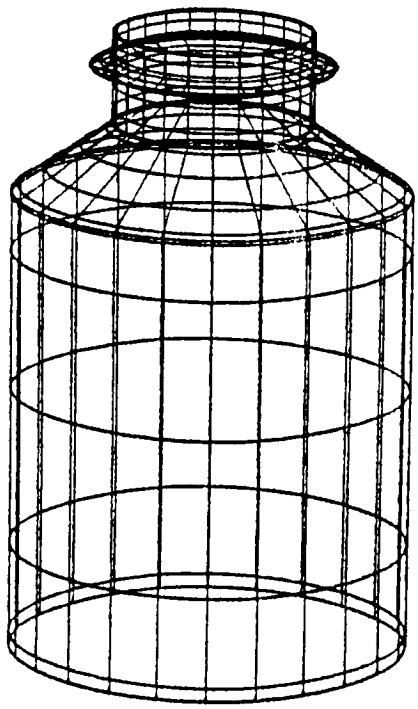


Figure 4.8 Critical B-spline bottle

Chapter 5 Multivariate B-Splines

The approximation method given by (4.1) has most of the desirable features of the univariate scheme; specifically local behavior, positive basis functions summing to one. This technique has enjoyed an immense popularity as a tool in the field of **CAGD**. Unfortunately tensor product schemes also possess some severe limitations; perhaps their most restrictive shortcoming is their dependency on rectilinear knot distributions.

A knot topology of this configuration does not easily lend itself to modelling a wide variety of objects. For example, see figure (5.1), (Kochevar, [1984]), which depicts the wing-body fairing of an airplane. The region where the leading edge of the wing joins the fuselage cannot be satisfactorily modelled unless some special construction is attempted there such as coalescing knot points or substituting some non B-spline surface. Either attempt, aside from the added complexity of the construction, has a tendency to circumvent the desirable qualities which prompted the use of B-splines initially. We need an alternate generalization of the univariate B-spline approximation that admits arbitrary knot configurations while still preserving its desirable features. One possible answer is the multivariate B-spline that can be defined for almost any knot configuration giving approximations on any compact polyhedral region in a Euclidean space of any dimension.

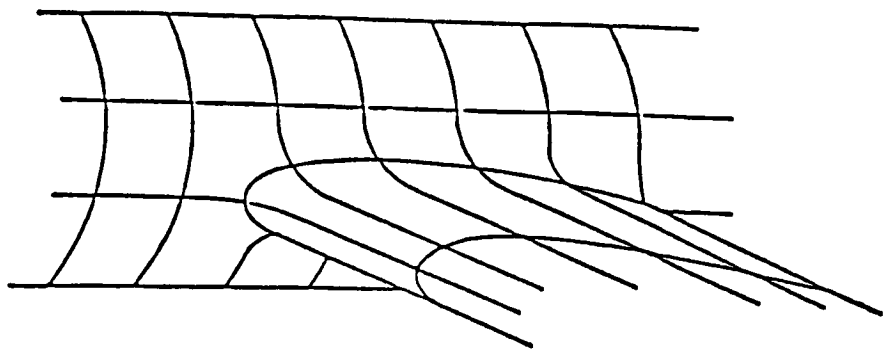


Figure 5.1 A surface not easily modeled by tensor product B-splines (see Kochevar [1984], page 161)

First, we introduce notation to be used throughout the sequel. Elements of the Euclidean space $\mathbf{R}^s$, $s \geq 1$, are denoted by bold face $\mathbf{x}, \mathbf{y}, \ldots$; superscripts will denote indices for elements of $\mathbf{R}^s$, and subscripts will give the component of a particular vector. So, $x_q{}^p$ indicates the q^{th}-component of the p^{th}-element from a collection of vectors. Any lower case letter without bold face, such as $t, x, \ldots$, denotes a real number. If S is a collection of vectors then $[S]$ will denote the complex hull of S, and if $\mathbf{y}^0, \mathbf{y}^1, \ldots, \mathbf{y}^n$ is a set of points in $\mathbf{R}^n$ then $\sigma = [\mathbf{y}^0, \mathbf{y}^1, \ldots, \mathbf{y}^n]$ will be referred to as the n-simplex which has the points $\mathbf{y}^0, \mathbf{y}^1, \ldots, \mathbf{y}^n$ as its vertices. In addition $\mathrm{Vol}_s(A) = \mathrm{Vol}_s A$ denotes the s-dimensional volume. $\chi_A(x)$ is the characteristic function of a given set A.

5.1 The Curry - Schoenberg Approach

The object of this chapter is to motivate the notion of multivariate B-spline as well as to discuss some of its basic properties. In order to reach this goal, we recall some properties of the univariate B-spline which will serve as a guide for the multivariate case.

There are several ways to define the (univariate) the B-spline, the most useful of which involves divided differences.

