A STUDY OF THE PARAMETRIC UNIFORM B-SPLINE CURVE AND SURFACE REPRESENTATIONS

by

Brian A. Barsky

Berkeley Computer Graphics Laboratory
Computer Science Division
University of California
Berkeley, California 94720
U.S.A.

Technical Report No. UCB/CSD 83/118

May 1983

A Study of the Parametric Uniform B-spline Curve and Surface Representations

Brian A. Barsky

ABSTRACT

This paper explains the parametric uniform B-spline curve and surface representations. The parametric representation is discussed, the properties of the B-spline representation are described, and a detailed derivation of the B-spline basis functions is presented. Various end conditions and boundary conditions are described in order to enable the B-spline user to select which of the many options would be appropriate for a particular application. Efficient algorithms are designed and analyzed for B-spline basis function evaluation, and for the evaluation and perturbation of both B-spline curves and surfaces. Finally, difference techniques to accomplish this are also developed.

June 7, 1983

Table of Contents

List of Figures

List of Tables

I. INTRODUCTION

The B-spline curve and surface representations provide very appropriate mathematical models for computer implementation. Although Schoenberg [14, 15, 24] developed the mathematical theory of spline approximation in 1946, it was not until 1973 that Riesenfeld [18, 21] applied B-splines to computer aided geometric design (CAGD). B-splines have since been widely used and generally accepted as a standard tool and method for the design and modeling of free-form curves and surfaces, and have become prevalent in many geometry packages and turnkey geometry systems.

By far the most ubiquitous of the many possible versions of B-splines available is the cubic B-spline curve and bicubic B-spline surface with unrepeated uniformly spaced parametric knot values [9, 10, 20]. The cubic/bicubic B-spline is usually chosen because it is the lowest degree B-spline which yields shapes which are sufficiently "smooth" for most applications. The uniform knot spacing is usually used because of the simplicity in concept and implementation, its concise matrix formulation, and its amenability to implementation as an efficient, precomputed "table lookup" algorithm capable of realtime execution. This paper gives a detailed presentation of the methods and computational exploitations which are possible for this special yet very important case.

II. THE PARAMETRIC REPRESENTATION

II.1. Motivation and properties

The conventional scalar-valued, explicit, functional form in the Cartesian coordinate system,

$y = f(x)$ and

$z = f(x, y)$

is only capable of describing a small class of curves and surfaces. For example, such a function cannot represent a nonplanar curve twisted in space. Furthermore, it cannot be used to describe a multiple-valued curve or surface; that is, a curve having more than one value of y for a single value of x, or a surface with more than one value of z for a single (x, y) pair. Moreover, there is nothing inherently special about either x or y or z; thus, it is unnatural to single out one of the variables to be an independent variable, leaving the others to assume the role of dependent variables. The choice of coordinate system should have no effect on the shape of the curve or surface, because shape is an attribute of an object dependent solely on the intrinsic relation between the points on the curve or surface. The shape should be independent of the orientation of the coordinate axes and should not change if a different coordinate system is employed. All these requirements can be satisfied with the parametric representation [17].

II.2. Explanation

The parametric representation of a curve has each component expressed as a separate univariate (single parameter) function while that of a surface has each component defined by a separate bivariate (two parameter) function. The coordinates of a point can be written as a row vector as follows:

[X(u) Y(u)] for a curve in Euclidean two-space,

[X(u) Y(u) Z(u)] for a curve in Euclidean three-space, and

[X(u,v) Y(u,v) Z(u,v)] for a surface in Euclidean three-space.

For notational convenience, the row vectors for a curve and surface will be denoted as $\underline{Q}(u)$ and $\underline{Q}(u,v)$, respectively.

A parametric derivative, with respect to some parameter or parameters, can also be represented as a row vector. Each component is the derivative, with respect to that parameter or parameters, of the function corresponding to its coordinate. These parametric derivative vectors are then:

$$\frac{d^a}{du^a}\underline{Q}(u) = \left[\frac{d^a}{du^a}X(u) \quad \frac{d^a}{du^a}Y(u)\right]$$

for a curve in Euclidean two-space,

$$\frac{d^a}{du^a}\underline{Q}(u) = \left[\frac{d^a}{du^a}X(u) \quad \frac{d^a}{du^a}Y(u) \quad \frac{d^a}{du^a}Z(u)\right]$$

for a curve in Euclidean three-space, and

$$\frac{\partial^{a+b}}{\partial u^a \partial v^b}\underline{Q}(u,v) = \left[\frac{\partial^{a+b}}{\partial u^a \partial v^b}X(u,v) \quad \frac{\partial^{a+b}}{\partial u^a \partial v^b}Y(u,v) \quad \frac{\partial^{a+b}}{\partial u^a \partial v^b}Z(u,v)\right]$$

for a surface in Euclidean three-space.

For notational convenience define

$$\underline{Q}^{(a)}(c) = \frac{d^a}{du^a}\underline{Q}(u)\Big|_{u=c}$$

$$\underline{Q}^{(a,b)}(c,d) = \frac{\partial^{a+b}}{\partial u^a \partial v^b}\underline{Q}(u,v)\Big|_{u=c,\ v=d} \tag{II.1}$$

This parametric representation can be conceptualized as a mapping from parameter space to Euclidean space. For a given parametric value, it yields the coordinates of a point on the curve or surface. As the parametric value varies, the curve or surface is traced out. This mapping is illustrated for each of the three cases in Figures II-1, II-2, and II-3, respectively.

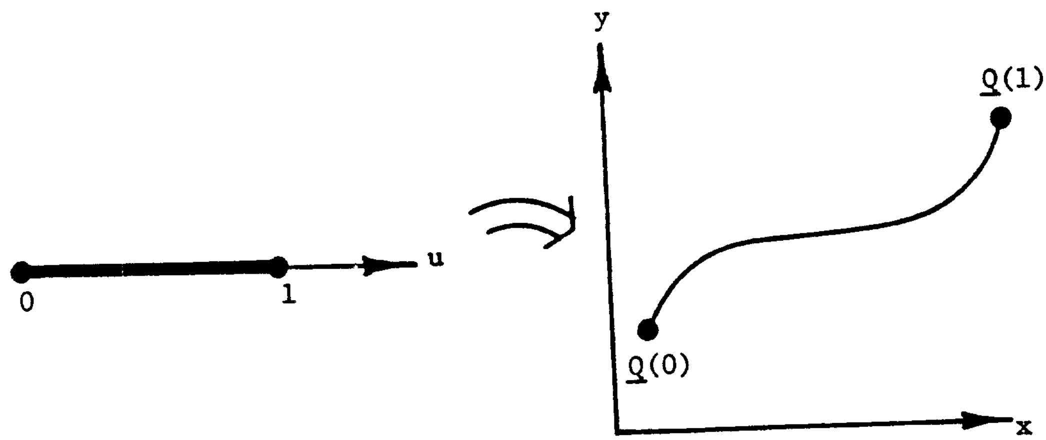


Figure II-1: Mapping from u parameter space to Euclidean two-space for a planar curve.

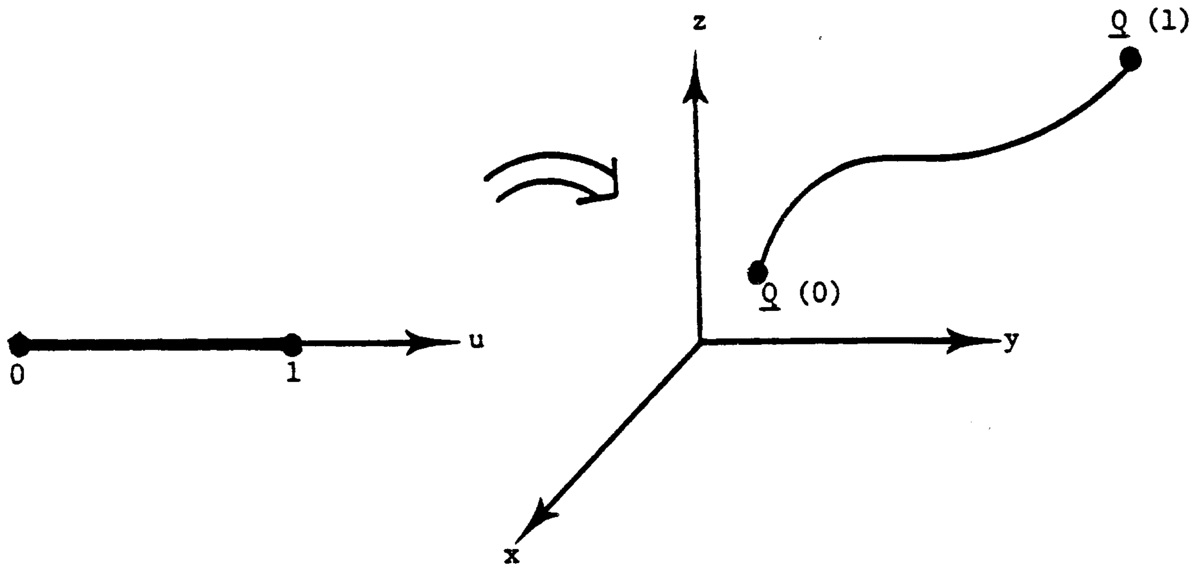


Figure II-2: Mapping from u parameter space to Euclidean three-space for a space curve.

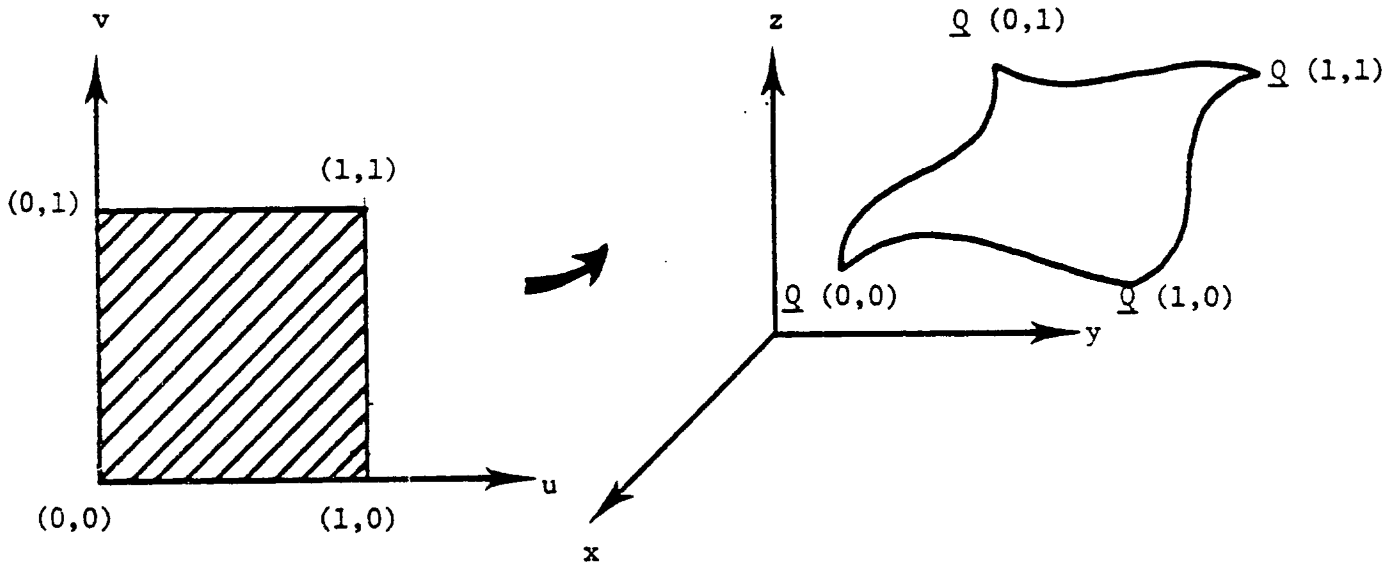


Figure II-3: Mapping from (u,v) parameter space to Euclidean three-spac for a surface.

III. OVERVIEW AND PROPERTIES OF THE B-SPLINE CURVE AND SURFACE REPRESENTATIONS

III.1. Control vertices

A B-spline curve or surface is specified by a set of points called <u>control vertices</u>. Although these vertices do not generally lie on the generated curve or surface, their positions completely determine its shape. The vertices for a curve are an ordered sequence and are connected in succession to form (a closed or an open) <u>control polygon</u>. Figure III-1 is an image from the curve representation system described in [5, 4] which shows a B-spline control polygon and the curve which it defines. For a surface, the vertices are organized as a two dimensional graph with a rectangular topology, which will be referred to as a <u>control graph</u> (see Section IX.1 for explanation). Figure III-2 is a photograph taken from the surface representation system described in [3] showing a B-spline control graph with its generated surface.

The generated curve or surface tends to mimic the overall shape of the control polygon or graph, and the manipulation of a control vertex causes a predictable modification in the resulting shape.

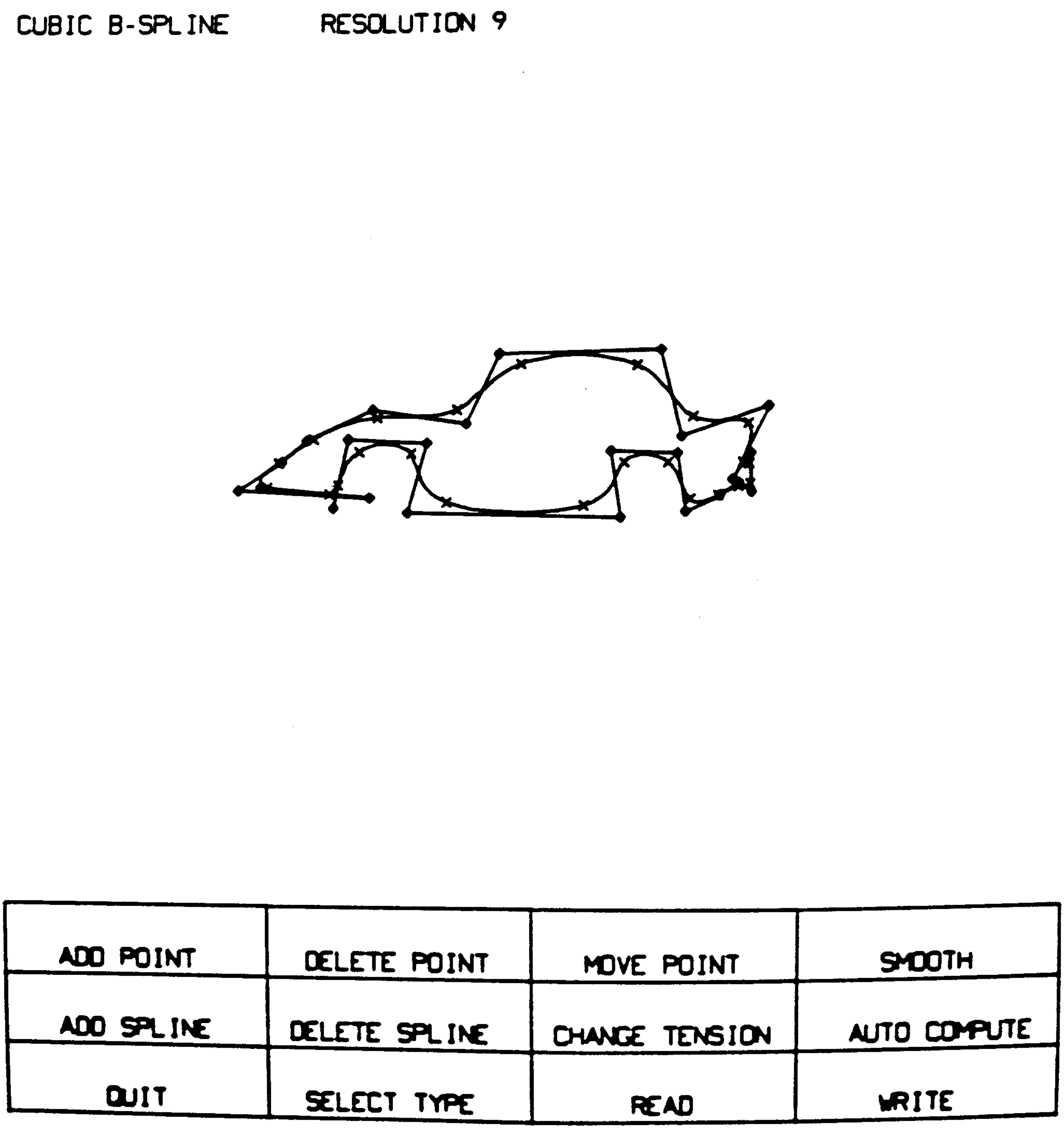


Figure III-1: A B-spline control polygon with its generated curve.

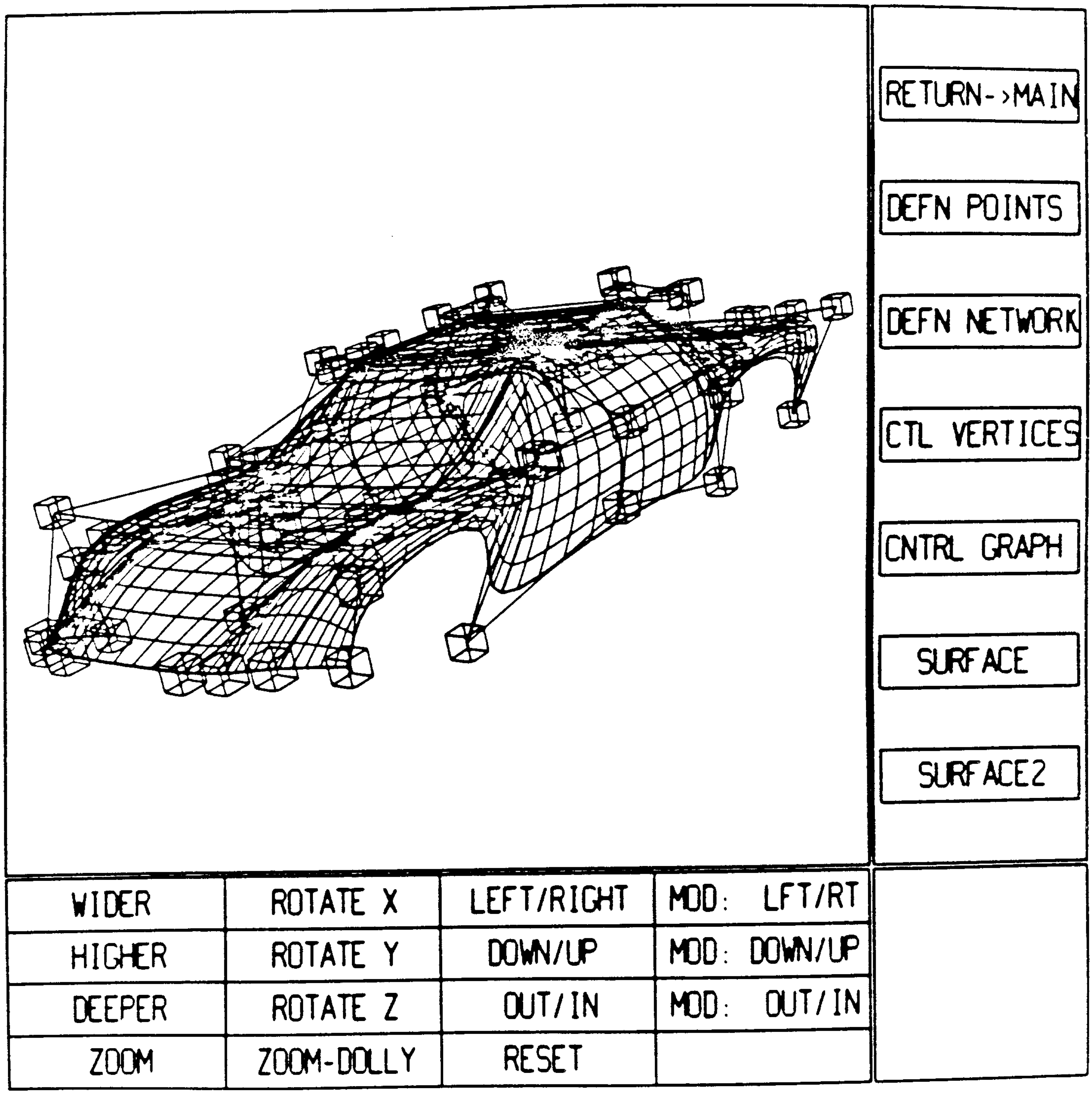


Figure III-2: A B-spline control graph with its generated surface.

III.2. Piecewise representation

While an entire curve or surface is not easily defined by a single analytic function, it can be apportioned into a set of smaller pieces, each described by a separate analytic function, to form a piecewise representation. A B-spline curve of degree d (order d+1) is composed of a sequence of polynomials of degree not exceeding d, called B-spline curve segments (Figure III-3). A bipolynomial B-spline surface of degree d and e in each parameter is a mosaic of surface patches, each of which is a bipolynomial of degree not exceeding d and e in each parameter, respectively (Figure III-4).

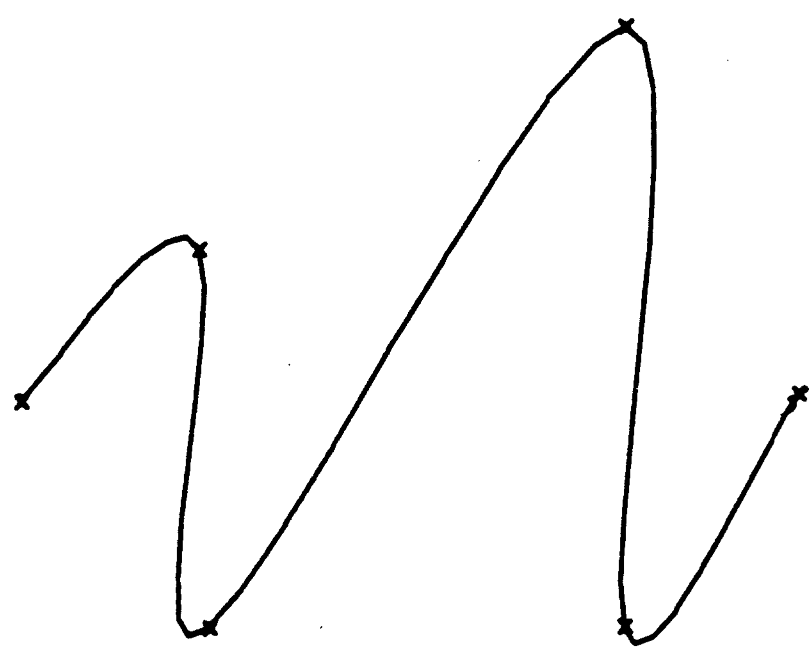


Figure III-3: B-spline curve is composed of a sequence of curve segment

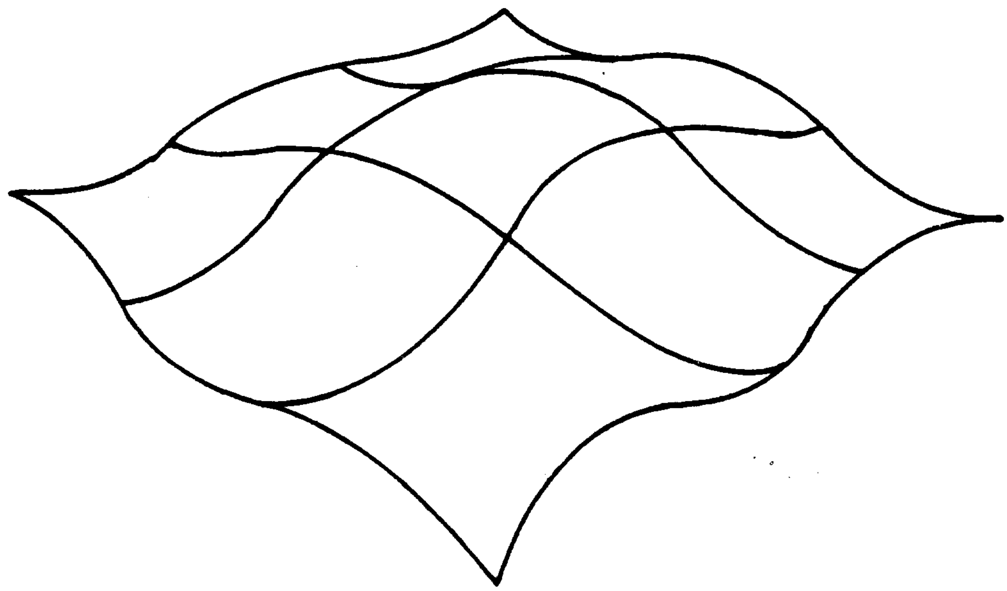


Figure III-4: A B-spline surface is a mosaic of surface patches.

III.3. Local control

The B-spline basis is a local basis; that is, each B-spline basis function has <u>local support</u> (nonzero over a minimal number of spans). Since each control vertex is associated with a basis function, it influences only a local portion of the curve or surface and has no effect on the remainder of it. The effect of moving a single control vertex, then, is localized to a predetermined portion of the curve or surface. This enables the user of the B-spline representation to have precise control over the resulting shape by moving one control vertex at a time. This action consequently modifies only a local portion without the undesired side effect of disturbing the other portions. Moreover, since only part of the curve or surface is affected, only that part need be recomputed. This is much more computationally efficient then what is required to modify global formulations where any change necessitates the recomputation of the entire curve or surface.

The B-spline formulation exploits the piecewise representation, in order to achieve local control, by defining each piece in terms of only a few nearby vertices. For B-spline curves of degree d (order d+1), each curve segment is controlled by only d+1 of the control vertices and is completely unaffected by all the other control vertices. Equivalently, a given control vertex influences only d+1 curve segments and has no effect whatsoever on the remaining segments. This means that the effects of moving a control vertex are confined to d+1 segments. Figure III-5 illustrates that the effects of moving a cubic B-spline control vertex are confined to four segments.

A bipolynomial B-spline surface of degree d and e in each parameter, respectively, has each surface patch controlled by (d+1)(e+1) control vertices and is unaffected by all other control vertices. Again, this is equivalent to the fact that a given control vertex exerts influence over only (d+1)(e+1) surface patches and has no effect on the remaining patches. Thus, the effects of manipulating one control vertex are limited to (d+1)(e+1) patches.

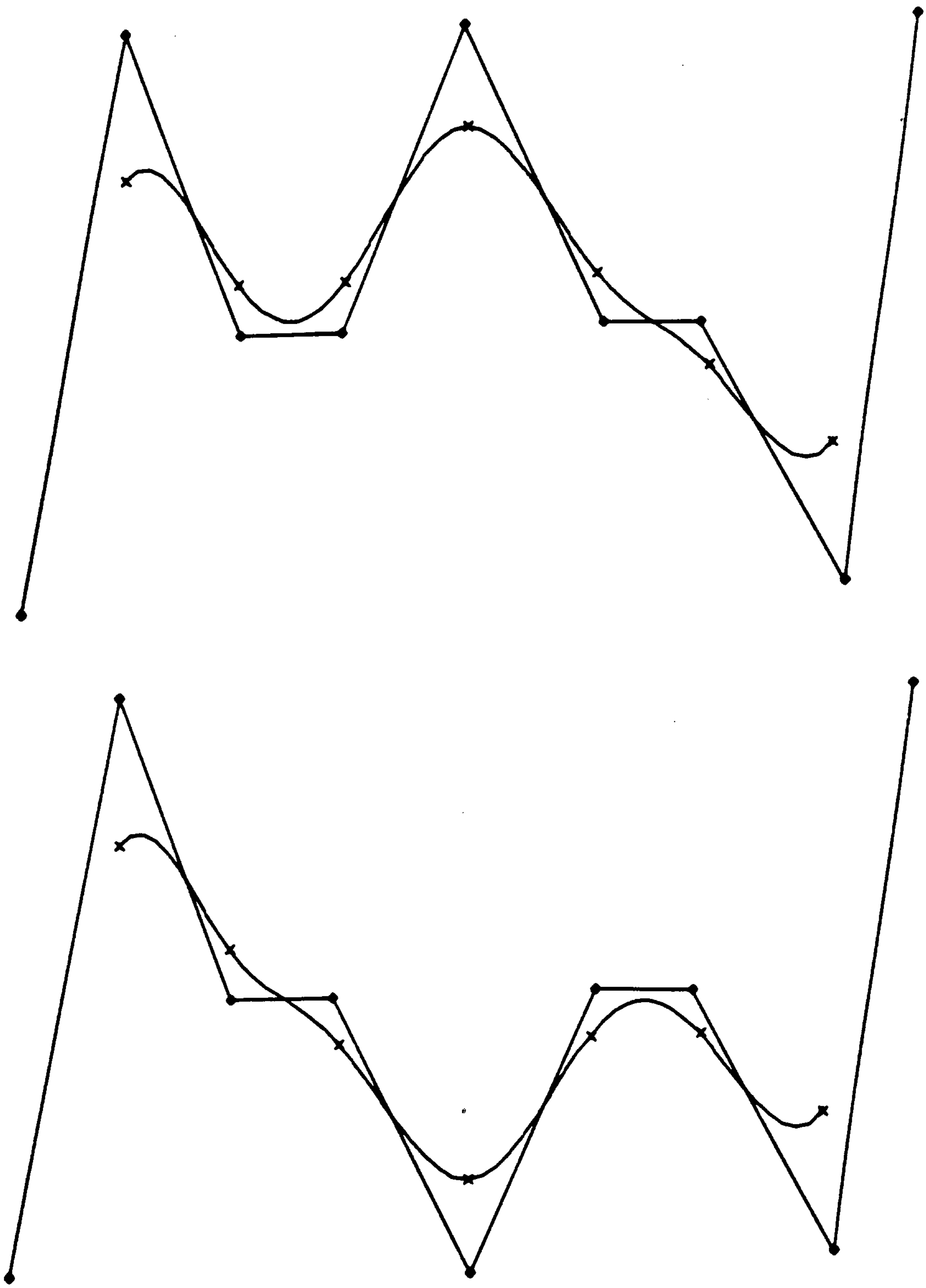


Figure III-5: The effects of moving a cubic B-spline control vertex are confined to four segments.

III.4. Continuity

It is desirable that the joint where curve segments meet or the border between adjacent surface patches be smooth. Many applications require continuity not only of position and the first derivative, but of the second derivative as well. That is, there should not be any "jumps" in position or in first or second derivatives. Clearly, a discontinuity in position or first derivative is quite perceptible. What is not so obvious, however, is that the same is true of the second derivative as well. For example, a circular arc joined to a straight line can meet with first derivative continuity, but the joint will be very noticeable nonetheless. Even in less contrived examples, lack of second derivative continuity will yield flatter, less full shapes and continuous smooth shading of such surfaces can have perceptible discontinuities in colour. Furthermore, surface representations with second derivative continuity are desirable for some higher order shell finite element formulations [26].

B-spline curve segments of degree d (order d+1) join with continuity of the parametric first d-1 derivative vectors. Since each curve segment is a polynomial, it is analytic; therefore, an entire B-spline curve of degree d is everywhere continuous along with its parametric first d-1 derivative vectors.

A bipolynomial B-spline surface of degree d and e in each parameter, respectively, is thus continuous along with its parametric first d-1 derivative vectors in one parametric direction, and with its parametric first e-1 derivative vectors in the other.

III.5. Order

The order of the B-spline representation is independent of the number of control vertices. Although the order must be sufficiently high to offer enough freedom to satisfy various constraints, it is desirable to maintain the order as low as possible. This inhibits the tendency of oscillation and increases computational efficiency. Note that the order is defined to be one more than the degree; that is, order = degree + 1.

III.6. Variation-diminishing property

The B-spline curve representation possesses the variation-diminishing property [19, 25]. Although the generated curve reflects the shape of the control polygon, it does so in a much smoother fashion with less undulations. More precisely, for a planar B-spline curve, any arbitrary straight line crosses the curve no more often than it intersects the control polygon (Figure III-6).

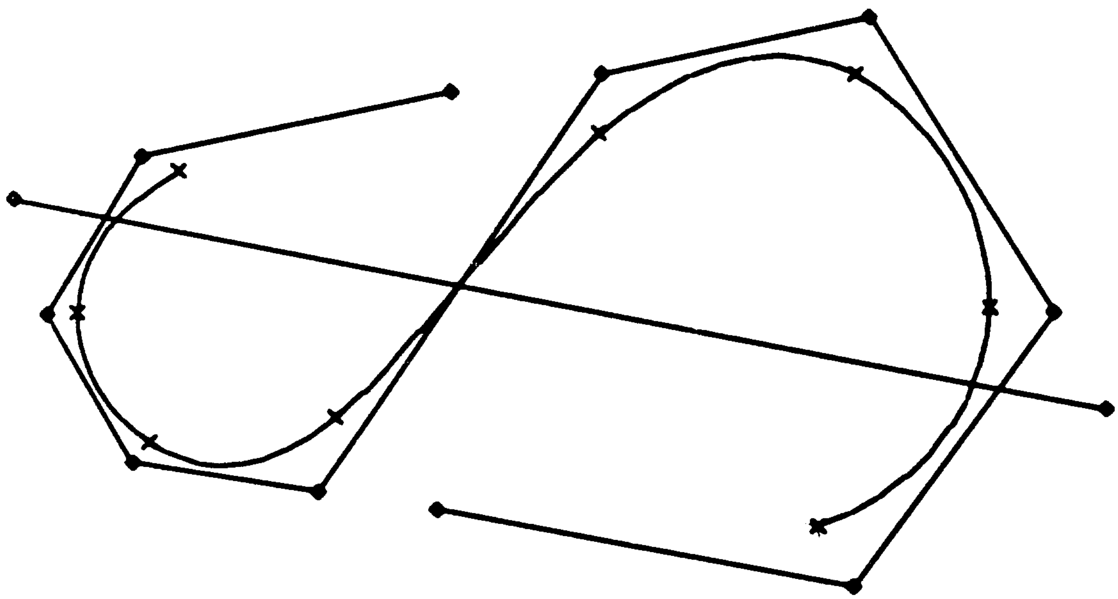


Figure III-6: Variation-diminishing property.

Any arbitrary straight line crosses a planar B-spline curve no more often than it intersects the control polygon.

III.7. Convex hull property

A point on a B-spline curve of degree d (order d+1) is a convex combination of d+1 control vertices. The set of all possible convex combinations of these vertices is their convex hull; that is, the "smallest" convex set which contains these vertices. Thus, each order d+1 B-spline curve segment lies within the convex hull of its d+1 defining control vertices. Figure III-7 shows the convex hull of the four control vertices $\underline{V}_{i+r}$, r = -2, -1, 0, 1, which will contain the i-th cubic B-spline segment.

Since an order d+1 B-spline curve is composed of a sequence of such segments, it follows that the entire B-spline curve will pass through the union of the convex hulls of each successive set of d+1 control

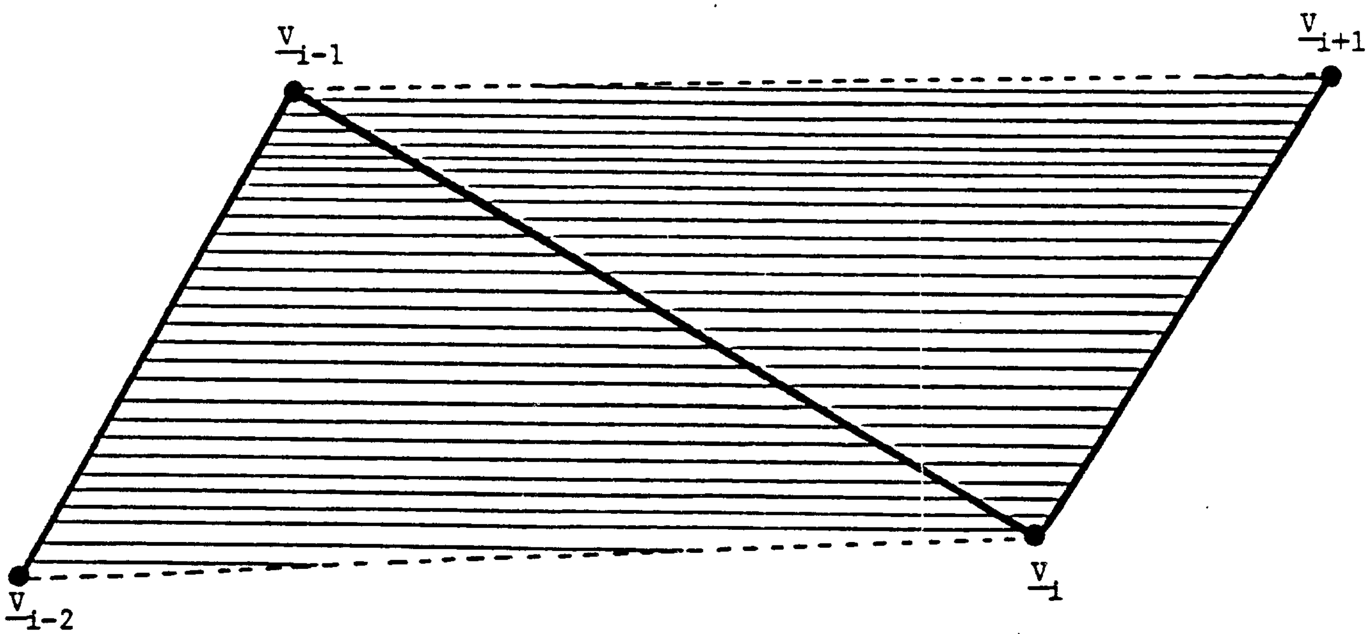
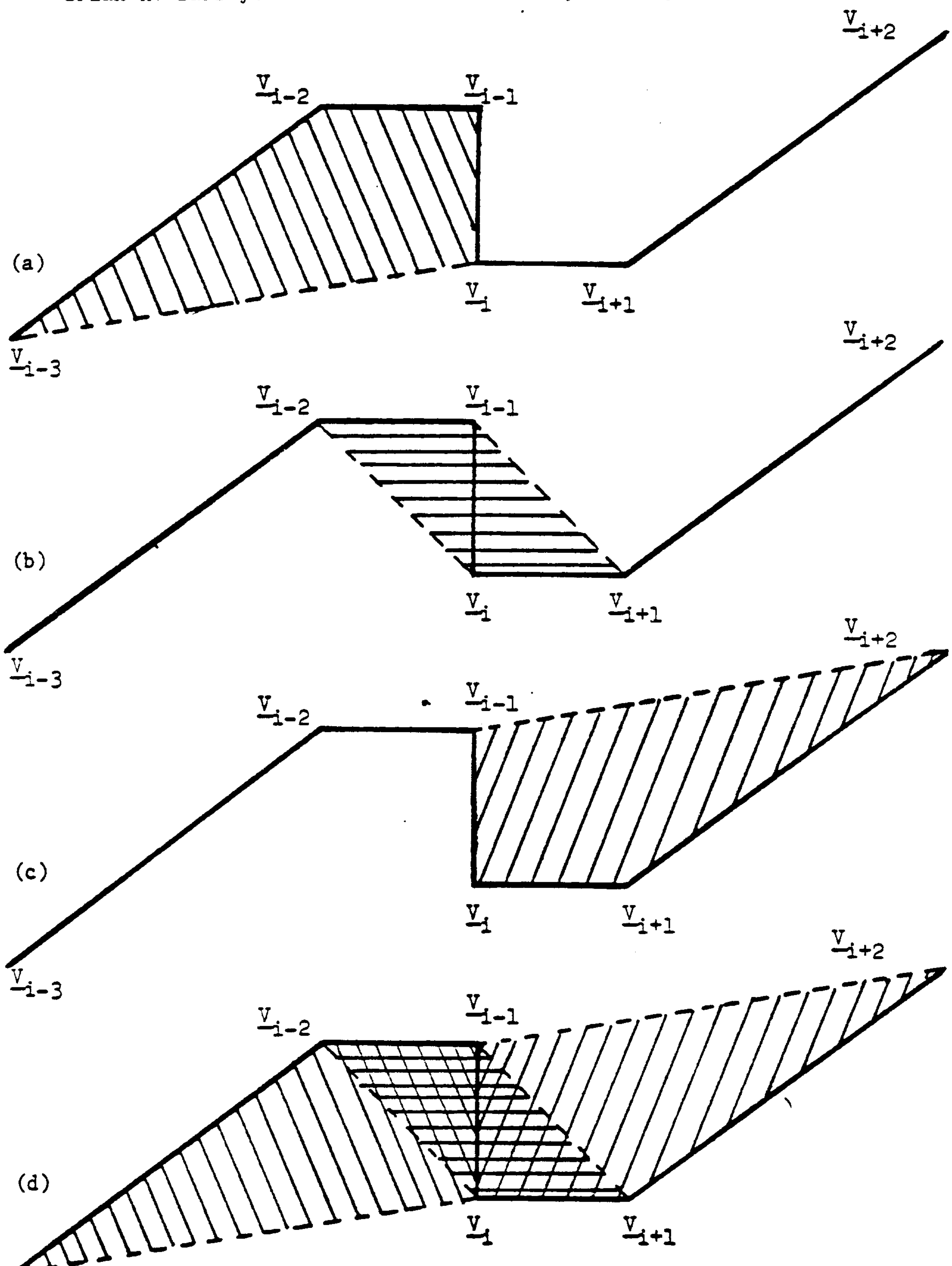


Figure III-7: Convex hull of the four control vertices $\underline{V}_{i+r}$, r=-2,-1,0,

vertices. In particular, a cubic B-spline curve will lie in the union of the convex hulls of each successive set of four control vertices (Figure III-8).