Let the sequence $\{t_i\}$ of the knots be given, and denote the truncated power function by

$$(5.1) \qquad T(x,t) = (t-x)_+^k = \begin{cases} (t-x)^k & \text{if } t \geq x \\ 0 & \text{if } t \leq x. \end{cases}$$

Then the i^{th} normalized B-spline of degree k (order k+1) is defined to be

$$(5.2) \qquad B_{i,k}(x) = (t_{i+k+1} - t_i)\ T[\, x\, ;\, t_i, t_{i+1}, \cdots, t_{i+k+1}\,]$$

Here, the notation for divided differences of a function of two variables uses a separating semicolon, and then the points of the divided differences are listed on each side of it. In this case the differences are applied only to the t variable; x still varies in the

normal fashion.

It can be shown that definition (5.2) is equivalent to equations in (2.2) and (2.3). However, none of the above definitions suggest any way of defining the B-spline in several variables. The key to go beyond the univariate case is the Hermite-Gennchi formula for a divided difference.

Let

$$S^n = \{ (\tau_0, \ldots, \tau_n) \mid \tau_j \geq 0, \ \sum_{i=0}^{n} \tau_i = 1 \}$$

be the standard n-simplex; then

$$(5.3) \qquad [t_0, \ldots, t_n]\, g = \int_{S^n} g^{(n)} (t_0\tau_0 + \ldots + t_n\tau_n)\ d\tau_1 \ldots d\tau_n$$

where $[t_0, \ldots, t_n]\, g$ is the divided difference of g at $t_0, \ldots, t_n$. Applying this formula, gives, (Micchelli [1979]),

$$(5.4) \qquad \int_{-\infty}^{+\infty} g(x)\, M(x \mid t_0, \ldots, t_n)\, dx$$

$$= n! \int_{S^n} g^{(n)} (t_0\tau_0 + \ldots + t_n\tau_n)\ d\tau_1 \ldots d\tau_n$$

where

$$(5.5) \qquad M(x \mid t_0, \ldots, t_n) = n!\ T[x ; t_0, \ldots, t_n]$$

This equation obviously holds for any locally integrable univariate function g. The function $M(x \mid t_0, \ldots, t_n)$ is the (nonnormalized) univariate B-spline of order n or degree (n 1) which differs from that defined in (2.2), (2.3) or (5.2) by a constant factor.

Equation (5.4) also contains a geometric interpretation of B-splines which is due to Curry and Schoenberg. Their ideas is to transform the n-fold integral on the right side of (5.4) by using the following device. Let $y^0, \ldots, y^n$ be any vector in $\mathbf{R}^n$ whose first component agree with $t_0, \ldots, t_n$, respectively. That is, let

$$y^0 = (t_0, \ldots), \qquad y^1 = (t_1, \ldots), \quad \ldots\ldots, \qquad y^n = (t_n, \ldots).$$

We now view g as a function of n variables and perform the change of variables $t = \tau_0 y^0 + \tau_1 y^1 + \ldots + \tau_n y^n$ in (5.4). Then we can get, [Micchelli, 80], for $n \geq 2$ and $x \in \mathbf{R}^1$,

$$(5.6) \qquad M(x \mid t_0, t_1, \ldots, t_n) = \frac{\mathrm{Vol}_{n-1} \{ y \in \sigma \mid y_1 = x \}}{\mathrm{Vol}_n\, \sigma}$$

where σ is the simplex determined by the vectors $y^0, y^1, \ldots, y^n$,

$$\sigma = [\, y^0, y^1, \ldots, y^n \,]$$

$$= \{ y = \sum_{j=0}^{n} \tau_j y^j \mid (\tau_0, \tau_1, \ldots, \tau_n) \in S^n \}$$

and $\mathrm{Vol}_n \sigma$ is the n-dimensional volume of the set σ. Let us give two simple examples of the result (5.6).

First example, consider the case n = 2, or degree 1, and choose

$$y^0 = (t_0, 0), \qquad y^1 = (t_1, h), \qquad \text{and} \qquad y^2 = (t_2, 0),$$

where h is any nonnegative real number. In figure (5.2), the three points $(t_0, 0)$, (t_1, h) and $(t_2, 0)$ form the triangle T with area (1/2) h $(t_2 - t_0)$. The B-spline evaluated at x is the length of the set of points in the triangle whose first coordinate is x divided by the area of the triangle, that is,

$$(5.7a) \qquad M(x \mid t_0, t_1, t_2) = \frac{\text{length L}}{\text{area T}}$$

or, in more detail,

$$(5.7b) \quad M(x \mid t_0, t_1, t_2) = \begin{cases} \dfrac{2(x - t_0)}{(t_2 - t_0)(t_1 - t_0)} & \text{if } t_0 < t < t_1 \\ \dfrac{2(t_2 - x)}{(t_2 - t_0)(t_2 - t_1)} & \text{if } t_1 \le t < t_2 \\ 0, \text{ elsewhere} \end{cases}$$

Equation (5.7) differs from equation in (2.3) by the constant factor of $(t_2 - t_0)/2$.