For a bipolynomial B-spline surface of degree d and e in each parameter, respectively, each surface patch lies within the convex hull of its (d+1)(e+1) defining control vertices. Hence the entire B-spline surface will be contained in the union of the convex hulls of each set of (d+1)(e+1) defining control vertices. Specifically, a bicubic B-spline surface will lie in the union of the convex hulls of each set of sixteen defining control vertices.



V_{i-3} Figure III-8: Union of the convex hulls of each of three successive set of four control vertices.

IV. THE UNIFORM CUBIC B-SPLINE CURVE REPRESENTATION

IV.1. Explanation

A cubic B-spline curve segment is completely controlled by only four of the control vertices; therefore, a point on this curve segment can be regarded as a weighted average of these four control vertices. Associated with each control vertex is a weighting factor which is a scalar-valued function evaluated at some parametric value. For a uniform B-spline curve segment, this parameter indicates the location in the segment as it varies from a value of zero at the beginning of the segment to a value of unity at the end.

In particular, let the control polygon be composed of the sequence of control vertices

$$\underline{V} = [\underline{V}_0, \underline{V}_1, \ldots, \underline{V}_m].$$

Then a point on the i-th curve segment is a weighted average of the four control vertices $\underline{V}_{i+r}$, $r = -2, -1, 0, 1$. The coordinates of the point $\underline{Q}_i(u)$ on the i-th curve segment are then given by

$$\underline{Q}_i(u) = \sum_{r=-2}^{1} b_r(u)\, \underline{V}_{i+r} \quad \text{for } 0 \le u < 1. \tag{IV.1}$$

As the parameter u varies from zero to unity, the i-th curve segment is traced out. (Figure IV-1 shows the (i-1)st, i-th, and (i+1)st cubic B-spline curve segments.)

The weighting factors are the scalar-valued functions $b_r(u)$, r = -

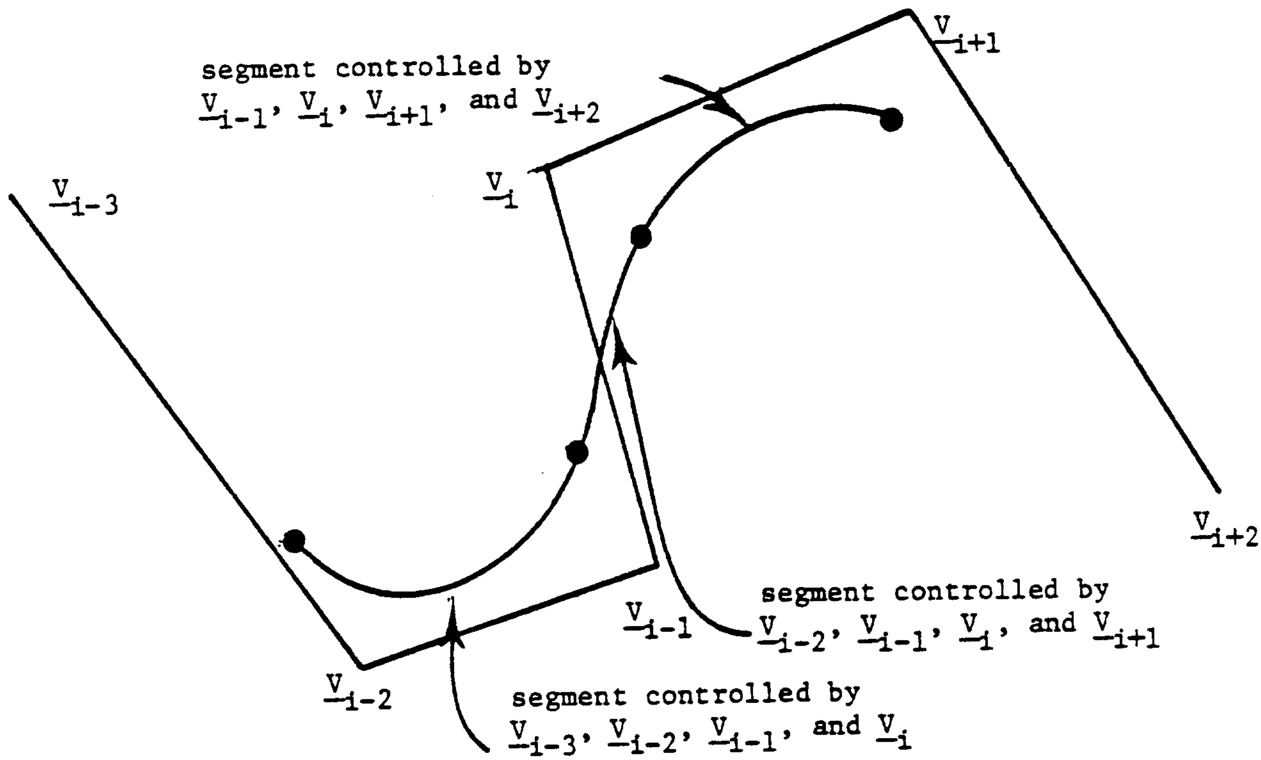


Figure IV-1: Cubic B-spline curve with its control polygon showing the control vertices controlling each segment.

2, -1, 0, 1, evaluated at some value of the parameter u. These functions are called basis functions because they form a basis; that is, they are linearly independent, and any possible B-spline curve segment can be expressed as a linear combination of them. Moreover, the combination coefficients of this linear combination are unique since the basis functions are linearly independent. Therefore, every B-spline curve segment has a unique representation as a linear combination of these basis functions, where the combination coefficients are the associated control vertices. The uniform cubic B-spline basis functions are derived in the following section.

The cubic B-spline curve segment $Q_i(u)$ is controlled by the control vertices V_{i+r}, r = -2, -1, 0, 1. For the next curve segment, $Q_{i+1}(u)$, the first of these control vertices, V_{i-2}, is dropped, and a new control vertex, $\underline{V}_{i+2}$, is added. The basis functions are shifted to this new sequence of control vertices so that the basis functions, $b_r(u)$, r = -2, -1, 0, 1, are now associated with the control vertices $\underline{V}_{i+r}$, r = -1, 0, 1, 2, respectively.

IV.2. Derivation of the uniform cubic B-spline basis functions.

A cubic B-spline curve is composed of a sequence of segments, each of which is a polynomial with a maximum degree of three. A cubic B-spline segment is a linear combination of the four basis functions, $b_r(u)$, r = -2, -1, 0, 1, where the combination coefficients are the associated control vertices.

Since each segment is a polynomial of degree at most three, and is a linear combination of basis functions, each basis function must be a polynomial with a maximum degree of three. That is,

$$b_r(u) = c_{3r}u^3 + c_{2r}u^2 + c_{1r}u + c_{0r} \quad \text{for } r = -2, -1, 0, 1. \qquad \text{(IV.2)}$$

The sixteen coefficients c_{qr}, q = 0, 1, 2, 3 and r = -2, -1, 0, 1, can be determined so that the continuity constraint is satisfied. Recall that this requires the cubic B-spline segments to join with continuity of the parametric first and second derivative vectors. Therefore,

$$\underline{Q}_i^{(a)}(1) = \underline{Q}_{i+1}^{(a)}(0) \quad \text{for } a = 0, 1, 2. \qquad \text{(IV.3)}$$

The mathematical formulations for a cubic B-spline segment $\underline{Q}_i(u)$ and its first and second derivative vectors, $\underline{Q}_i^{(1)}(u)$ and $\underline{Q}_i^{(2)}(u)$, are

$$\underline{Q}_i^{(a)}(u) = \sum_{r=-2}^{1} b_r^{(a)}(u)\, \underline{V}_{i+r} \quad \text{for } 0 \leq u < 1 \text{ for } a = 0, 1, 2. \qquad \text{(IV.4)}$$

Evaluating this expression at u=1 and u=0 for a = 0, 1, 2 and substituting it into equation (IV.3),

$$\sum_{r=-2}^{1} b_r^{(a)}(1)\, \underline{V}_{i+r} = \sum_{r=-2}^{1} b_r^{(a)}(0)\, \underline{V}_{i+1+r} \quad \text{for } a = 0, 1, 2. \tag{IV.5}$$

Solutions to this equation for a = 0, 1, 2 can be determined by equating coefficients of the vertices $\underline{V}_{i+r}$, r = -2, -1, 0, 1, 2, for a = 0, 1, 2:

$$\begin{aligned} &b_{-2}^{(a)}(1) = 0 \text{ for vertex } \underline{V}_{i-2} \\ &b_r^{(a)}(1) = b_{r-1}^{(a)}(0) \text{ for vertex } \underline{V}_{i+r},\ r = -1, 0, 1 \\ &0 = b_1^{(a)}(0) \text{ for vertex } \underline{V}_{i+2} \end{aligned} \tag{IV.6}$$

Consider these equations for a=0. Evaluating equation (IV.2) for $b_r(0)$ and $b_r(1)$, r = -2, -1, 0, 1, and substituting the resulting expressions into equation (IV.6) yields,

$$\begin{aligned} &c_{3,-2} + c_{2,-2} + c_{1,-2} + c_{0,-2} = 0 \\ &c_{3r} + c_{2r} + c_{1r} + c_{0r} = c_{0,r-1} \quad \text{for } r = -1, 0, 1 \\ &c_{01} = 0 \end{aligned} \tag{IV.7}$$

Now look at equation (IV.2) for a=1. Differentiating $b_r(u)$,

$$b_r^{(1)}(u) = 3c_{3r}u^2 + 2c_{2r}u + c_{1r} \quad \text{for } r = -2, -1, 0, 1,$$

evaluating at u=1 and u=0, and then substituting into equation (IV.6) yields

$$\begin{aligned} &3c_{3,-2} + 2c_{2,-2} + c_{1,-2} = 0 \\ &3c_{3r} + 2c_{2r} + c_{1r} = c_{1,r-1} \quad \text{for } r = -1, 0, 1 \\ &c_{11} = 0 \end{aligned} \tag{IV.8}$$

Finally, consider a=2 in equation (IV.2). Differentiating $b_r^{(1)}(u)$,

$$b_r^{(2)}(u) = 6c_{3r}u + 2c_{2r} \quad \text{for } r = -2, -1, 0, 1,$$

evaluating at u=1 and u=0, and then substituting into equation (IV.6) and simplifying, yields

$$3c_{3,-2} + c_{2,-2} = 0$$
$$3c_{3r} + c_{2r} = c_{2,r-1} \quad \text{for } r = -1,\ 0,\ 1 \qquad \text{(IV.9)}$$
$$c_{21} = 0$$

Equations (IV.7), (IV.8), and (IV.9) are now fifteen linear equations in the sixteen unknown coefficients c_{qr}, q = 0, 1, 2, 3 and r = -2, -1, 0, 1. However, they can be solved in terms of a constant k, the value of which is the same as $b_{-2}(0)$, $b_{-1}(1)$, $b_0(0)$, $b_1(1)$. Thus,

$$b_{-2}(u) = k\,(-u^3 + 3u^2 - 3u + 1)$$
$$b_{-1}(u) = k\,(3u^3 - 6u^2 + 4) \qquad \text{(IV.10)}$$
$$b_0(u) = k\,(-3u^3 + 3u^2 + 3u + 1)$$
$$b_1(u) = k\,u^3$$

One more constraint is required in order to determine this constant k. A useful constraint is to normalize the basis functions; that is, to require that they sum to unity for any value of the parameter u between zero and unity. Since each basis function is nonnegative (for $k \geq 0$) for $0 \leq u < 1$, this requirement means that a point on the i-th cubic B-spline curve segment is a convex combination of the four control vertices $\underline{V}_{i+r}$, r = -2, -1, 0, 1. As explained in Section III.7, this guarantees that the entire cubic B-spline curve will be contained in the union of the convex hulls of each successive set of four control vertices. Thus, k is determined by

$$\sum_{r=-2}^{1} b_r(u) = 1 \quad \text{for } 0 \leq u < 1. \qquad \text{(IV.11)}$$

Substituting the expressions given in equation (IV.10) for the basis functions into this equation yields

$$k = 1/6.$$

The basis functions are then

$$\begin{aligned} b_{-2}(u) &= (-u^3 + 3u^2 - 3u + 1) / 6 \\ b_{-1}(u) &= (3u^3 - 6u^2 + 4) / 6 \\ b_0(u) &= (-3u^3 + 3u^2 + 3u + 1) / 6 \\ b_1(u) &= u^3 / 6 \end{aligned} \qquad \text{(IV.12)}$$

This can be written in matrix form as

$$[b_{-2}(u)\ b_{-1}(u)\ b_0(u)\ b_1(u)] = [u^3\ u^2\ u\ 1]\ (1/6) \begin{vmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{vmatrix}. \qquad \text{(IV.13)}$$

Graphs of the uniform cubic B-spline basis functions are shown in Figure IV-2.

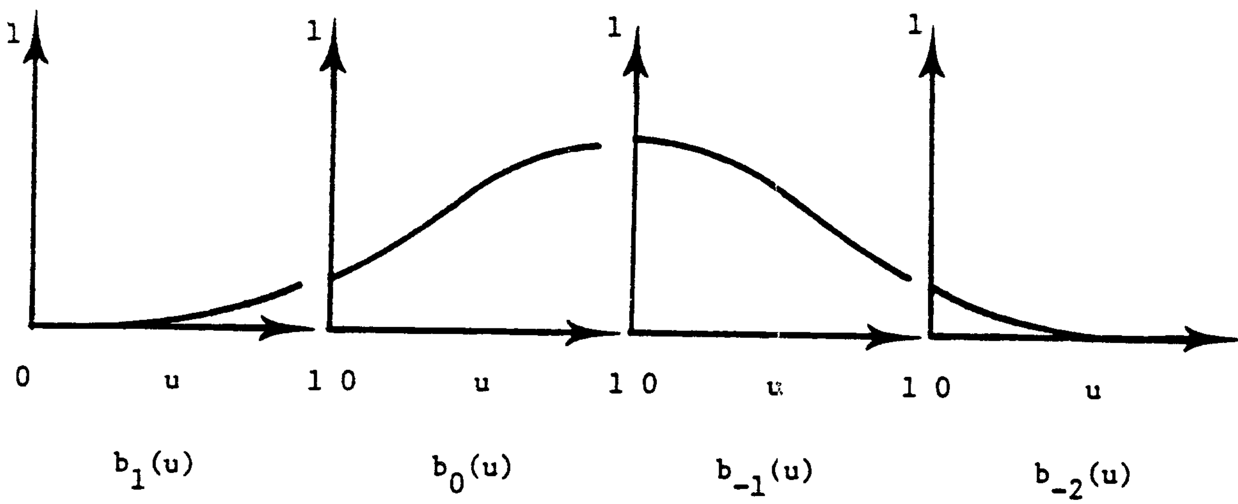


Figure IV-2: Graphs of the uniform cubic B-spline basis functions

V. B-SPLINE CURVE END CONDITIONS

V.1. Classification

Recall that the control polygon is composed of the m+1 control vertices

$$\{\underline{V}_0, \underline{V}_1, \ldots, \underline{V}_m\}$$

(Figure III-1). Considering the cubic B-spline curve formulation (equation (IV.1)), it can be seen that these vertices can be used to generate m-2 curve segments, specifically $\underline{Q}_2(u)$, $\underline{Q}_3(u)$, ..., $\underline{Q}_{m-1}(u)$ (Figure V-1). Note that the B-spline curve starts at

$$\underline{Q}_2(0) = (\underline{V}_0 + 4\underline{V}_1 + \underline{V}_2) / 6$$

and ends at

$$\underline{Q}_{m-1}(1) = (\underline{V}_{m-2} + 4\underline{V}_{m-1} + \underline{V}_m) / 6.$$

In order to have the curve start closer to $\underline{V}_0$ and end nearer $\underline{V}_m$, additional curve segments can be defined at the ends. The definition of these segments, however, cannot be done by evaluating equation (IV.1) in the usual way since this would reference nonexistent vertices. Various methods are available for defining these curve segments and will now be described. These techniques fall into two classifications, multiple vertices and phantom vertices. Different end condition techniques have various geometric properties which require careful study to enable the selection of an appropriate approach. The properties which each

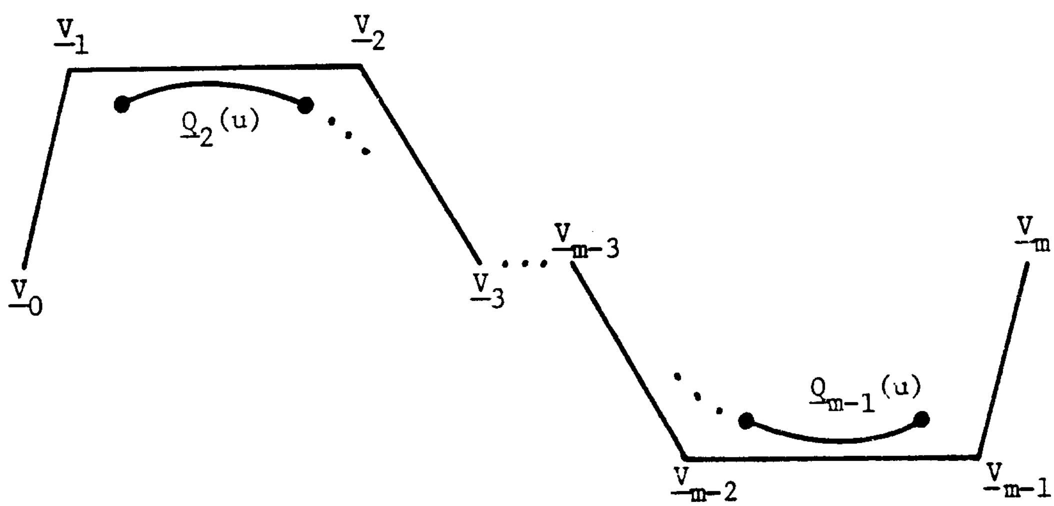


Figure V-1: Interior segments naturally defined by the control polygon.

engenders are investigated in [2].