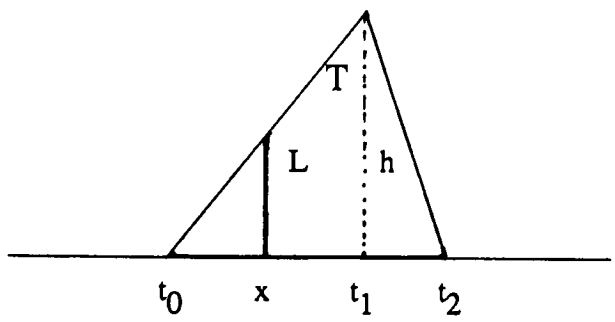


Figure 5.2 Geometric construction $M(x \mid t_0, t_1, t_2)$

Before discussing the second example, we would like to redefine the univariate B-spline with the geometric interpretations above. Let $\sigma = [\, y^0, \ldots, y^n]$ be any n-simplex in R^n such that its vertices project orthogonally onto each t_i, i.e. the first coordinate of y^i equals t_i for each $i = 0, 1, \ldots, n$. Then

$$(5.8)\qquad M(x \mid t_0, t_1, \ldots, t_n) = \begin{cases} \dfrac{\chi(x \mid [y^0, y^1])}{\mathrm{Vol}_1 \sigma} & \text{if } n = 1 \\ \dfrac{\mathrm{Vol}_{n-1}\{y \in \sigma \mid y_1 = x\}}{\mathrm{Vol}_n \sigma} & \text{if } n > 1 \end{cases}$$

where χ is the characteristic function of the set σ. Each function, $M(x \mid t_0, \ldots, t_n)$, is an unnormalized B-spline of degree $k = n - 1$ which differs from that defined in (2.2) and (2.3) by the constant

factor of $(t_n \quad t_0)/n$.

Now, we can give the second example. Consider the case n = 3 and degree k = 2, so we evaluate the function M(x | 0, 1, 2, 3). Lift 0, 1, 2, 3 to $\mathbf{R}^3$ by choosing y^0 = (0, 0, 0), y^1 = (1, 0, 2), y^2 = (2, 1, 0) and y^3 = (3, 0, 0), (see figure 5.3), then $M(x_0 \mid 0, 1, 2, 3) = A/V$ where A is the area of the intersection of the plane $x = x_0$ with the simplex $\sigma = [y^0, y^1, y^2, y^3]$, and V is the volume of σ. That is,

$$M(x|\ 0,1,2,3) = A/V.$$

where

V = (1/3) (base area of triangular pyramid) (height)

= (1/3) (3/2) (2)

= 1 cubic unit.

Table 5.1 gives the values of M(x | 0, 1, 2, 3) at the different values of x.

Table 5.1

The values of M(x | 0, 1, 2, 3) at the different values of x

x	M	x	M	x	M
0	0	6/5	33/50	13/6	25/72
1/4	1/32	3/2	3/4	5/2	1/8
1	1/2	2	1/2	3	0

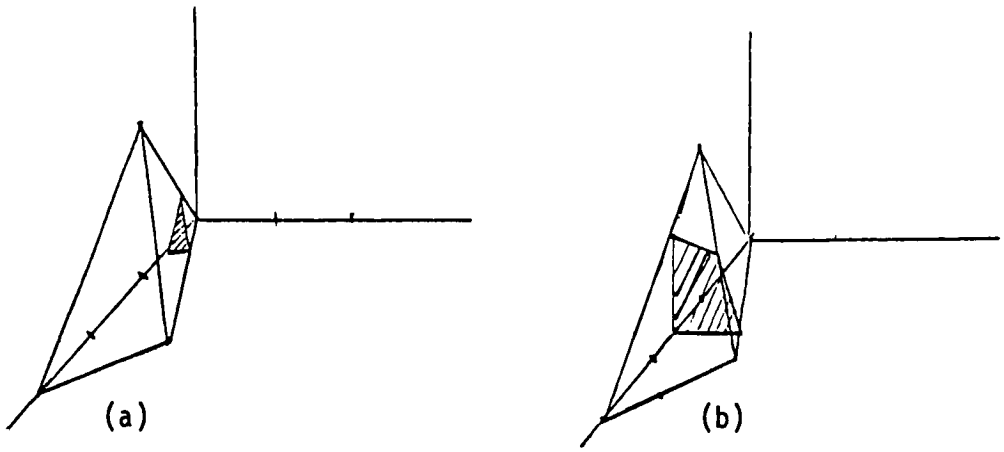


Figure 5.3 Geometric construction of M(x | 0, 1, 2, 3)

5.2 Multivariate B-Splines

The equation of Curry and Schoenberg

$$M(x \mid t_0, t_1, \ldots, t_n) = \frac{\mathrm{Vol}_{n-1} \{ y \in \sigma \mid y_1 = x \}}{\mathrm{Vol}_n \sigma}$$

led to the following multivariate version of this function, (de Boor, [1976]). de Boor generalized the geometric construction (5.8) to higher dimensions to define multivariate B-splines.

Let $x^0, x^1, \ldots, x^n$ be any point in R^s, $n \geq s + 1$, such that there is a proper simplex $\sigma = [\, y^0, y^1, \ldots, y^n \,]$, $(\mathrm{Vol}_n \sigma > 0)$, with vertices $y^0, y^1, \ldots, y^n$, having the property that

(5.9) $y^0 = (x^0, \ldots), \quad y^1 = (x^1, \ldots), \quad \ldots\ldots, \quad y^n = (x^n, \ldots).$

Equivalently, the dimension of the convex hull of $x^0, x^1, \ldots, x^n$ is

s. then the unnormalized s^{th}-variate B-spline of degree $k = n - s$ is defined by

(5.10) $M(x \mid x^0, x^1, \ldots, x^n)$

$$= \begin{cases} \dfrac{\chi(x \mid \sigma)}{\mathrm{Vol}_n \sigma} & \text{if } n = s \\ \dfrac{\mathrm{Vol}_{n-s}\{ y \in \sigma \mid y_j = x_j, j = 1, \ldots, s \}}{\mathrm{Vol}_n \sigma} & \text{if } n > s \end{cases}.$$

where $x = (x_1, x_2, \ldots, x_s) \in \mathbf{R}^s$. We shall notice that M defined by (5.10) is independent of the simplex σ, subject only to the condition (5.9).

Let us give an immediate example of this definition. For $s = 2$ and $n = 2$, the order 1 or degree 0 B-spline is defined below:

(5.11) $M(x \mid x^0, x^1, x^2)$

$$= \begin{cases} \dfrac{1}{\mathrm{Vol}_2 \sigma} = \dfrac{1}{\text{area of } \sigma} & \text{if } x \in \sigma = [x^0, x^1, x^2] \\ 0, \text{otherwise} \end{cases}$$

where x^0, x^1, x^2 is any three non-collinear points in the plane. In general M is nonnegative and zero outside of $\sigma = [x^0, x^1, x^2]$.

5.3 A Recurrence relation for the Multivariate B-Spline

On the last section, we define multivariate B-splines through a geometric construction. Although the geometric definition (5.10) is quite good conceptually. However, it is dreadful from a practical standpoint. Fortunately, a recursive definition having the univarite construction (2.3) as a special case was discoved independently by both Dehmen [1980] and Micchelli [1980] :

For $n > s + 1$ and $\mathrm{Vol}_s\, \sigma > 0$

$$(5.12)\quad M(x \mid x^0, x^1, \ldots, x^n) = \frac{n}{n-s} \sum_{j=0}^{n} \lambda_j\, M(x \mid x^0, \ldots, x^{j-1}, x^{j+1}, \ldots, x^n)$$

whenever

$$(5.13)\quad x = \sum_{j=0}^{n} \lambda_j x^j \qquad \text{and} \qquad \sum_{j=0}^{n} \lambda_j = 1.$$

Since the number of the vectors on the right side of the above equation (5.12) is n, one less than on the left-hand side, this formula can be used recursively to compute multivariate B-splines. Note that (5.12) actually comprises a variety of formulas, since the weights λ_j are in general not uniquely defined by (5.13). Danmen and Micchelli [83] or Kochevar [84] suggested

that the apparent choice for weights is to pick any s + 1 affinely independent knots $_{x}{}^{i}0, \ldots, {}_{x}{}^{i}s$ from the knot set $K = \{ x^0, x^1, \ldots, x^n \}$ (which is possible since $\mathrm{Vol}_s K > 0$) and use barycentric coordinates

$$(5.14)\quad \lambda_{i_j} = \lambda_{i_j}(x \mid {}_{x}{}^{i}0, \ldots, {}_{x}{}^{i}s) = \frac{\det(x^{i_0}, \ldots, x^{i_{j-1}}, x^{i_{j+1}}, \ldots, x^{i_s})}{\det(x^{i_0}, \ldots, x^{i_s})}$$

of x with respect to $_{x}{}^{i}0, \ldots, {}_{x}{}^{i}s$. Here we have used the short-hand notation

$$\det(x^0, \ldots, x^s) = \begin{vmatrix} x_1^0 & x_1^1 & \ldots & x_1^s \\ \ldots & \ldots & \ldots & \ldots \\ x_s^0 & x_s^1 & \ldots & x_s^s \\ 1 & 1 & \ldots & 1 \end{vmatrix}$$

We set $\lambda_m = 0$ for $m \neq i_j$ so that (5.13) is clearly satisfied. Thus, independent of n, we see that at most s+1 summands occur in the right hand side of (5.12).