V.2. Explanation of multiple vertices end conditions

V.2.i. Double vertices

Using this technique one additional curve segment is defined at each end by repeating the end vertex in the B-spline curve formulation. These segments are $\underline{Q}_1(u)$ and $\underline{Q}_m(u)$ (Figure V-2) and they are defined by equation (IV.1) in the usual manner except that vertices $\underline{V}_0$ and $\underline{V}_m$ are used when $\underline{V}_{-1}$ and $\underline{V}_{m+1}$, respectively, are referenced. Thus,

$$\underline{Q}_1(u) = [b_{-2}(u) + b_{-1}(u)]\underline{V}_0 + b_0(u)\underline{V}_1 + b_1(u)\underline{V}_2 \qquad (V.1)$$

and

$$\underline{Q}_m(u) = b_{-2}(u)\underline{V}_{m-2} + b_{-1}(u)\underline{V}_{m-1} + [b_0(u) + b_1(u)]\underline{V}_m. \qquad (V.2)$$

V.2.ii. Triple vertices

The triple vertices technique is an extension of the double vertices technique. In fact, it is simply the double vertices technique with the definition of another additional curve segment at each end. The segments $\underline{Q}_1(u)$ and $\underline{Q}_m(u)$ are defined by the double vertices technique and the segments $\underline{Q}_0(u)$ and $\underline{Q}_{m+1}(u)$ (Figure V-3) are defined by equation (IV.1) using $\underline{V}_0$ whenever $\underline{V}_{-1}$ or $\underline{V}_{-2}$ is referenced and using $\underline{V}_m$ for $\underline{V}_{m+1}$ or $\underline{V}_{m+2}$. Thus,

$$\begin{aligned} \underline{Q}_0(u) &= [b_{-2}(u) + b_{-1}(u) + b_0(u)]\underline{V}_0 + b_1(u)\underline{V}_1 \\ \underline{Q}_{m+1}(u) &= b_{-2}(u)\underline{V}_{m-1} + [b_{-1}(u) + b_0(u) + b_1(u)]\underline{V}_m. \end{aligned} \quad (V.3)$$

V.3. Explanation of phantom vertices end conditions

V.3.i. Description

With these techniques, an auxiliary vertex is created at each end of the control polygon. This can then be used to define an additional curve segment at each end by evaluating the B-spline curve formulation (equation (IV.1)) in the same manner as for the curve segments defined by the original control polygon (Figure V-4).

The auxiliary vertices are created for the sole purpose of defining the additional curve segments, and are inaccessible to the user and not displayed; thus they will be referred to as phantom vertices. The phantom vertices are completely defined in terms of the original control vertices in such a manner so as to satisfy some end condition. Several such end conditions are discussed in the following sections.

This phantom vertices concept was developed independently of that which was mentioned in [12] by Coons, although the underlying idea is similar. The latter case was developed only for B-spline curves and not

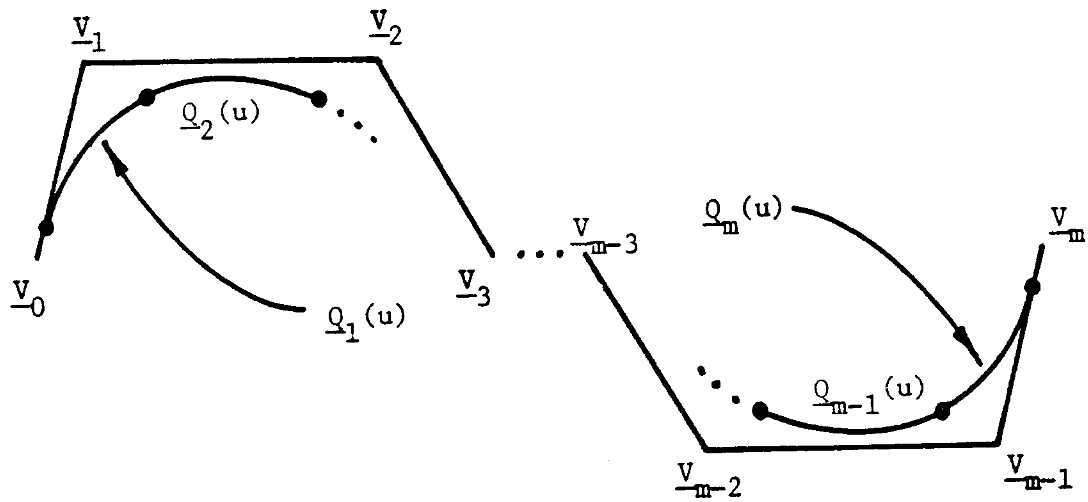


Figure V-2: Additional segments from double vertices end condition.

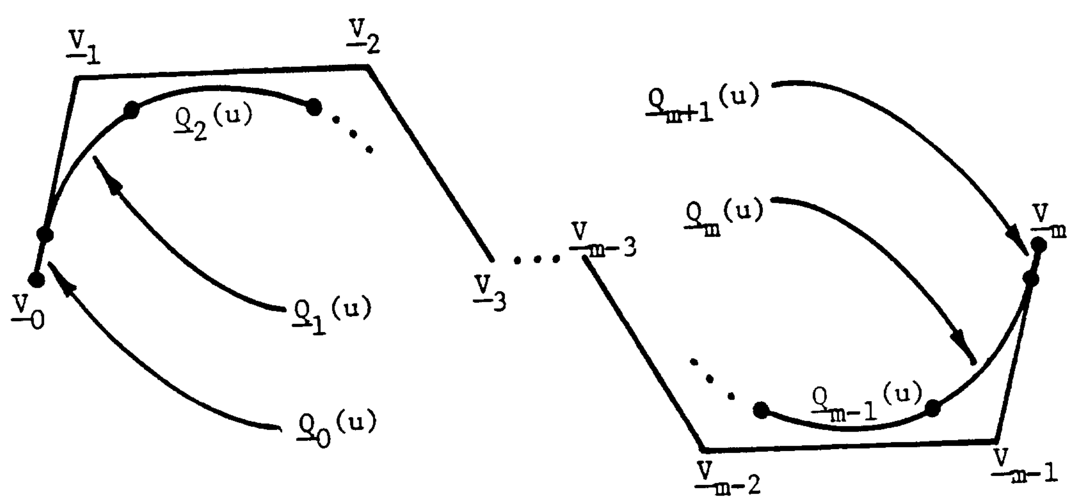


Figure V-3: Additional segments from triple vertices end condition.

for surfaces, and this description of a curve involved a total of four phantom vertices: two initiating vertices at the beginning of the curve, and two terminating vertices at the end. The two initiating vertices are defined in terms of the first non-phantom vertex and two furnished points. Similarly, the two phantom terminating vertices are expressed in terms of the final non-phantom vertex and another two specified points.

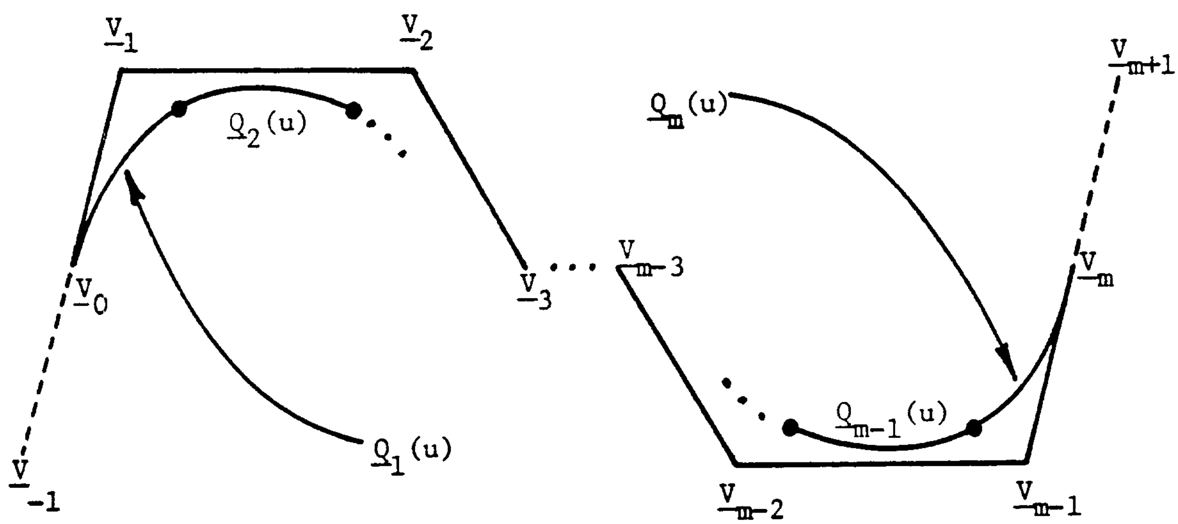


Figure V-4: The phantom vertices and additional curve segments defined by phantom vertices end condition.

V.3.ii. Position specification

Using this end condition, the phantom vertices are defined such that each endpoint of the curve interpolates a specified point. That is,

$$\underline{Q}_1(0) = \underline{P}_0 \qquad \underline{Q}_m(1) = \underline{P}_m. \tag{V.4}$$

Evaluating the left hand sides of these equations by substituting the extreme parametric values 0 and 1 into equation (IV.1) results in

$$\underline{V}_{-1}/6 + (2/3)\underline{V}_0 + \underline{V}_1/6 = \underline{P}_0 \qquad \underline{V}_{m-1}/6 + (2/3)\underline{V}_m + \underline{V}_{m+1}/6 = \underline{P}_m. \tag{V.5}$$

Thus, the phantom vertices are

$$\underline{V}_{-1} = 6\underline{P}_0 - (4\underline{V}_0 + \underline{V}_1) \qquad \underline{V}_{m+1} = 6\underline{P}_m - (4\underline{V}_m + \underline{V}_{m-1}). \tag{V.6}$$

V.3.iii. End vertex interpolation

Although the freedom to select the initial and terminal position of the curve is frequently desirable, it is often convenient to constrain these positions to coincide with the initial and terminal vertex, respectively; that is, to have the curve start at $\underline{V}_0$ and end at $\underline{V}_m$. This is a special case of the previous end condition where $\underline{P}_0 = \underline{V}_0$ and $\underline{P}_m = \underline{V}_m$.

Making this substitution into equation (V.6) yields the following expressions for the phantom vertices:

$$\underline{V}_{-1} = 2\underline{V}_0 + \underline{V}_1 \qquad \underline{V}_{m+1} = 2\underline{V}_m + \underline{V}_{m-1}. \tag{V.7}$$

V.3.iv. Parametric first derivative vector specification

This end condition defines the phantom vertices by setting the parametric first derivative vector, at each end, equal to some specified value. Specifically,

$$\begin{aligned} \underline{Q}_1^{(1)}(0) &= \underline{P}_0^1 \\ \underline{Q}_m^{(1)}(1) &= \underline{P}_m^1. \end{aligned} \qquad \text{(V.8)}$$

The left hand side of these equations can be evaluated by using the first derivative of equation (IV.1) yielding

$$\begin{aligned} (\underline{V}_1 - \underline{V}_{-1}) \,/\, 2 &= \underline{P}_0^1 \\ (\underline{V}_{m+1} - \underline{V}_{m-1}) \,/\, 2 &= \underline{P}_m^1. \end{aligned} \qquad \text{(V.9)}$$

The phantom vertices are then

$$\begin{aligned} \underline{V}_{-1} &= \underline{V}_1 - 2\underline{P}_0^1 \\ \underline{V}_{m+1} &= 2\underline{P}_m^1 + \underline{V}_{m-1}. \end{aligned} \qquad \text{(V.10)}$$

V.3.v. Parametric second derivative vector specification

The phantom vertices are defined by this end condition by setting the parametric second derivative vector, at each end, equal to some specified value. In particular,

$$\begin{aligned} \underline{Q}_1^{(2)}(0) &= \underline{P}_0^2 \\ \underline{Q}_m^{(2)}(1) &= \underline{P}_m^2. \end{aligned} \qquad \text{(V.11)}$$

Evaluating the left hand side of these equations, they can be rewritten as

$$\begin{aligned} \underline{V}_{-1} - 2\underline{V}_0 + \underline{V}_1 &= \underline{P}_0^2 \\ \underline{V}_{m-1} - 2\underline{V}_m + \underline{V}_{m+1} &= \underline{P}_m^2. \end{aligned} \qquad \text{(V.12)}$$

Solving for the phantom vertices then yields

$$\underline{V}_{-1} = 2\underline{V}_0 - \underline{V}_1 + \underline{P}_0^2$$
$$\underline{V}_{m+1} = 2\underline{V}_m - \underline{V}_{m-1} + \underline{P}_m^2. \qquad (V.13)$$

V.3.vi. Zero parametric second derivative vector

While the capability of specifying the parametric second derivative vector at each end of the curve is often desirable, it is frequently convenient to have them set to zero by default. However, it was shown in [2] that the end vertex interpolation end condition had just this property. Thus, the zero parametric second derivative vector end condition is equivalent to the end vertex interpolation end condition.

VI. EVALUATION AND PERTURBATION OF A B-SPLINE CURVE

VI.1. Basis function evaluation

The uniform cubic basis functions can be evaluated efficiently, for a given parametric value, by storing u^2 and u^3 and rewriting the expressions using as much factoring as possible. That is,

```
u2 := u*u;
u3 := u2*u;
b_-2(u) := (1 - 3*(u-u2) - u3)/6;
b_-1(u) := u3/2 - u2 + 2/3;
b_0(u) := (1 + 3*(u+u2-u3))/6;
b_1(u) := u3/6;
```

This requires three additions, five subtractions, four multiplications, and four divisions for each value of the parameter u.

However, a more efficient computation can be performed by exploiting the symmetry of the basis functions. Observe that

$$b_{-2}(1-u) = b_1(u)$$
$$b_{-1}(1-u) = b_0(u). \qquad \text{(VI.1)}$$

Thus, the computational requirements can be reduced if the basis functions are evaluated at p+1 symmetrically spaced parametric values; that is, at $u = u_0(=0), u_1, u_2, \ldots, u_p(=1)$ satisfying

$$u_a - u_{a-1} = u_{p-a+1} - u_{p-a} \quad a = 1, 2, \ldots, \; p/2 \tag{VI.2}$$

where $u_0=0$ and $u_p=1$.

A special case of symmetrical spacing is equal spacing; that is,

$$u_a - u_{a-1} = 1/p \quad \text{for } a = 1, 2, \ldots, p \tag{VI.3}$$

The algorithm to compute this equally spaced case is:

```
for u := 0 step 1/p to 1 do
     begin

     u2 := u*u;

     u3 := u2*u;

     t := 1-u;

     b_-2(t) := b_1(u) := u3/6;

     b_-1(u) := b_0(t) := u3/2 - u2 + 2/3

     end;
```

This algorithm requires p+1 additions, 2(p+1) subtractions, 2(p+1) multiplications, and 2(p+1) divisions. Note that this is less than half what would have been required if each basis function were evaluated at each successive parametric value.

VI.2. Curve evaluation

The computation of a set of points on a single curve segment requires the values of the four basis functions at each of the corresponding parametric values. For a curve composed of m segments, straightforward curve evaluation would require this basis function computation to be performed m times. However, this can be avoided by exploiting the fact that the uniform basis functions for all the B-spline curve segments are identical -- only the set of parametric values at which they are evaluated may differ. By restricting this set of parametric values to be the same for all segments, the evaluation of the basis functions becomes identical for every segment and therefore needs to be performed only once and stored in a table. Then the evaluation of each coordinate of each point on the curve requires four multiplications and three additions; thus, the computation of the p+1 curve points on the m segments requires 4m(p+1) multiplications and 3m(p+1) additions. Note that this does not impose any restriction on the selection of a particular set of parametric values, although a convenient choice is equally spaced values (equation (VI.3)).

VI.3. Curve perturbation

If an already-existing curve is to be modified, it is not necessary to recompute the entire curve. Careful consideration of the properties of the B-spline representation enables an existing curve to be modified in a manner which is more efficient than a complete recomputation.

Consider the consequences to an existing curve when the position of one control vertex is modified. Since a single control vertex influences only four curve segments and has no effect on the other segments, the consequences of moving one vertex are limited to four segments. Computationally, this implies that the movement of a control vertex requires the re-evaluation of only four segments.

Moreover, even the four affected segments need not be completely recomputed. Although each of these segments is controlled by four vertices, only one of these vertices has changed position. Therefore, the change in each of these segments is due only to the modification of the position of one control vertex. Recalling the mathematical formulation for the i-th curve segment given in equation (IV.1), the change in this segment, $\underline{Q}_i(u)$, can be written as

$$\underline{Q}_i^{\Delta}(u) = \sum_{r=-2}^{1} b_r(u)\, \underline{V}_{i+r}^{\Delta} \qquad \text{(VI.4)}$$

where $\underline{V}_i^{\Delta}$ is the change in position of control vertex $\underline{V}_i$.

Denoting the modified control vertex as $\underline{V}_{\hat{i}}$, and assuming all the other vertices remain unchanged, then

$$\underline{V}_i^{\Delta} = 0 \quad \text{for } i \neq \hat{i}. \qquad \text{(VI.5)}$$

Thus, only one of the four terms in equation (VI.4) is nonzero, and hence this equation reduces to

$$\underline{Q}^{\Delta}_{i}(u) = b_r(u)\ \underline{V}^{\Delta}_{i+r} \tag{VI.6}$$

where $i+r = \hat{i}$.

Rewriting this equation as

$$\underline{Q}^{\Delta}_{\hat{i}-r}(u) = b_r(u)\ \underline{V}^{\Delta}_{\hat{i}} \quad \text{for } r = -2,\ -1,\ 0,\ 1, \tag{VI.7}$$

it is easily seen that the four affected curve segments are $\underline{Q}_i(u)$, where

$$i = \hat{i} - r \quad \text{for } r = -2,\ -1,\ 0,\ 1. \tag{VI.8}$$

Therefore, the change in position of control vertex $\underline{V}_{\hat{i}}$ perturbs the segments $\underline{Q}_i(u)$ by

$$\underline{Q}^{\Delta}_{i}(u) = b_{\hat{i}-i}(u)\ \underline{V}^{\Delta}_{\hat{i}} \tag{VI.9}$$

where i takes on the values specified by equation (VI.8).