Now, we may modify the definition (5.10) to an efficient and practical algorithm for the computation of $M(x \mid x^0, \ldots, x^s)$ at any point $x \in \mathbf{R}^s$. For n = s, we have

$$(5.15a)\quad M(x \mid x^0, \ldots, x^s) = \frac{\chi(x \mid [x^0, \ldots, x^s])}{\mathrm{Vol}_s([x^0, \ldots, x^s])}$$

for $n > s$, we have

$$(5.15b)\quad M(x \mid x^0, \ldots, x^n) = \frac{n}{n-s} \sum_{j=0}^{s} \lambda_{i_j} M(x \mid x^{i_0}, \ldots, x^{i_{j-1}}, x^{i_{j+1}}, \ldots, x^{i_s})$$

Since the structure of such an algorithm is typical for evaluating various other functionals of B-splines, it is worthwhile to illustrate it with the following two-dimensional examples $n = 3$, $s = 2$, or degree 1.

In the first example, we consider four points $x^0 = (0, 0)$, $x^1 = (2, -1)$, $x^2 = (3, 0)$, and $x^3 = (1, 2)$ in the plane, (see figure 5.4). We have several options for the choice of affinely independent knots. Choosing x^0, x^1, x^2, (5.15b) gives

$$(5.16)\quad M(x \mid x^0, x^1, x^2, x^3) = (-x_1 + x_2 + 3)\, M(x \mid x^1, x^2, x^3) - 3\, x_2\, M(x \mid x^0, x^2, x^3) + (x_1 + 2\, x_2)\, M(x \mid x^0, x^1, x^3)$$

where each B-spline on the right hand side is a piecewise constant given by (5.11) or (5.15a). In particular, for example, a point x in figure 5.4 is chosen to be in $[x^1, x^2, x^3]$, then only

two nonzero summands appear in (5.16). That is,

$$M(\mathbf{x} \mid \mathbf{x}^0, \mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3) = (-x_1 + x_2 + 3)\, M(\mathbf{x} \mid \mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3) - 3\, x_2\, M(\mathbf{x} \mid \mathbf{x}^0, \mathbf{x}^2, \mathbf{x}^3)$$

Some remarks we shall mention here. First, there is a practical complication which occurs when the evaluation point $\mathbf{x}$ lies on a line segment connnecting two knots (say, point $\mathbf{y}$ in figure 5.4). A simple continuity argument shows that $M(\mathbf{y} \mid \mathbf{x}^0, \mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3)$ will be miscalculated if $\mathbf{y}$ is attributed to both sets $[\mathbf{x}^0, \mathbf{x}^1, \mathbf{x}^2]$ and $[\mathbf{x}^0, \mathbf{x}^2, \mathbf{x}^3]$. This difficulty can be overcome by redefined the initial B-spline (5.15a) as $1/((v+1)\ \mathrm{Vol}_s([\mathbf{x}^0, \mathbf{x}^1, \mathbf{x}^2]))$ where v is the number of barycentric coordinates vanishing at $\mathbf{y}$.

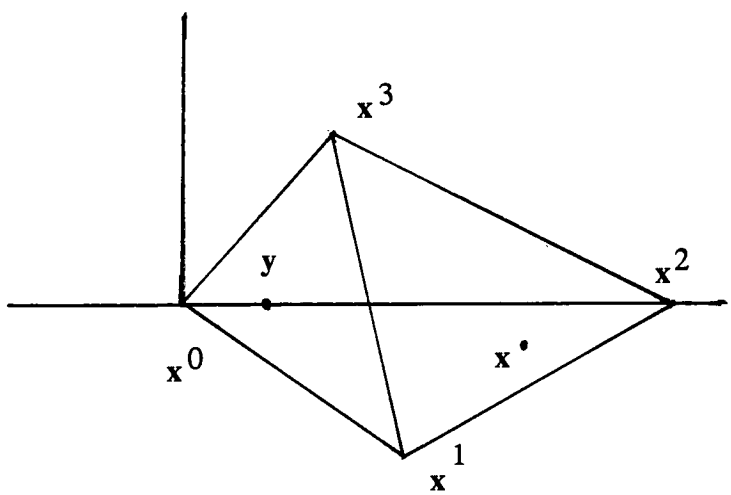


Figure 5.4 Noncollinear knot region for $M(\mathbf{x} \mid \mathbf{x}^0, \mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3)$

Second, suppose the point $\mathbf{x}^2$ has been changed to be $\mathbf{x}^2 = (5/3, 0)$, or three points $\mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3$ are collinear (see figure 5.5). Then $M(\mathbf{x} | \mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3)$ is a degenerate B-spline so that its value strictly by expression (5.10) or (5.11) is infinite for any $\mathbf{x}$ lying on the line from $\mathbf{x}^1$ to $\mathbf{x}^3$. In this case, we artificially set the coefficient to zero. In other words, the first term makes no contribution to the sum.

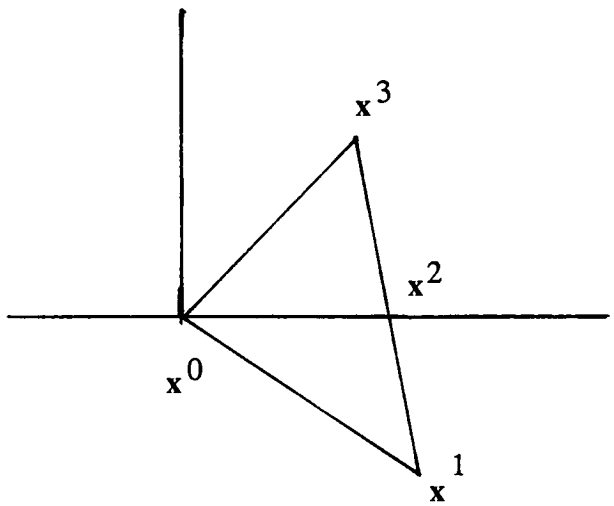


Figure 5.5 Collinear knot region for $M(\mathbf{x} | \mathbf{x}^0, \mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3)$

Chapter 6 Conclusions and Recommendation for Further Research

6.1 Summary

For successful application in computer-aided design, a thorough understanding of the mathematical properties of shape representations is essential. This understanding also helps in the design of interactive systems that permit the designer to refine the model to his satisfaction. Therefore, let us review the properties of a B-spline curve or surface that make it an unusually effective interactive design tool.

First, the curve/surface lies entirely within the convex figure defined by the extreme points of the polygon/polyhedron and generally mimics the gross features of the polygon/polyhedron.

Second, they are variation-diminishing. This means that they never osillate widly away from their defining control points.

Third, they are axis-independent and the parametric formulation allows to represent multiple-valued shapes, such as a closed shape.

Finally, and most importantly, control points influence curve shape or surface in a predictable, natural way, making B-spline a good candidate for use in an interactive environment. This advantage is the same as that of the famous Bezier curves or surfaces in the UNISURF system, used by Renault since 1972 to design the sculptured surface of many of their automobile bodies.

However, it is the local control of the curve shape or surface possible with B-spline that gives the technique an advantage over the Bezier technique, as does the ability to add control points without increasing the degree of the B-spline. The 'local control' means that a small change in the position of a characteristic polygon/ployhedron tends only to locally change the resulting shape; for examples, see figures (3.1), (4.7) and (4.8). The reason B-spline can avoid global propagation of changes is that it uses a special set of blending functions that have only local influence and depend on only a few neighboring control points.

The foremost advantage of using B-spline surfaces to define shapes is that complete models can be created for objects that have both complex and simple shapes. Figure (6.1) shows this advantage. This picture depicts the effect of the blend surface of a bottle transitioning from around top to a nearly square body.

Figure (6.1) is constructed by using algorithm (4.1), called the tensor product surface. This approximation method has most of the desirable features of the univariate scheme, specifically local behavior and positive basis functions summing to one. This technique has enjoyed an immense popularity as a tool in the field of **CAGD**. Unfortunately tensor product schemes also possess some severe limitations with perhaps their most restrictive shortcoming being their dependency on rectilinear knot distributions.