Since equation (VI.9) represents the change in the curve segment $\underline{Q}_i(u)$, the new segment can be determined by incrementing the old segment by this change:

$$\underline{Q}^{new}_{i}(u) = \underline{Q}^{old}_{i}(u) + b_{\hat{i}-i}(u)\ \underline{V}^{\Delta}_{\hat{i}} \tag{VI.10}$$

To compute the new curve resulting from modifying the position of the control vertex $\underline{V}_{\hat{i}}$, equation (VI.10) is evaluated for all the necessary values of the parameter u and for each of the four curve segments $\underline{Q}_i(u)$ with the values of i given by equation (VI.8). Thus, the algorithm to compute the four perturbed curve segments at the parametric values $u = u_0,\ u_1,\ \ldots,\ u_p$ is:

```
for i := î-1 to î+2 do
    for each u in {u0, u1, ..., up} do
        Qi^new(u) := Qi^old(u) + b_(î-i)(u)*Vî^Δ;
```

This algorithm requires one multiplication and one addition for each coordinate of each of the p+1 points on each of the four segments; thus, the total computational requirement is 4(p+1) multiplications and 4(p+1) additions per component. This is one-quarter of the multiplications and one-third of the additions that would be required if the four affected segments were completely recomputed, and far less than the computational requirements of recomputing the entire curve.

It should be emphasized that this algorithm is based on the assumption that only one control vertex has been moved. If the positions of more than one vertex are to be modified, they must be moved one at a time, and the algorithm must be performed for each such change. The algorithm is not valid in cases where the positions of several vertices are modified simultaneously.

VII. DIFFERENCE TECHNIQUES FOR THE EVALUATION AND PERTURBATION OF A B-SPLINE CURVE

VII.1. Background

The forward difference of f(s) with respect to the difference δ is defined as

$$\Delta_\delta f(s) = f(s+\delta) - f(s). \tag{VII.1}$$

The k-th forward difference is defined recursively as

$$\Delta_\delta^k f(s) = \begin{cases} \Delta_\delta^{k-1} f(s+\delta) - \Delta_\delta^{k-1} f(s) & k = 1, 2, 3, \ldots \\ f(s) & k = 0 \end{cases} \tag{VII.2}$$

From equation (VII.1), it can be seen that the first forward difference of a d-th degree polynomial is a (d-1)st degree polynomial. This result can then be used to show that the (d+1)st forward difference of this polynomial is zero and hence, by induction, so are all the succeeding forward differences.

Consider a generic cubic polynomial

$$f(s) = \sum_{k=0}^{3} a_k \, s^k \tag{VII.3}$$

Applying equation (VII.2),

$$\Delta_\delta f(0) = \sum_{k=1}^{3} a_k \delta^k = a_1\delta + a_2\delta^2 + a_3\delta^3$$

$$\Delta_\delta^2 f(0) = \sum_{k=2}^{3} 2a_k \delta^k (2^{k-1}-1) = 2a_2\delta^2 + 6a_3\delta^3 \qquad \text{(VII.4)}$$

$$\Delta_\delta^3 f(0) = \sum_{k=3}^{3} 3a_k \delta^k (3^{k-1}-2^k+1) = 6a_3\delta^3$$

and, from above,

$$\Delta_\delta^k f(0) = 0 \quad \text{for } k = 4, 5, 6, \ldots \qquad \text{(VII.5)}$$

VII.2. Curve evaluation

Substituting $\underline{Q}_i(u)$ for f(s) in equation (VII.2),

$$\Delta_\delta^k \underline{Q}_i(u) = \begin{cases} \Delta_\delta^{k-1}\underline{Q}_i(u+\delta) - \Delta_\delta^{k-1}\underline{Q}_i(u) & k = 1, 2, 3, \ldots \\ \underline{Q}_i(u) & k = 0 \end{cases} \qquad \text{(VII.6)}$$

and therefore

$$\Delta_\delta^{k-1}\underline{Q}_i(u+\ ) = \Delta_\delta^{k-1}\underline{Q}_i(u) + \Delta_\delta^k \underline{Q}_i(u) \quad \text{for } k = 1, 2, 3, \ldots \qquad \text{(VII.7)}$$

From equations (IV.1) and (VII.2), it can be shown that the k-th forward difference of $\underline{Q}_i(u)$ is

$$\Delta_\delta^k \underline{Q}_i(u) = \sum_{r=-2}^{1} \Delta_\delta^k b_r(u)\, \underline{V}_{i+r} \qquad \text{(VII.8)}$$

Since the basis functions are cubic polynomials, the fourth and succeeding forward differences are zero

$$\Delta_\delta^k b_r(u) = 0 \quad \text{for } r = -2, -1, 0, 1 \text{ and } k = 4, 5, 6, \ldots \qquad \text{(VII.9)}$$

From equations (VII.8) and (VII.9), then,

$$\Delta_\delta^k \underline{Q}_i(u) = 0 \quad \text{for } k = 4, 5, 6, \ldots \qquad \text{(VII.10)}$$

Substituting equation (VII.10) into equation (VII.7) with k=4 yields

$$\Delta_\delta^3 \underline{Q}_i(u+\delta) = \Delta_\delta^3 \underline{Q}_i(u) \qquad \text{(VII.11)}$$

and therefore $\Delta_\delta^3 \underline{Q}_i(u)$ is constant. Thus, the differences to be computed, at each step, are as follows:

$$\begin{aligned} \underline{Q}_i(u+\delta) &= \underline{Q}_i(u) + \Delta_\delta \underline{Q}_i(u) \\ \Delta_\delta \underline{Q}_i(u+\delta) &= \Delta_\delta \underline{Q}_i(u) + \Delta_\delta^2 \underline{Q}_i(u) \\ \Delta_\delta^2 \underline{Q}_i(u+\delta) &= \Delta_\delta^2 \underline{Q}_i(u) + \Delta_\delta^3 \underline{Q}_i(0) \end{aligned} \qquad \text{(VII.12)}$$

The initialization requires the values of $\Delta_\delta^k \underline{Q}_i(0)$, $k = 0, 1, 2, 3$ which can be computed from equation (VII.8) given the values of $\Delta_\delta^k b_r(0)$, $k = 0, 1, 2, 3$; $r = -2, -1, 0, 1$. The latter are determined using equation (VII.4) for the four basis functions and are tabulated in Table VII-1.

r	$b_r(0)$	$\Delta_\delta b_r(0)$	$\Delta_\delta^2 b_r(0)$	$\Delta_\delta^3 b_r(0)$
-2	1/6	$(-\delta^3 + 3\delta^2 - 3\delta)$ / 6	$-\delta^3 + \delta^2$	$-\delta^3$
-1	2/3	$(3\delta^3 - 6\delta^2)$ / 6	$3\delta^3 - 2\delta^2$	$3\delta^3$
0	1/6	$(-\delta^3 + \delta^2 + \delta)$ / 2	$-3\delta^3 + \delta^2$	$-3\delta^3$
1	0	δ^3 / 6	δ^3	δ^3

Table VII-1: Forward differences of the four basis functions at zero.

This difference technique can be used to evaluate equally spaced points on the curve. Let

$$u_a = a\delta \text{ where } \delta = 1/p. \qquad \text{(VII.13)}$$

Then the algorithm to evaluate $\underline{Q}_i(u)$ for $a = 0, 1, \ldots, p$ and $i = 1, \ldots, m$ is:

```
Compute Δ_δ^k b_r(0) for k := 0 to 3 and r := -2 to 1;

for i := 1 to m do
    begin (* curve segment i *)

    for k := 0 to 3 do
        begin (* compute Δ_δ^k Q_i(0) *)

        Δ_δ^k Q_i(0) := Δ_δ^k b_-2(0)*V_i-2;

        for r := -1 to 1 do Δ_δ^k Q_i(0) := Δ_δ^k Q_i(0) + Δ_δ^k b_r(0)*V_i+r;

        end (* compute Δ_δ^k Q_i(0) *);

    for a := 1 to p do
        begin (* compute differences at u=u_a *)

        Q_i(u_a) := Q_i(u_a-1) + Δ_δ Q_i(u_a-1);

        Δ_δ Q_i(u_a) := Δ_δ Q_i(u_a-1) + Δ_δ^2 Q_i(u_a-1);

        Δ_δ^2 Q_i(u_a) := Δ_δ^2 Q_i(u_a-1) + Δ_δ^3 Q_i(0)

        end (* compute differences at u=u_a *)

    end (* curve segment i *);
```

This algorithm requires the computation of $\Delta_\delta^k b_r(0)$ for k = 0, 1, 2, 3 and r = -2, -1, 0, 1, plus 16m multiplications and m(12+3p) additions per coordinate to compute the p+1 curve points on the m segments.

VII.3. Curve perturbation

The forward difference technique can also be used to modify an already existing curve. This can be achieved by making two modifications to the algorithm given in the previous section. First, only four segments must be re-evaluated, and second, the computation of $\Delta_\delta^k \underline{Q}_i(0)$, k = 0, 1, 2, 3, can be accomplished simply by incrementing by the appropriate change in value. That is,

```
for i := î-1 to î+2 do
    begin (* curve segment i *)
    for k := 0 to 3 do
        Δδ^k Q_i(0) := Δδ^k Q_i(0) + Δδ^k b_{î-i}(0)*ΔV_î;
    for a := 1 to p do
        for k := 0 to 2 do
            Δδ^k Q_i(u_a) := Δδ^k Q_i(u_{a-1}) + Δδ^{k+1} Q(u_{a-1})
    end (* curve segment i *);
```

This algorithm requires 16 multiplications and 4(4+3p) additions per component to compute the p+1 curve points on the four perturbed segments.

VII.4. Conclusion

The advantage of the foregoing technique lies in the minimal number of multiplications required. The inherent difficulty, however, is that since each point is dependent on the previous one, there is an accumulating error. Furthermore, in order for a curve to appear smooth, it is desirable to evaluate many closely spaced points, but, at the same time, this increases the sensitivity to cumulative error [11].

VIII. GEOMETRICAL INTERPRETATIONS FOR $\underline{Q}_i(0)$, $\underline{Q}_i^{(1)}(0)$, and $\underline{Q}_i^{(2)}(0)$

Given a control polygon, a B-spline curve can be easily sketched because there are straightforward geometrical interpretations for the point at the beginning of the i-th B-spline curve segment, and the first and second derivative vectors there. Considering the expressions for these quantities,

$$\begin{aligned} \underline{Q}_i(0) &= (\underline{V}_{i-2} + 4\underline{V}_{i-1} + \underline{V}_i)/6 \\ \underline{Q}_i^{(1)}(0) &= (\underline{V}_i - \underline{V}_{i-2}) \;/\; 2 \\ \underline{Q}_i^{(2)}(0) &= \underline{V}_{i-2} - 2\,\underline{V}_{i-1} + \underline{V}_i \end{aligned} \qquad \text{(VIII.1)}$$

the following observations can be made. The second derivative vector at the beginning of the segment is the sum of two vectors emanating from $\underline{V}_{i-1}$: one to $\underline{V}_{i-2}$ and one to $\underline{V}_i$. The first derivative vector there is in the direction from $\underline{V}_{i-2}$ to $\underline{V}_i$, but its magnitude is half that distance. The beginning point can be expressed as

$$\underline{Q}_i(0) = \underline{V}_{i-1} + \underline{Q}_i^{(2)}(0)/6 \qquad \text{(VIII.2)}$$

and this is located one-sixth along the second derivative vector from $\underline{V}_{i-1}$ (Figure VIII-1).

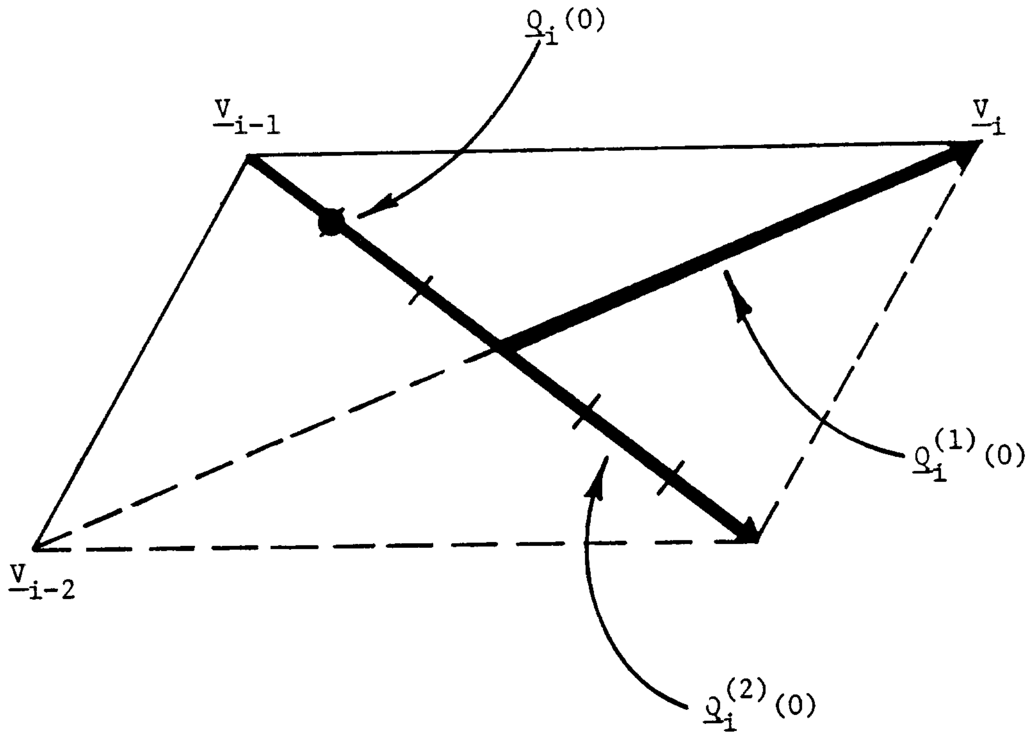


Figure VIII-1: Geometrical interpretations

for the beginning point of the i-th segment
and for the first and second derivative vectors there.

IX. THE UNIFORM BICUBIC B-SPLINE SURFACE REPRESENTATION

IX.1. Control graph

A B-spline surface is defined by, but does not interpolate, a set of control vertices, in three-dimensional x-y-z space, which are organized as a two dimensional graph with a rectangular topology. Each vertex is either an interior vertex or a boundary vertex. An interior vertex has four well-defined neighbouring vertices. A boundary vertex has three neighbouring vertices, except for the four corner vertices, each of which has only two neighbours.

This notion can be formalized quite elegantly by drawing on graph theory. The set of control vertices can be considered as a graph $\{\underline{V}, \underline{E}\}$ whose vertices form the set

$$\underline{V} = \{\underline{V}_{ij} \mid i = 0, \ldots, m;\ j = 0, \ldots, n\}$$

and with the set of edges

$$\underline{E} = \{(\underline{V}_{ij}, \underline{V}_{i,j+1}) \mid i = 0, \ldots, m;\ j = 0, \ldots, n-1\} \cup \{(\underline{V}_{ij}, \underline{V}_{i+1,j}) \mid i = 0, \ldots, m-1;\ j = 0, \ldots, n\}.$$

The interior vertices are the vertices

$$\underline{V}_{ij} \text{ where } 1 \leq i \leq m-1 \quad\text{and}\quad 1 \leq j \leq n-1,$$

and the boundary vertices are

$\underline{V}_{0j}$, j = 0, ..., n-1;

$\underline{V}_{in}$, i = 0, ..., m-1;

$\underline{V}_{mj}$, j = 1, ..., n; and

$\underline{V}_{i0}$, i = 1, ..., m.

To emphasize this graph theoretic interpretation, the author has chosen the term <u>control graph</u> to describe the set of control vertices (Figure III-2).

IX.2. Explanation

A point on the (i,j)-th bicubic B-spline surface patch is a weighted average of the sixteen vertices $\underline{V}_{i+r,j+s}$, r = -2, -1, 0, 1, and s = -2, -1, 0, 1. The mathematical formulation for the patch $\underline{Q}_{ij}(u,v)$ is then

$$\underline{Q}_{ij}(u,v) = \sum_{r=-2}^{1} \sum_{s=-2}^{1} bb_{rs}(u,v)\, \underline{V}_{i+r,j+s} \qquad \text{(IX.1)}$$

for $0 \leq u < 1$ and $0 \leq v < 1$.

The set of bivariate uniform basis functions is the tensor product of the set of univariate uniform basis functions. That is,

$$bb_{rs}(u,v) = b_r(u)\, b_s(v) \qquad \text{(IX.2)}$$

for r = -2, -1, 0, 1 and s = -2, -1, 0, 1.

Therefore, this formulation can be rewritten as

$$\underline{Q}_{ij}(u,v) = \sum_{r=-2}^{1} \sum_{s=-2}^{1} b_r(u)\, \underline{V}_{i+r,j+s}\, b_s(v) \qquad \text{(IX.3)}$$

for $0 \leq u < 1$ and $0 \leq v < 1$.

Observing that $b_r(u)$ is independent of s, it can be treated as a constant multiplier in the inner sum; thus, equation (IX.3) can be rewritten in the following form:

$$\underline{Q}_{ij}(u,v) = \sum_{r=-2}^{1} [b_r(u) \sum_{s=-2}^{1} \underline{V}_{i+r,j+s}\, b_s(v)] \qquad \text{(IX.4)}$$

for $0 \leq u < 1$ and $0 \leq v < 1$.

These mathematical formulations are used in the design of an algorithm to construct the B-spline surface as explained in Section XI.

X. B-SPLINE SURFACE BOUNDARY CONDITIONS

X.1. Classification

The (m+1)x(n+1) control graph described in section IX.1 (Figure III-2) contains the set of control vertices

$$V = \{\underline{V}_{ij} \mid i = 0, \ldots, m;\ j = 0, \ldots, n\}.$$

Using the bicubic B-spline surface formulation (equation (IX.4)), it can be seen that these vertices naturally define the interior patches

$$\underline{Q}_{ij}(u,v),\ i = 2, \ldots, m-1;\ j = 2, \ldots, n-1$$

(Figure X-1). It is desirable to have additional patches around the periphery which are more dominated by the boundary vertices. Analogous to the B-spline curve formulation, such additional patches cannot be defined simply by evaluating the B-spline surface formulation (equation (IX.4)) in the usual manner because this would reference nonexistent vertices. This problem can be circumvented using two different types of boundary condition techniques, <u>multiple vertices</u> and <u>phantom vertices</u>, which will now be described. The corresponding geometric properties are analyzed in [2].

X.2. Explanation of multiple vertices boundary conditions

X.2.i. Double vertices

With this technique, additional surface patches are defined around the periphery of the interior patches which were naturally defined by the control graph, by repeating boundary vertices in the B-spline surface formulation. The interior patches are then surrounded by the additional patches

$\underline{Q}_{1j}(u,v)$, $j = 1, \ldots, n-1$;

$\underline{Q}_{in}(u,v)$, $i = 1, \ldots, m-1$;

$\underline{Q}_{mj}(u,v)$, $j = 2, \ldots, n$; and

$\underline{Q}_{i1}(u,v)$, $i = 2, \ldots, m$

(see Figure X-2).

The additional patches are defined by evaluating the usual B-spline surface formulation (equation (IX.4)) except that whenever a nonexistent vertex is referenced, the "nearest" boundary vertex is used instead. Specifically, let $\underline{V}_{ij}$ be the referenced vertex. Then the selection of the appropriate vertex is accomplished as follows:

<u>if</u> $i<0$ <u>then</u> use $\underline{V}_{0j}$

<u>else</u> <u>if</u> $i>m$ <u>then</u> use $\underline{V}_{mj}$;

<u>if</u> $j<0$ <u>then</u> use $\underline{V}_{i0}$

<u>else</u> <u>if</u> $j>n$ <u>then</u> use $\underline{V}_{in}$;

Note that an out-of-range value of the subscript i does not preclude an out-of-range value of the subscript j.