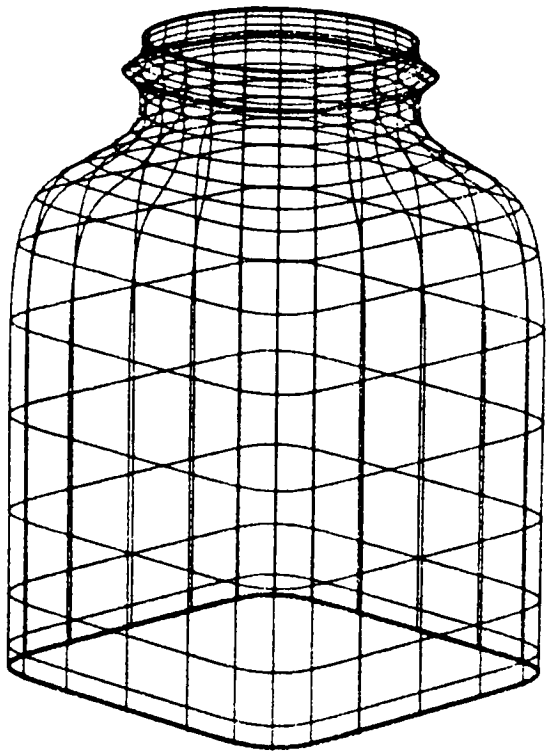


Figure 6.1 B-spline bottle with round top and square body

A knot topology of this configuration does not easily lend itself to modelling a wide variety of objects as can be seen, for examples, in figure (5.1) or in figure (6.2). Both figures show the difficulty in modeling the joinning parts between two B-spline surfaces. In figure (6.2), the two B-spline surfaces, the mug handle and the mug body, are joined artificially. This means that we must find the intersection points and then manipulate the two ending parts of the mug handle so that these two B-spline surfaces match properly.

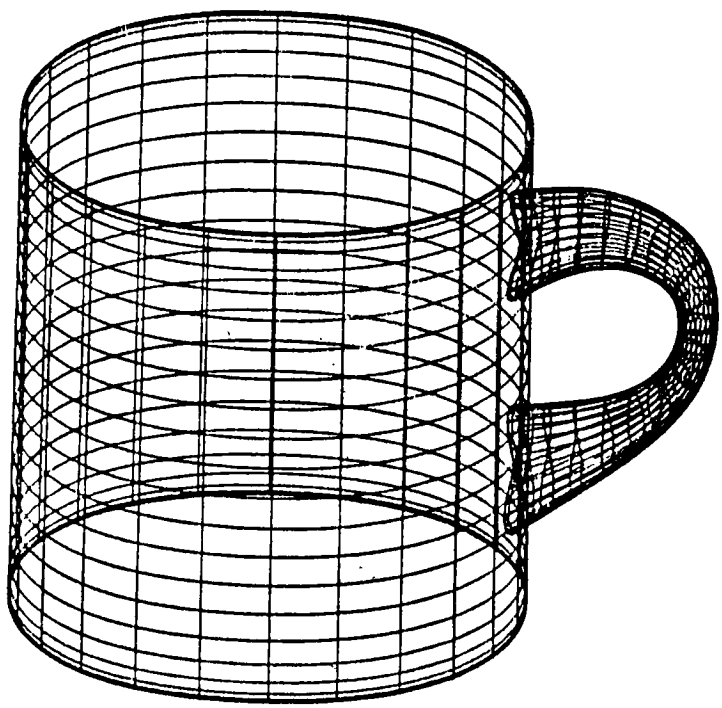


Figure 6.2 B-spline mug

This attempt, aside from the added complexity of the construction, has a tendency to circumvent the desirable qualities which prompted the use of B-splines initially. What is truly needed therefore is an alternate generalization of the univariate B-spline approximation that admits arbitrary knot configurations while still preserving its desirable features. One possible answer is the multivariate B-spline that can be defined for almost any knot configuration giving approximations on any compact polyhedral region in a Euclidean space of any dimension. It has

been demonstrated that the mathematical basis used for most CAGD applications is readily extendable to higher dimensions in principle. But, whether or not this extension will turn out to be a viable alternative in practice is still an open question. This the one direction for the further investagion.

6.2 Recommendation for Further Research

Continued research related to B-splines could concentrate on the following subjects:

(1) Non-uniform B-splines: In this thesis we construct any picture based on uniform knots. Barsky [1984] introduced a generalization of the uniform cubic B-spline called the beta-spline.

(2) Shading: B-spline representation can provide exact calculation of the direction that is perpendicular to a surface at any point. This information is used to calculate how light is reflected by the surface in order to make realistically shaded pictures that represent the modeled object accurately. Because shaded pictures can be used by tooling and fixturing designs. Thus, such pictures can be used to detect incorrect shapes or shapes that do not mate properly with others.

(3) Numerical control machine tool: There is one possible way to product such as fillet surface part, see figure (1.4), on page 6. That is, with providing accurate surface normal

information, to locate accurately position the tool center so that the cutting edge removes material in a direction parallel to the desired surface in order to best approximate the surface shape in metal.

(4) In order to solve the joining parts between two B-spline surfaces, there are two approaches. The first one is followed.

Intersection algorithm: The intersection algorithm locates every intersection point betwwen two surfaces. This capability is particularly useful for defining surfaces that have extremely arbitrary shapes such as a sculptured surface shown in the figure (1.6) on page 7. Although the intersection of two B-spline surfaces is well understood theoretically, however, implementing the algorithm for use in a modeling system is a demanding programming task.

The second approach is the multivariate B-spline approximation. As mentioned above, there is still a gap in complexity betwwen in theory and in practice, there is still a gap in complexity between one and higher dimensions. Hopefully the apparent advantages of multivariate B-spline approximation over current multidimensional methods will more than make up for it.