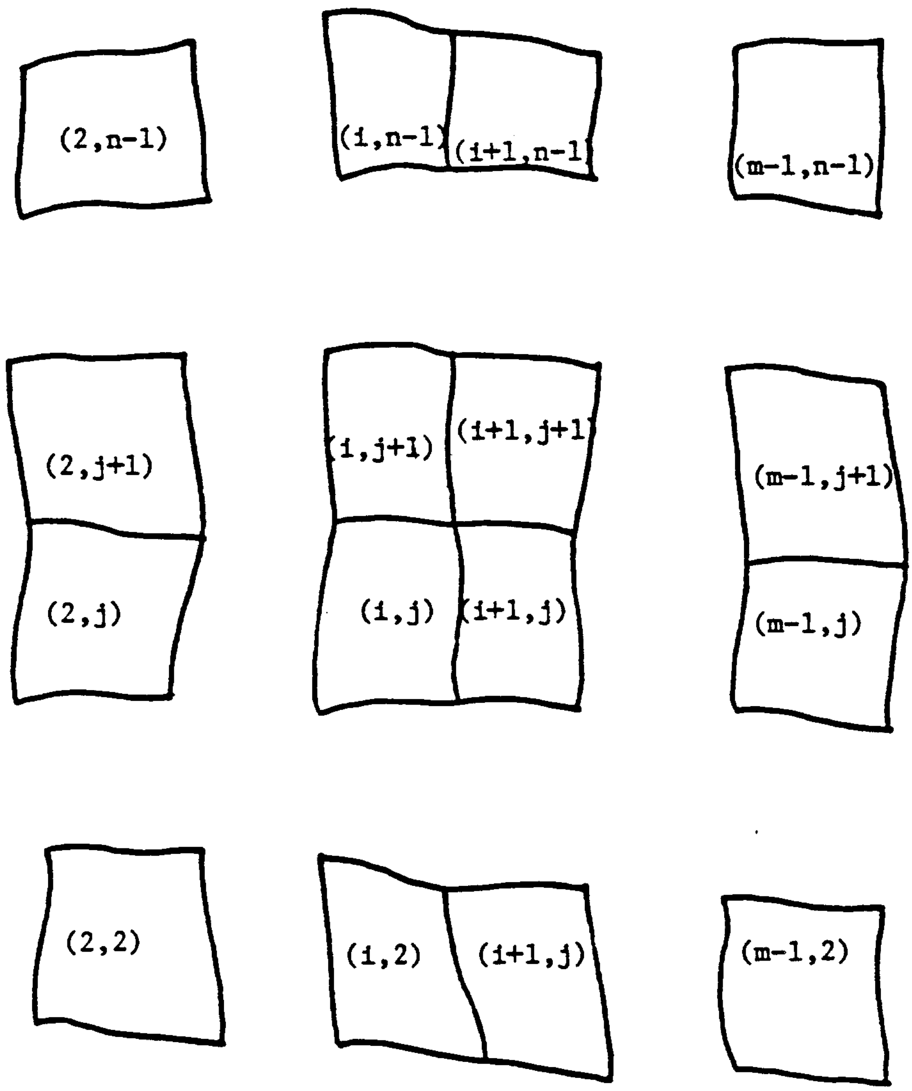


Figure X-1: Interior patches naturally defined by the control graph.

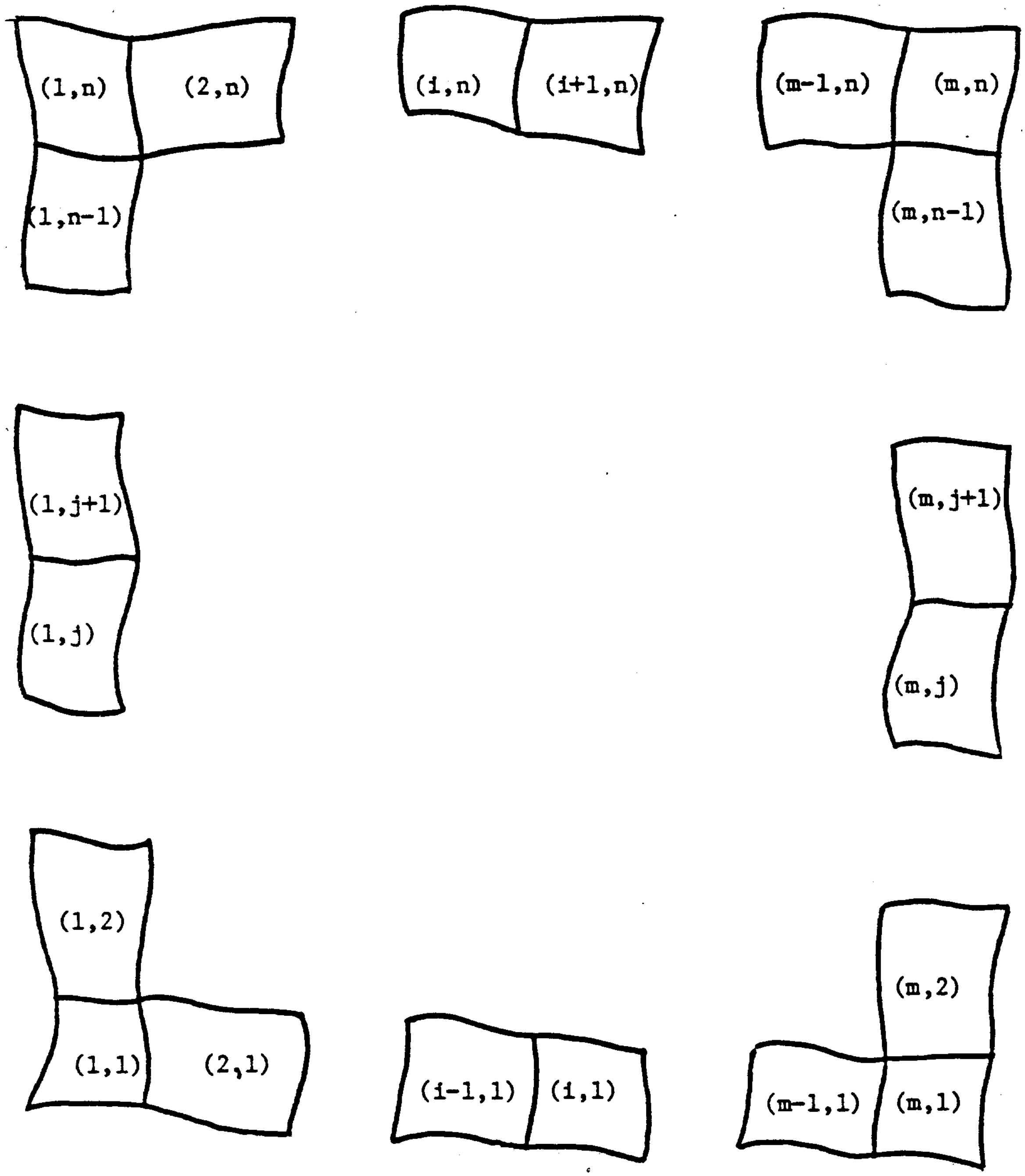


Figure X-2: Additional patches from double vertices boundary condition.

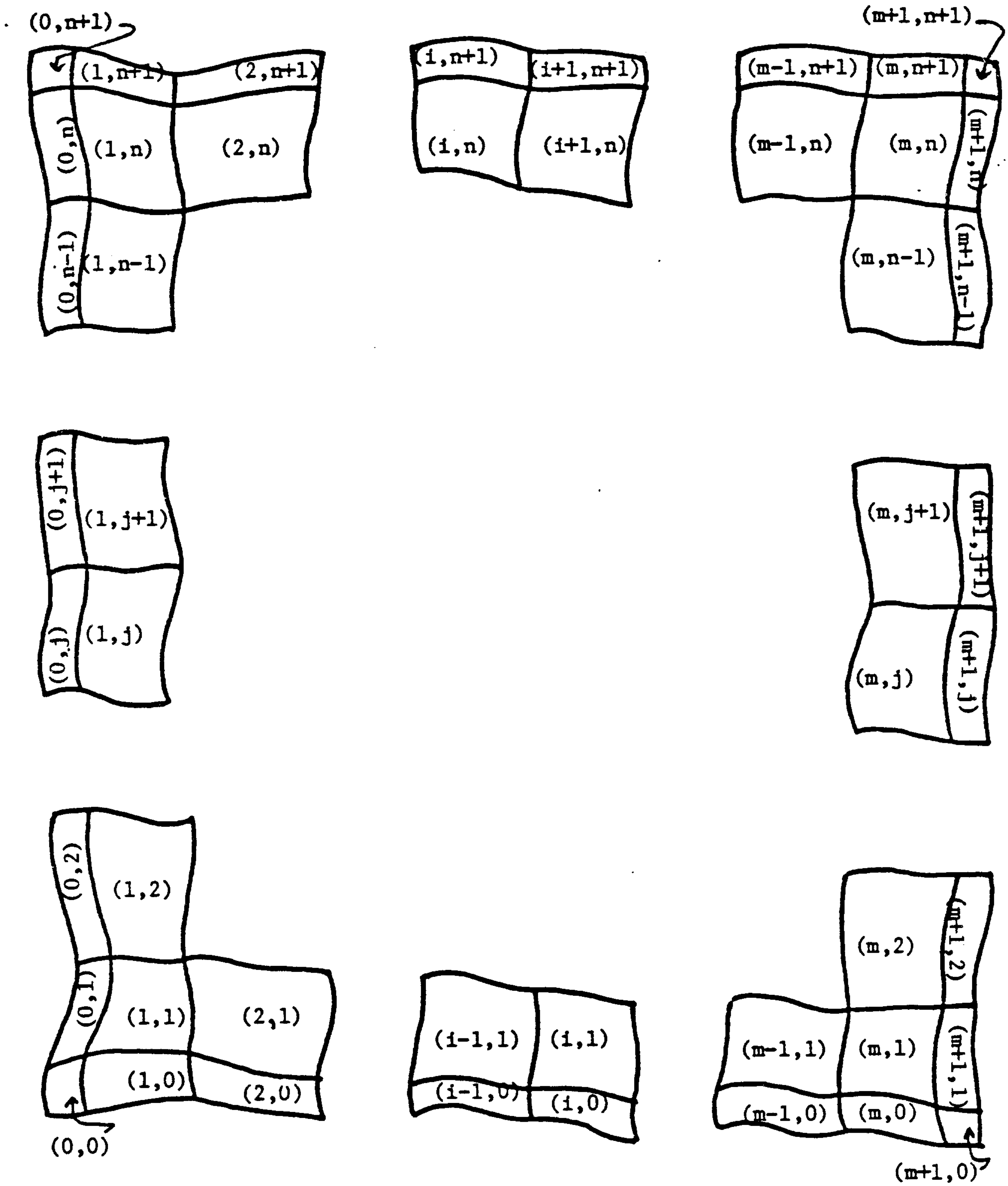


Figure X-3: Additional patches from triple vertices boundary condition.

X.2.ii. Triple vertices

This technique extends the double vertices technique by defining another set of additional surface patches around the periphery of those which were defined by the double vertices technique. This second set of additional patches is

$\underline{Q}_{0j}(u,v)$, $j = 0, 1, \ldots, n$;

$\underline{Q}_{i,n+1}(u,v)$, $i = 0, 1, \ldots, m$;

$\underline{Q}_{m+1,j}(u,v)$, $j = 1, \ldots, n+1$; and

$\underline{Q}_{i0}(u,v)$, $i = 1, \ldots, m+1$;

(Figure X-3) and they are defined by the same algorithm as that for the patches defined by the double vertices technique.

Using this algorithm for the patch $\underline{Q}_{0j}(u,v)$ yields

$$\underline{Q}_{0j}(u,v) = (b_{-2}(u) + b_{-1}(u) + b_0(u)) \sum_{s=-2}^{1} \underline{V}_{0,j+s} b_s(v) + b_1(u) \sum_{s=-2}^{1} \underline{V}_{1,j+s} b_s(v). \qquad (X.1)$$

X.3. Explanation of phantom vertices boundary conditions

The surface analogue of this type of end condition for a curve creates a set of phantom vertices around the boundaries of the original control graph. These phantom vertices are used to define additional surface patches (Figure X-4) around the patches which were naturally defined by the control graph. Analogous to this type of end condition for a curve, the phantom vertices are completely defined in terms of the original control vertices in order to satisfy some boundary condition. However, the direct specification of positions or of parametric first or second derivative vectors around the boundaries of the surface would be unwieldy. A convenient condition is to set the appropriate parametric second partial derivative vector to zero at the endpoint along each boundary curve between adjacent surface patches. The appropriate derivative is with respect to the parametric direction across the boundary.

Evaluating these derivatives at the appropriate parametric values, substituting the values of the second derivative of each basis function (as tabulated in Table X-3), and setting the resulting expression to zero yields an underspecified system containing $2m+2n+4$ equations in the $2m+2n+8$ unknown phantom vertices [1]. A solution is easily found yielding the following explicit expressions for each phantom vertex.

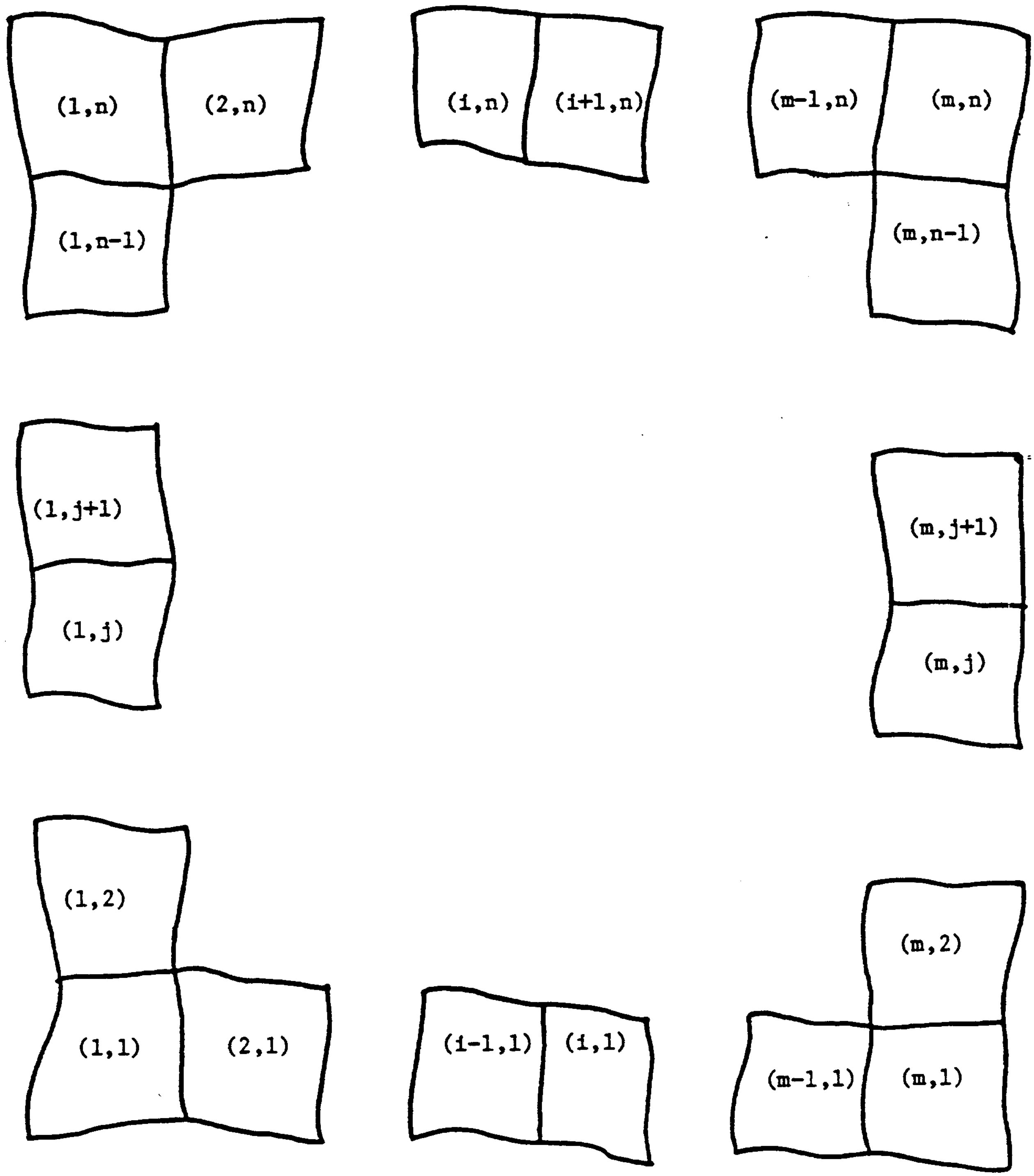


Figure X-4: Additional patches from phantom vertices boundary condition

$$
\begin{aligned}
&\underline{V}_{-1,-1} = 4\underline{V}_{00} - 2\underline{V}_{01} - 2\underline{V}_{10} + \underline{V}_{11} \\
&\underline{V}_{-1,j} = 2\underline{V}_{0j} - \underline{V}_{1j} \quad \text{for } j = 0, \ldots, n \\
&\underline{V}_{-1,n+1} = 4\underline{V}_{0n} - 2\underline{V}_{0,n-1} - 2\underline{V}_{1n} + \underline{V}_{1,n-1} \\
&\underline{V}_{i,n+1} = 2\underline{V}_{in} - \underline{V}_{i,n-1} \quad \text{for } i = 0, \ldots, m \\
&\underline{V}_{m+1,n+1} = 4\underline{V}_{mn} - 2\underline{V}_{m,n-1} - 2\underline{V}_{m-1,n} + \underline{V}_{m-1,n-1} \\
&\underline{V}_{m+1,j} = 2\underline{V}_{mj} - \underline{V}_{m-1,j} \quad \text{for } j = 0, \ldots, n \\
&\underline{V}_{m+1,-1} = 4\underline{V}_{m0} - 2\underline{V}_{m,1} - 2\underline{V}_{m-1,0} + \underline{V}_{m-1,1} \\
&\underline{V}_{i,-1} = 2\underline{V}_{i0} - \underline{V}_{i,1} \quad \text{for } i = 0, \ldots, m
\end{aligned}
\tag{X.2}
$$

k	$b_k(u)$	at u=0	at u=1
-2	$(-u^3 + 3u^2 - 3u + 1) / 6$	1/6	0
-1	$(3u^3 - 6u^2 + 4) / 6$	2/3	1/6
0	$(-3u^3 + 3u^2 + 3u + 1) / 6$	1/6	2/3
1	$u^3 / 6$	0	1/6

Table X-1: The uniform cubic B-spline basis functions.

k	$b_k^{(1)}(u)$	at u=0	at u=1
-2	$(-u^2 + 2u - 1) / 2$	-1/2	0
-1	$(-3u^2 + 4u) / 2$	0	-1/2
0	$(-3u^2 + 2u + 1) / 2$	1/2	0
1	$u^2 / 2$	0	1/2

Table X-2: The first derivative of the B-spline basis functions.

k	$b_k^{(2)}(u)$	at u=0	at u=1
-2	$-u + 1$	1	0
-1	$3u - 2$	-2	1
0	$-3u + 1$	1	-2
1	u	0	1

Table X-3: The second derivative of the B-spline basis functions.

XI. EVALUATION AND PERTURBATION OF A B-SPLINE SURFACE

XI.1. Surface Evaluation

A B-spline surface patch is described by equation (IX.4) as the parameters u and v both vary continuously from 0 to 1. To display it involves the computation of points on the surface for many different parametric values. The determination of a point on the patch requires the evaluation of the surface formulation at an appropriate (u,v) value. This entails the evaluation of the four basis functions at the value of u and of v, and then the computation of the sum which requires 20 multiplications and 15 additions for each coordinate.