References

Barnhill, R. E.,(1985), Surfaces in Computer Aided Geometric Design: A Survey with New Results, Computer Aided Geometric Design ,Vol. 2 ,No. 1, Sept. 1985, pp 1-17.

Barnhill, R. E., and Boehm, W., (1983), Surfaces in Computer Aided Geometric Design, North-Holland, New York.

Barnhill, R. E., and Riesenfeld, R. F., (1974), Computer Aided Geometric Design, Academic Press, New York.

Barsky, B. A., (1984), A Description and Evaluation of Various 3-D Models. IEEE Computer Graphics and Applications, Vol. 4, No.1, Jan. 1984, pp 38-52.

Bohm, W., Farin, G., and Kahmann, J., (1984), A Survey of Curve and Surface Methods in CAGD, Computer Aided Geometric Design, Vol.1, No.1,July 1984, pp 1-60.

Casale, M. S. and Stanton, E. L., (1985), An Overview of Analytic Solid Modeling, IEEE Computer Graphics and Applications, Vol. 5, No. 2, Feb. 1985, pp 45-56.

Cheney, W. and Kincaid, D., (1985), Numerical Mathematics and Computing, 2nd ed., Brooks/Cole, Calif..

Curry, H.B. and Schoenberg, I.J., (1966), On Polya Frequency Functions IV: the Fundamental Spline Functions and Their Limits, J. d'Analyse Mathematique 17(1966), pp 71-107.

Dahmen, W., (1980), On Multivariate B-Splines, SIAM J. Numer. Anal. 17(1980), pp 179-191.

Dahmen, D., (1981), Approximation by Linear Combinations of Multivariate B-Splines, J. Approximation Theory, 31(1981), pp 299-324.

Dahmen, W. and Micchelli, C. A., (1983) Multivariate Splines - A New Constructive Approach, In Surfaces in Computer Aided Geometric Design, edited by R. E. Barnhill and W. Boehm, pp191-215. North-Holland, New York.

de Boor, C., (1976), Splines as Linear Combinations of B-Splines, in Approximation Theory II, G.G. Lorentz, C.K. Chui, and L.L. Schumaker, pp 1-47, Academic Press, New York, 1976.

de Boor, C. (1978), A Practical Guild to Spline, Springer-Verlag, New York.

Hakopian, H., (1982), Multivariate Spline Functions, B-Spline Basis and Polynomial Interpolations, SIAM J. Numer. Anal. 19(1982), pp 510-517.

Hanna, S. L., Abel, J. F. and Greenberg, D. P., (1983), Intersection of Parametric Surfaces by Means of Look-up Tables, IEEE computer Graphics and Applications, Vol. 3, No. 7, Oct. 1983, pp 39-48.

Hollig, K., (1982), Multivariate Splines, SIAM J. Numer. Anal., 19(1982), pp 1013-1031.

Knapp,L. C., (1984), Surface Definition Techniques for Solid Modeling, Computer-Aided Design, Engineering, and Drafting, Auerbach.

Kochevar, P., (1984), An Application of Multivariate B-Splines to Computer-Aided Geometric Design, Rocky Mountain J. of Math., Winter 1984, pp 159-175.

Lee, R. B. and Fredricks, D. A.,(1984), Intersection of Parametric Surfaces and a Plane, IEEE computer Graphics and Applications, Vol. 4, No. 8, Aug. 1984, pp 48-51.

Micchelli, C.A., (1979), On a Numerically Efficient Method for Computing Multivariate B-Splines , in Multivariate Approximation Theory, W. Schempp, K. Zeller, eds., Birkhauser, Basel, (1979), pp 211-248.

Micchelli, C.A., (1980), A Constructive Approach to Kergin Interpolation in R^k: Multivariate B-Splines and Lagrange, Rocky Mountains J. Math. 10, (1980), pp 485-497.
Mortenson, M. E., (1985), Geometric Modeling, Wiley, New York.

Phillips M. B. and Odell, G. M., (1984), An Algorithm for Locating and Displaying the intersection of two Arbitrary Surfaces, IEEE computer Graphics and Applications, Vol. 4, No. 9, Sept. 1984, pp 48-58.

Prautzsch, H.,(1984), Short proof of the Oslo algorithm, Computer Aided Geometric Design, Vol.1, No.1, July 1984, pp 95-96.

Riesenfeld, R. F., (1973), Applications of B-spline Approximation to Geometric Problems of Computer Aided Design, Ph.D Thesis, Syracuse Univ., (1973).

Schoenberg, I. J., (1967). On Spline Function, In Inequalities, edited by O. Shisha, pp. 255-291. Academic Press, New York.

Tiller, W., (1983) Rational B-Splines for Curve and Surface Representation, IEEE Computer Graphics and Applications, Vol. 3, No. 6, Sept. 1983, pp 61-69.