Consider the computation of a set of points on one patch. Let $u = u_0, u_1, \ldots, u_p$ and $v = v_0, v_1, \ldots, v_q$ be the set of parametric values at which the patch is to be evaluated. Straightforward surface evaluation requires the evaluation of the four basis functions at all the parametric values plus 20(p+1)(q+1) multiplications and 15(p+1)(q+1) additions to determine each coordinate of the (p+1)(q+1) surface points on the patch. For a surface which is a mosaic of m by n patches, these computations would be performed mn times resulting in a very computationally intensive process. However, the computational requirements can be reduced by exploiting some of the properties of the B-spline representation.

Since the uniform basis functions are independent of the particular patch being computed, the evaluation of the basis functions can be made identical for all patches by restricting the set of parametric values to be the same for all of them. In this manner, the basis function evaluation can be performed only once and stored in a table. Algorithms to tabulate the uniform cubic B-spline basis functions were designed in Section VI.1. It should be noted that this does not impose any restriction on the selection of a particular set of parametric values.

After tabulating the basis functions, the surface points must be computed. However, this evaluation can be accomplished more efficiently by exploiting the tensor product form of the B-spline surface formulation. Observe that a point on the surface can be thought of as a point on a B-spline curve defined by an appropriate set of intermediate control vertices. Specifically, equation (IX.4) can be rewritten as

$$\underline{Q}_{ij}(u,v) = \sum_{r=-2}^{1} b_r(u)\, \underline{W}_{i+r,j}(v)$$

$$\text{where } \underline{W}_{i+r,j}(v) = \sum_{s=-2}^{1} \underline{V}_{i+r,j+s}\, b_s(v) \qquad \text{(XI.1)}$$

This mathematical formulation is the nucleus of the following algorithm to construct a uniform bicubic B-spline surface:

```
compute b_r(t) for each t in {u_0, u_1, ..., u_p} {v_0, v_1, ..., v_q}
     and r := -2 to 1;

for i := 1 to m do

     for j := 1 to n do

          for each v in {v_0, v_1, ..., v_q} do
               begin

               for r := -2 to 1 do
                    begin

                    W_r := V_{i+r,j-2}*b_{-2}(v);

                    for s := -1 to 1 do

                         W_r := W_r + V_{i+r,j+s}*b_s(v)

                    end;

               for each u in {u_0, u_1, ..., u_p} do
                    begin

                    Q_ij(u,v) := b_{-2}(u)*W_{-2};

                    for r := -1 to 1 do

                         Q_ij(u,v) := Q_ij(u,v) + b_r(u)*W_r

                    end

               end;
```

This algorithm to construct the entire surface requires the computation of the basis functions for all required values of the parameters u and v, plus $mn(q+1)[4*4 + (p+1)4] = 4mn(p+5)(q+1)$ multiplications and $mn(q+1)[4*3 + (p+1)3] = 3mn(p+5)(q+1)$ additions per coordinate.

XI.2. Surface Perturbation

Analogous to curve perturbation, the modification of an already-existing surface does not require the recomputation of the entire surface. Rather the modification of an existing surface can be accomplished much more efficiently by exploiting various properties of the B-spline representation.

Consider the consequences to an existing surface when the position of one control vertex is modified. Since a single control vertex affects only sixteen surface patches, the consequences of moving one vertex are confined to sixteen patches, and hence the movement of a control vertex requires the re-evaluation of only sixteen patches.

Furthermore, even these sixteen patches do not need to be completely recomputed. Although each of these patches is controlled by sixteen vertices, only one vertex has changed position. Therefore, the change in each of these segments is due only to the movement of one control vertex. Recalling the mathematical formulation for the (i,j)th surface patch given in equation (IX.4), the change in this patch $\underline{Q}^{\Delta}_{ij}(u,v)$ can be written as

$$\underline{Q}^{\Delta}_{ij}(u,v) = \sum_{r=-2}^{1} [b_r(u) \sum_{s=-2}^{1} \underline{V}^{\Delta}_{i+r,j+s}\, b_s(v)] \qquad \text{(XI.2)}$$

where $\underline{V}^{\Delta}_{ij}$ is the change in position of the control vertex $\underline{V}_{ij}$.

Assuming that the modified control vertex is the only one that has moved,

$$\underline{V}^{\Delta}_{ij} = 0 \text{ for } i \neq \hat{\imath} \text{ or } j \neq \hat{\jmath} \qquad \text{(XI.3)}$$

where $\underline{V}_{\hat{\imath}\hat{\jmath}}$ denotes the modified vertex.

Hence, equation (XI.2) contains only one nonzero term; thus, it reduces to

$$\underline{Q}^{\triangle}_{ij}(u,v) = b_r(u)\ \underline{V}^{\triangle}_{i+r,j+s}\ b_s(v) \qquad \text{(XI.4)}$$

where $i+r = \hat{i}$ and $j+s = \hat{j}$

Rewriting equation (XI.4) as

$$\underline{Q}^{\triangle}_{\hat{i}-r,\hat{j}-s}(u,v) = b_r(u)\ \underline{V}^{\triangle}_{\hat{i}\hat{j}}\ b_s(v) \qquad \text{(XI.5)}$$

for $r = -2, -1, 0, 1$ and $s = -2, -1, 0, 1$,

it is easily seen that the sixteen affected surface patches are

$$\underline{Q}_{ij}(u,v) \qquad \text{(XI.6)}$$

where $i = \hat{i}-r$ for $r = -2, -1, 0, 1$

and $j = \hat{j}-s$ for $s = -2, -1$ $0, 1.$

Hence the movement of control vertex $\underline{V}_{\hat{i}\hat{j}}$ perturbs the patches $\underline{Q}_{ij}(u,v)$ by

$$\underline{Q}^{\triangle}_{ij}(u,v) = b_{\hat{i}-i}(u)\ \underline{V}^{\triangle}_{\hat{i}\hat{j}}\ b_{\hat{j}-j}(v) \qquad \text{(XI.7)}$$

where i and j take on the values specified by equation (XI.6)

Since equation (XI.7) represents the change in the surface patch $\underline{Q}_{ij}(u,v)$, the new patch can be computed by incrementing the old patch by this change:

$$\underline{Q}^{new}_{ij}(u,v) = \underline{Q}^{old}_{ij}(u,v) + b_{\hat{i}-i}(u)\ \underline{V}^{\triangle}_{\hat{i}\hat{j}}\ b_{\hat{j}-j}(v) \qquad \text{(XI.8)}$$

The sixteen perturbed surface patches resulting from modifying the position of the control vertex $\underline{V}_{\hat{i}\hat{j}}$ can be computed by the following algorithm:

for j := $\hat{j}-1$ to $\hat{j}+2$ do

  for each v in $\{v_0, v_1, \ldots, v_q\}$ do

    begin

    $\underline{W} := \Delta\underline{V}_{\hat{i}\hat{j}} * b_{\hat{j}-j}(v)$;

    for i := $\hat{i}-1$ to $\hat{i}+2$ do

      for each u in $\{u_0, u_1, \ldots, u_p\}$ do

        $\underline{Q}_{ij}^{new}(u,v) := \underline{Q}_{ij}^{old}(u,v) + b_{\hat{i}-i}(u) * \underline{W}$

    end;

This algorithm requires a total of $4(4p+5)(q+1)$ multiplications and $16(p+1)(q+1)$ additions per component for the evaluation of the $(p+1)(q+1)$ surface points on the sixteen patches. Compare this to the $64(p+5)(q+1)$ multiplications and $48(p+5)(q+1)$ additions which would be required to completely recompute the sixteen affected patches, and to the $4mn(p+5)(q+1)$ multiplications and $3mn(p+5)(q+1)$ additions required for the recomputation of the entire surface.

As with curve perturbation, this algorithm is based on the assumption that no more than one control vertex has been moved. To perturb the surface by moving more than one vertex, they must be moved one at a time, and the algorithm must be performed after each such change. The algorithm is not valid if several vertices are modified simultaneously.

XII. DIFFERENCE TECHNIQUES FOR THE EVALUATION AND PERTURBATION OF A B-SPLINE SURFACE

XII.1. Background

The forward difference technique described in Section VII.1 for curves can also be applied to surfaces. For a bivariate function h(s,t), the forward difference with respect to the difference δ in the first parameter is

$$\Delta_{\delta,\epsilon}^{1,0}h(s,t) = h(s+\delta,t) - h(s,t) \qquad \text{(XII.1)}$$

The k-th forward difference with respect to the same parameter is defined recursively as

$$\Delta_{\delta,\epsilon}^{k,0}h(s,t) = \begin{cases} \Delta_{\delta,\epsilon}^{k-1,0}h(s+\ ,t) - \Delta_{\delta,\epsilon}^{k-1,0}h(s,t) & k = 1,\ 2,\ 3,\ \ldots \\ h(s,t) & k = 0 \end{cases} \qquad \text{(XII.2)}$$

The forward differences with respect to the second parameter are defined in an analogous manner.

XII.2. Surface evaluation

These definitions of forward differences can be used for the surface $\underline{Q}(u,v)$. From equation (XII.2),

$$\Delta^{k,0}_{\delta,\epsilon}\underline{Q}_{ij}(u,v) = \begin{cases} \Delta^{k-1,0}_{\delta,\epsilon}\underline{Q}_{ij}(u+\ ,v) - \Delta^{k-1,0}_{\delta,\epsilon}\underline{Q}_{ij}(u,v) & k=1,2,3,\ldots \\ \underline{Q}_{ij}(u,v) & k=0 \end{cases} \quad \text{(XII.3)}$$

and therefore

$$\Delta^{k-1,0}_{\delta,\epsilon}\underline{Q}_{ij}(u+\delta,v) = \Delta^{k-1,0}_{\delta,\epsilon}\underline{Q}_{ij}(u,v) + \Delta^{k,0}_{\delta,\epsilon}\underline{Q}_{ij}(u,v) \quad \text{(XII.4)}$$

for $k = 1, 2, 3, \ldots$

Substituting the surface formulation given in equation (IX.4) and using the recursive definition in equation (XII.2), it can be shown that

$$\Delta^{k,0}_{\delta,\epsilon}\underline{Q}_{ij}(u,v) = \sum_{r=-2}^{1} [\ {}^{k}b_r(u) \sum_{s=-2}^{1} \underline{V}_{i+r,j+s}\ b_s(v)] \quad \text{(XII.5)}$$

Analogous results hold for $\Delta^{0,1}_{\delta,\epsilon}\underline{Q}_{ij}(u,v)$.

Since

$$\Delta^{k,l}_{\delta,\epsilon}\underline{Q}_{ij}(u,v) = \Delta^{k,0}_{\delta,\epsilon}[\Delta^{0,l}_{\delta,\epsilon}\underline{Q}_{ij}(u,v)] \quad \text{(XII.6)}$$

for $k = 0, 1, 2, 3, \ldots$ and $l = 0, 1, 2, 3, \ldots$

hence

$$\Delta^{k,l}_{\delta,\epsilon}\underline{Q}_{ij}(u,v) = \sum_{r=-2}^{1} \Delta^{k}_{\delta}b_r(u) \sum_{s=-2}^{1} [\underline{V}_{i+r,j+s}\ \Delta^{l}_{\epsilon}b_s(v)] \quad \text{(XII.7)}$$

for $k = 0, 1, 2, 3, \ldots$ and $l = 0, 1, 2, 3, \ldots$

Recalling from equation (VII.9) that the fourth and succeeding forward differences of the basis functions are zero,

$$\Delta^{k,l}_{\delta,\epsilon}\underline{Q}_{ij}(u,v) = 0 \quad \text{for } k = 4, 5, 6, \ldots \text{ or } l = 4, 5, 6, \ldots \quad \text{(XII.8)}$$

From equations (XII.4) and (XII.8) with k=4 it can be seen that

$$\Delta^{3,0}_{\delta,\epsilon}\underline{Q}_{ij}(u+\ ,v) = \Delta^{3,0}_{\delta,\epsilon}\underline{Q}_{ij}(u,v) \qquad \text{(XII.9)}$$

and therefore $\Delta^{3,0}_{\delta,\epsilon}\underline{Q}_{ij}(u,v)$ and similarly $\Delta^{0,3}_{\delta,\epsilon}\underline{Q}_{ij}(u,v)$ are constant. Thus,

$$\begin{aligned}
\underline{Q}_{ij}(u+\delta,v) &= \underline{Q}_{ij}(u,v) + \Delta^{1,0}_{\delta,\epsilon}\underline{Q}_{ij}(u,v) \\
\Delta^{1,0}_{\delta,\epsilon}\underline{Q}_{ij}(u+\ ,v) &= \Delta^{1,0}_{\delta,\epsilon}\underline{Q}_{ij}(u,v) + \Delta^{2,0}_{\delta,\epsilon}\underline{Q}_{ij}(u,v) \\
\Delta^{2,0}_{\delta,\epsilon}\underline{Q}_{ij}(u+\ ,v) &= \Delta^{2,0}_{\delta,\epsilon}\underline{Q}_{ij}(u,v) + \Delta^{3,0}_{\delta,\epsilon}\underline{Q}_{ij}(0,v)
\end{aligned} \qquad \text{(XII.10)}$$

and similarly for the differences with respect to the second parameter.

The initialization requires the values of $\Delta^{k,l}_{\delta,\epsilon}\underline{Q}_{ij}(0,0)$, for $k = 0, 1, 2, 3$ and $l = 0, 1, 2, 3$, which can be computed from equation (XII.7). This requires the values of the forward differences of the four basis functions in each direction. The $\Delta^{k}_{\delta}b_r(0)$ are tabulated in Table VII-1 in terms of δ and the analogous tabulation can be prepared for the $\Delta^{l}_{\epsilon}b_s(0)$ in terms of the difference in the other direction, ϵ.

This difference technique can be used to evaluate surface points that are equally spaced in each direction. Let

$$\begin{aligned}
u_a &= a\delta \text{ where } \delta = 1/p \\
v_b &= b\epsilon \text{ where } \epsilon = 1/q
\end{aligned} \qquad \text{(XII.11)}$$

Then the surface can be constructed by the following algorithm:

```
Compute \Delta_\delta^k b_r(0) for k := 0 to 3 and r := -2 to 1;

Compute \Delta_\epsilon^l b_s(0) for l := 0 to 3 and s := -2 to 1;

for i := 1 to m do

    for j := 1 to n do
        begin (* patch Q_ij(u,v) *)

        compute \Delta_{\delta,\epsilon}^{k,l} Q_ij(0,0) for k := 0 to 3 and l := 0 to 3;

        for b := 1 to q do
            begin(*compute differences along curve Q_ij(0,v)*)

            \Delta_{\delta,\epsilon}^{0,0} Q_ij(0,v_b) :=
                \Delta_{\delta,\epsilon}^{0,0} Q_ij(0,v_{b-1}) + \Delta_{\delta,\epsilon}^{0,1} Q_ij(0,v_{b-1});
            \Delta_{\delta,\epsilon}^{0,1} Q_ij(0,v_b) :=
                \Delta_{\delta,\epsilon}^{0,1} Q_ij(0,v_{b-1}) + \Delta_{\delta,\epsilon}^{0,2} Q_ij(0,v_{b-1});
            \Delta_{\delta,\epsilon}^{0,2} Q_ij(0,v_b) :=
                \Delta_{\delta,\epsilon}^{0,2} Q_ij(0,v_{b-1}) + \Delta_{\delta,\epsilon}^{0,3} Q_ij(0,0)

            end (*compute differences along curve Q_ij(0,v)*);

        for a := 1 to p do
            begin (* fix u=u_a *)

            for l := 0 to 3 do
                begin (* compute differences to start next
                                                  curve *)

                \Delta_{\delta,\epsilon}^{0,l} Q_ij(u_a,0) :=
                    \Delta_{\delta,\epsilon}^{0,l} Q_ij(u_{a-1},0) + \Delta_{\delta,\epsilon}^{1,l} Q_ij(u_{a-1},0);
                \Delta_{\delta,\epsilon}^{1,l} Q_ij(u_a,0) :=
                    \Delta_{\delta,\epsilon}^{1,l} Q_ij(u_{a-1},0) + \Delta_{\delta,\epsilon}^{2,l} Q_ij(u_{a-1},0);
                \Delta_{\delta,\epsilon}^{2,l} Q_ij(u_a,0) :=
                    \Delta_{\delta,\epsilon}^{2,l} Q_ij(u_{a-1},0) + \Delta_{\delta,\epsilon}^{3,l} Q_ij(0,0)

                end (* compute differences to start next
                                                  curve *);
```

```
for b := 1 to q do
    begin (* compute differences along curve
                                         Q_ij(u_a,v) *)
    Δ^{0,0}_{δ,ε}Q_ij(u_a,v_b) :=
        Δ^{0,0}_{δ,ε}Q_ij(u_a,v_{b-1}) + Δ^{0,1}_{δ,ε}Q_ij(u_a,v_{b-1});
    Δ^{0,1}_{δ,ε}Q_ij(u_a,v_b) :=
        Δ^{0,1}_{δ,ε}Q_ij(u_a,v_{b-1}) + Δ^{0,2}_{δ,ε}Q_ij(u_a,v_{b-1});
    Δ^{0,2}_{δ,ε}Q_ij(u_a,v_b) :=
        Δ^{0,2}_{δ,ε}Q_ij(u_a,v_{b-1}) + Δ^{0,3}_{δ,ε}Q_ij(u_a,0)
    end (* compute differences along curve
                                         Q_ij(u_a,v) *)

  end (* fix u=u_a *)

end (* patch Q_ij(u,v) *);
```

The computation of $\Delta^{k,l}_{\delta,\epsilon}\underline{Q}_{ij}(0,0)$ can be performed efficiently by exploiting the tensor product in a similar manner as was done for the evaluation of a surface point in Section XII.1. From equation (XII.7) with u=v=0,

$$\Delta^{k,l}_{\delta,\epsilon}\underline{Q}_{ij}(0,0) = \sum_{r=-2}^{1} \Delta^{k}_{\delta}b_r(0)\ \underline{W}_{i+r,j,l} \qquad \text{(XII.12)}$$

$$\text{where } \underline{W}_{i+r,j,l} = \sum_{s=-2}^{1} \underline{V}_{i+r,j+s}\,\Delta^{l}_{\epsilon}b_s(0)$$

Using this formulation, the following algorithm can be used to compute $\Delta^{k,l}_{\delta,\epsilon}\underline{Q}_{ij}(0,0)$ for k = 0, 1, 2, 3 and l = 0, 1, 2, 3:

for l := 0 **to** 3 **do**

begin

for r := -2 **to** 1 **do**

begin

$\underline{W}_r := \underline{V}_{i+r,j-2} * \Delta^{l}_{\epsilon} b_{-2}(0)$;

for s := -1 **to** 1 **do** $\underline{W}_r := \underline{W}_r + \underline{V}_{i+r,j+s} * \Delta^{l}_{\epsilon} b_{s}(0)$

end;

for k := 0 **to** 3 **do**

begin

$\Delta^{k,l}_{\delta,\epsilon} \underline{Q}_{ij}(0,0) := \Delta^{k}_{\delta} b_{-2}(0) * \underline{W}_{-2}$;

for r := -1 **to** 1 **do**

$\Delta^{k,l}_{\delta,\epsilon} \underline{Q}_{ij}(0,0) := \Delta^{k,l}_{\delta,\epsilon} \underline{Q}_{ij}(0,0) + \Delta^{k}_{\delta} b_{r}(0) * \underline{W}_r$

end

end;

This algorithm requires 128 multiplications and 96 additions per coordinate to perform this computation. The complete algorithm to construct the entire surface thus requires the computation of $\Delta^{k}_{\delta} b_r(0)$ for k = 0, 1, 2, 3 and r = -2, -1, 0, 1, and $\Delta^{l}_{\epsilon} b_s(0)$ for l = 0, 1, 2, 3 and r = -2, -1, 0, 1, plus 128mn multiplications and mn[96 + q*3 + p(4*3+q*3)] = 3mn[32 + q + p*(4+q)] additions per coordinate.

XII.3. Surface perturbation

The forward difference technique can also be used to perturb an already existing surface. Analogous to the use of this technique for curve perturbation, the algorithm for surface perturbation requires two modifications to the algorithm for surface evaluation. First, only sixteen patches need to be re-evaluated. Second, the computation of $\Delta^{k,l}_{\delta,\epsilon}\underline{Q}_{ij}(0,0)$ can be accomplished easily by incrementing it by the appropriate change in value. This approach yields the following algorithm to compute $\Delta^{k,l}_{\delta,\epsilon}\underline{Q}_{ij}(0,0)$ for k = 0, 1, 2, 3 and l = 0, 1, 2, 3:

```
for l := 0 to 3 do
    begin
    W := ΔV_ij * l b_ĵ-j(0);
    for k := 0 to 3 do
        Δ^{k,l}_{δ,ε} Q_ij(0,0) := Δ^{k,l}_{δ,ε} Q_ij(0,0) + Δ^k_δ b_î-i(0)*W
    end;
```

This algorithm requires 20 multiplications and 16 additions per coordinate. Thus, the complete algorithm to perturb the surface requires 16*20 = 320 multiplications and 16[16 + q*3 + p(4*3+q*3)] = 16[16 + 3(q + p*(4+q))] additions per coordinate.

XII.4. Conclusion

As with the application of forward difference techniques to curves, the advantage is the reduction of the number of necessary multiplications. However, this saving is at the expense of cumulative error.

XIII. CONCLUSION

The intent of this paper is to provide sufficient information to understand and implement parametric cubic B-spline curves and bicubic B-spline surfaces in the case of uniformly spaced parametric knot values without multiplicity. The parametric representation was discussed, the properties of the B-spline representation were described, and a detailed derivation of the B-spline basis functions was presented. A systematic discussion of various choices of end condition and boundary condition specification was also provided so that the B-spline user can decide which technique is appropriate for a particular application. A distinction was made between the evaluation of a new, nonexistent curve or surface, and the perturbation of an already-existing one. Efficient algorithms, which exploit the repetitiveness of the uniform case, were designed and analyzed for B-spline basis function evaluation, and for the evaluation and perturbation of both B-spline curves and surfaces. When multiplications are much less desirable than additions, finite difference techniques provide another computational approach which is probably the method of choice as regards efficiency. Algorithms for that situation which accomplish the evaluation and perturbation of both curves and surfaces were also designed and analyzed.

The methods and economics of B-spline evaluation vary considerably with the generality of the implementation, and it is only by restricting attention to this case that the computational savings realized in this paper can be achieved. The goal of this paper was to give a rather thorough treatment of this specialized yet frequently used case. There is a large body of literature which has grown rapidly since a practical algorithm with stable and efficient characteristics for computing B-splines was first availed to the heretofore stifled applications area by Cox [13] and de Boor [6]. Most of these results, which are independent of the ones presented herein, are concerned with the more general problem, and the interested reader is referred to [7, 8, 16, 22, 23].

REFERENCES

1. Barsky, Brian A. A Method for Describing Curved Surfaces by Transforming between Interpolatory Spline and B-spline Representations, Master's Thesis, Cornell University, Ithaca, N.Y., January 1979.

2. Barsky, Brian A. "End Conditions and Boundary Conditions for Uniform B-spline Curve and Surface Representations", Computers in Industry. Special Steven A. Coons Memorial Issue, Vol. 3, No. 1, 2, March, June 1982, pp. 17-29.

3. Barsky, Brian A. and Greenberg, Donald P. "An Interactive Surface Representation System Using a B-spline Formulation with Interpolation Capability", Computer-Aided Design, Vol. 14, No. 4, July 1982, pp. 187-194.

4. Barsky, Brian A. and Thomas, Spencer W. "TRANSPLINE Curve Representation System", Tech. Report No. UUCS-80-104, Department of Computer Science, University of Utah, April 1980.

5. Barsky, Brian A. and Thomas, Spencer W. "TRANSPLINE -- A System for Representing Curves Using Transformations among Four Spline Formulations", The Computer Journal, Vol.24, No. 3, August 1981, pp. 271-277.

6. de Boor, Carl. "On Calculating with B-splines", Journal of Approximation Theory, Vol. 6, 1972, pp. 50-62.

7. de Boor, Carl. "Splines as Linear Combinations of B-splines. A Survey". In Approximation Theory II, edited by Lorentz, G. G.; Chui, C. K. and Schumaker, L. L., Academic Press, New York, 1976, pp. 1-47.

8. de Boor, Carl. A Practical Guide to Splines, Springer-Verlag, Applied Mathematical Sciences Vol. 27, 1978.

9. Clark, James H. 3-D Design of Free-Form B-spline Surfaces, Ph.D. Thesis, University of Utah, Salt Lake City, Utah, September 1974.

10. Clark, James H. "Designing Surfaces in 3-D", Communications of the ACM, Vol. 19, No. 8, August 1976, pp. 454-460.

11. Coons, Steven A. "Surfaces for Computer-Aided Design of Space Forms", Tech. Report No. MAC-TR-41, Project MAC, M.I.T., June 1967.

12. Coons, Steven A. "Surface Patches and B-spline Curves". In Computer Aided Geometric Design, edited by Barnhill, Robert E. and Riesenfeld, Richard F., Academic Press, 1974, pp. 1-16.

13. Cox, Morris G. "The Numerical Evaluation of B-splines", Tech. Report No. 4, Division of Numerical Analysis and Computing, National Physical Laboratory, Teddington, Middlesex, England, August 1971. Also in J. Inst. Maths. Applics., Vol. 10, 1972, pp. 134-149.

14. Curry, H. B. and Schoenberg, I. J. "On Spline Distributions and their Limits: the Polya Distribution Functions, Abstract 380t", Bulletin of the American Math. Society, Vol. 53, 1947, pp. 1114.

15. Curry, H. B. and Schoenberg, I. J. "On Polya Frequency Functions IV: The Fundamental Spline Functions and their Limits", Journal d'Analyse Mathematique, Vol. 17, 1966, pp. 71-107.

16. Faux, Ivor D. and Pratt, Michael J. Computational Geometry for Design and Manufacture, Ellis Horwood Ltd., 1979.

17. Forrest, A. Robin. "Mathematical Principles for Curve and Surface Representation". In Proceedings of Curved Surfaces in Engineering, Cambridge, England, 15-17th March 1972, pp. 5-13.

18. Gordon, William J. and Riesenfeld, Richard F. "B-spline Curves and Surfaces". In Computer Aided Geometric Design, edited by Barnhill, Robert E. and Riesenfeld, Richard F., Academic Press, 1974, pp. 95-126.

19. Lane, Jeffrey M. and Riesenfeld, Richard F. "A Geometric Proof of the Variation Diminishing Property of B-spline Approximation", Journal of Approximation Theory. To appear.

20. Prenter, Paddy M. Splines and Variational Methods, John Wiley & Sons, Inc., 1975.

21. Riesenfeld, Richard F. Applications of B-spline Approximation to Geometric Problems of Computer-Aided Design, Ph.D. Thesis, Syracuse University, Syracuse, N.Y., May 1973. Available as Tech. Report No. UTEC-CSc-73-126, Department of Computer Science, University of Utah.

22. Riesenfeld, Richard F. "Nonuniform B-spline Curves". In Proc. 2nd USA-Japan Computer Conference, AFIPS, 1975, pp. 551-555.

23. Rogers, David F. and Adams, J. Alan. Mathematical Elements for Computer Graphics, McGraw-Hill, 1976.

24. Schoenberg, Isaac J. "Contributions to the Problem of Approximation of Equidistant Data by Analytic Functions", Quarterly Applied Math., Vol. 4, No. 1, 1946, pp. 45-99 and 112-141.

25. Schoenberg, Isaac J. "On Spline Functions". In Inequalities, edited by Shisha, Oved, Academic Press, 1967, pp. 255-291. With supplement by T. N. E. Greville.

26. Wu, Sheng-Chuan and Abel, John F. "Representation and Discretization of Arbitrary Surfaces for Finite Element Shell Analysis", International Journal for Numerical Methods in Engineering, Vol. 14, No. 6, 1979, pp. 813-836.

BIBLIOGRAPHY ON SPLINES

1. Adams, J. Alan. "A Comparison of Methods for Cubic Spline Curve Fitting", Computer-Aided Design, Vol. 6, No. 1, January 1974, pp. 1-9.

2. Ahlberg, J. H. and Nilson, E. N. "Convergence Properties of the Spline Fit", SIAM J. Appl. Math., Vol. 11, 1963, pp. 95-104.

3. Ahlberg, J. H.; Nilson, E. N.; and Walsh, J. L. "Fundamental Properties of Generalized Splines". In Proceedings of the National Academy of Sciences, U.S.A., Vol. 52, 1964, pp. 1412-1419.

4. Ahlberg, J. H.; Nilson, E. N.; and Walsh, J. L. "Convergence Properties of Generalized Splines". In Proceedings of the National Academy of Sciences, U.S.A., 1965, pp. 344-350.

5. Ahlberg, J. H.; Nilson, E. N.; and Walsh, J. L. The Theory of Splines and their Applications, Academic Press, New York, 1967.

6. Ahuja, D. V. "An Algorithm for Generating Spline-Like Curves", IBM Systems Journal, Vol. 7, No. 3/4, 1968, pp. 206-217.

7. Ahuja, D. V. and Coons, S. A. "Geometry for Construction and Display", IBM Systems Journal, Vol. 7, No. 3/4, 1968, pp. 188-205.

8. Anselone, P. J. and Laurent, P. J. "A General Method for the Construction of Interpolating or Smoothing Spline-Functions", Numer. Anal., Vol. 12, 1968, pp. 66-82.

9. Armit, Andrew P. "Example of an Existing System in University Research. Multipatch and Multiobject Design Systems". In Proceedings of the Royal Society of London, Vol. A321, 1971, pp. 235-242.

10. Armit, Andrew P. "Interactive 3D Shape Design -- Multipatch and Multiobject". In Proceedings of Curved Surfaces in Engineering, Churchill College, edited by Brown, I. J., IPC Science and Technology Press, Ltd., Cambridge, England, 15-17th March 1972.

11. Atkinson, K. E. "On the Order of Convergence of Natural Cubic Spline Interpolation", SIAM Journal on Numerical Analysis, Vol. 5, 1968, pp. 89-101.

12. Barnhill, Robert E. "Representation and Approximation of Surfaces". In Mathematical Software III, edited by Rice, John. R., Academic Press, New York, 1977, pp. 68-119.

13. Barnhill, Robert E.; Brown, James H.; and Klucewicz, I. Michele. "A New Twist for Computer Aided Geometric Design", Computer Graphics and Image Processing, Vol. 8, No. 1, August 1978, pp. 78-91.

14. Barnhill, Robert E. and Riesenfeld, Richard F. (editors). Computer Aided Geometric Design, Academic Press, New York, 1974.

15. Barnhill, Robert E. and Riesenfeld, Richard F. "Surface Representation for Computer Aided Design". In Data Structures, Computer Graphics, and Pattern Recognition, edited by Klinger, A.; Fu, K. S.; and Kunii, T. L., Academic Press, New York, 1977, pp. 413-426.

16. Barrow, D. L. and Smith, P. W. "Asymptotic Properties of Best $L_2[0,1]$ Approximation by Splines with Variable Knots". To appear.

17. Barsky, Brian A. A Method for Describing Curved Surfaces by Transforming between Interpolatory Spline and B-spline Representations, Master's Thesis, Cornell University, Ithaca, N.Y., January 1979.

18. Barsky, Brian A. The Beta-spline: A Local Representation Based on Shape Parameters and Fundamental Geometric Measures, Ph.D. Thesis, University of Utah, Salt Lake City, Utah, December 1981.

19. Barsky, Brian A. "Computer Aided Geometric Design: A Bibliography with keywords and Classified Index", IEEE Computer Graphics and Applications, Vol. 1, No. 3, July 1981, pp. 67-109. Also reprinted in ACM Computer Graphics, Vol. 16, No. 1, May 1982, pp. 119-159.

20. Barsky, Brian A. "Exponential and Polynomial Methods for Applying Tension to an Interpolating Spline Curve", Computer Vision, Graphics, and Image Processing. To appear.

21. Barsky, Brian A. "End Conditions and Boundary Conditions for Uniform B-spline Curve and Surface Representations", Computers in Industry. Special Steven A. Coons Memorial Issue, Vol. 3, No. 1, 2, March, June 1982, pp. 17-29.

22. Barsky, Brian A. and Greenberg, Donald P. "Determining a Set of B-spline Control Vertices to Generate an Interpolating Surface", Computer Graphics and Image Processing, Vol. 14, No. 3, November 1980, pp. 203-226.

23. Barsky, Brian A. and Greenberg, Donald P. "An Interactive Surface Representation System Using a B-spline Formulation with Interpolation Capability", Computer-Aided Design, Vol. 14, No. 4, July 1982, pp. 187-194.

24. Barsky, Brian A. and Thomas, Spencer W. "TRANSPLINE Curve Representation System", Tech. Report No. UUCS-80-104, Department of Computer Science, University of Utah, Salt Lake City, Utah, April 1980.

25. Barsky, Brian A. and Thomas, Spencer W. "TRANSPLINE -- A System for Representing Curves Using Tranformations among Four Spline Formulations", The Computer Journal, Vol. 24, No. 3, August 1981, pp. 271-277.

26. Bezier, Pierre E. Emploi des machines a commande numerique, Masson et Cie., Paris, 1970. Translated by Forrest, A. Robin and Pankhurst, Anne F. as Numerical Control -- Mathematics and Applications, John Wiley and Sons, Ltd., London, 1972.

27. Bezier, Pierre E. "Mathematical and Practical Possibilities of UNISURF". In Computer Aided Geometric Design, edited by Barnhill, Robert E. and Riesenfeld, Richard F., Academic Press, New York, 1974.

28. Birkhoff, Garrett and de Boor, Carl. "Error Bounds for Spline Interpolation", J. Math. Mech., Vol. 13, 1964, pp. 827-836.

29. Birkhoff, Garrett and de Boor, Carl. "Piecewise Polynomial Interpolation and Approximation". In Approximation of Functions, American Elsevier, New York, 1965, pp. 164-190.

30. Birkhoff, G. and Priver, A. "Hermite Interpolation Error for Derivatives", Journal of Mathematics and Physics, Vol. 46, 1967, pp. 440-447.

31. Birkhoff, Garrett; Schultz, Martin H.; and Varga, Richard S. "Piecewise Hermite Interpolation in One and Two Variables with Applications to Partial Differential Equations", Numerische Mathematik, Vol. 11, 1968, pp. 232-256.

32. Boehm, Wolfgang. "Parameterdarstellung, Kubisher und Bikubisher Splines", Computing, Vol. 17, 1976, pp. 87-92.

33. Boehm, Wolfgang. "Ueber die Konstruktion von B-Spline-Kurven", Computing, Vol. 18, No. 2, 1977, pp. 161-167.

34. Boehm, Wolfgang. "Cubic B-spline Curves and Surfaces in Computer-Aided Geometric Design", Computing, Vol. 19, No. 1, 1977, pp. 29-34.

35. Boehm, Wolfgang. "Inserting New Knots into B-spline Curves", Computer-Aided Design, Vol. 12, No. 4, July 1980, pp. 199-201.

36. Boehmer, K. Spline-Funktionen, Teubner-Studienbucher, Stuttgart, 1974.

37. Boehmer, K.; Meinardus, G.; and Schempp, W. Tagung ueber Spline-Funktionen, Bibliographisches Institut, Mannheim, 1974.

38. Boere, H. The Design of Surfaces with the Coons' Method, Ph.D. Thesis, Technical University of Delft, The Netherlands, November 1970.

39. de Boor, Carl. "Bicubic Spline Interpolation", Journal of Mathematics and Physics, Vol. 41, 1962, pp. 212-218.

40. de Boor, Carl. "On Uniform Approximation by Splines", Journal of Approximation Theory, No. 1, 1968, pp. 219-235.

41. de Boor, Carl. "On the Convergence of Odd Degree Spline Interpolation", Journal of Approximation Theory, Vol. 1, 1968, pp. 452-463.

42. de Boor, Carl. "Subroutine Package for Calculating with B-splines", Tech. Report No. LA-4728-MS, Los Alamos Scientific Laboratory, August 1971.

43. de Boor, Carl. "On Calculating with B-splines", Journal of Approximation Theory, Vol. 6, No. 1, July 1972, pp. 50-62.

44. de Boor, Carl. "Good Approximation by Splines with Variable Knots". In Spline Functions and Approximation Theory, edited by Meir, A. and Sharma, A., Birkhauser Verlag Basel, 1973, pp. 57-72.

45. de Boor, Carl. "Good Approximation by Splines with Variable Knots. II". In Numerical Solution of Differential Equations, edited by Watson, G. A., Springer Verlag, 1974, pp. 12-20. In Lecture Notes in Math. No. 363.

46. de Boor, Carl. "On Bounding Spline Interpolation", Journal of Approximation Theory, No. 14, 1975, pp. 191-203.

47. de Boor, Carl. "Splines as Linear Combinations of B-splines. A Survey". In Approximation Theory II, edited by Lorentz, G. G.; Chui, C. K.; and Schumaker, L. L., Academic Press, New York, 1976, pp. 1-47.

48. de Boor, Carl. "Total Positivity of the Spline Collocation Matrix", Ind. Univ. J. Math., No. 25, 1976, pp. 541-551.

49. de Boor, Carl. "On Local Linear Functionals which Vanish at All B-splines but One". In Theory of Approximation with Applications, edited by Law, A. G. and Sahney, B. N., Academic Press, New York, 1976, pp. 120-145.

50. de Boor, Carl. "Package for Calculating with B-splines", SIAM Journal on Numerical Analysis, Vol. 14, No. 3, 1977, pp. 441-472.

51. de Boor, Carl. A Practical Guide to Splines, Springer-Verlag, Applied Mathematical Sciences Vol. 27, 1978.

52. de Boor, Carl. "A Taut Cubic Spline". To appear.

53. de Boor, Carl and Lynch, Robert E. "On Splines and their Minimum Properties", J. Math. Mech., Vol. 15, No. 6, 1966, pp. 953-969.

54. de Boor, Carl and Rice, John R. "Least Squares Cubic Spline Approximation, I -- Fixed Knots; II -- Variable Knots", CSD TR No. 20 and 21, Purdue University, West Lafayette, Indiana, 1968.

55. de Boor, Carl and Swartz, Blair K. "Piecewise Monotone Interpolation", Journal of Approximation Theory, Vol. 21, No. 4, December 1977, pp. 411-416.

56. Bramble, J. and Hilbert, S. "Bounds for a Class of Linear Functionals with Application to Hermite Interpolation", Numerische Mathematik, Vol. 16, 1971, pp. 362-369.

57. Brewer, John A. and Anderson, David C. "Visual Interaction with Overhauser Curves and Surfaces". In Proceedings of SIGGRAPH '77, ACM, July 1977, pp. 132-137.

58. Briggs, I. C. "Machine Contouring Using Minimum Curvature", Geophysics, Vol. 39, No. 1, February 1974, pp. 39-48.

59. Buneman, O. "On-Line Spline Fitting", SUIPR Report No. 507, Stanford Institute for Plasma Research, Stanford, California, February 1973.

60. Burchard, H. G. "Splines (with Optimal Knots) are Better", J. Applicable Analysis, Vol. 3, 1974, pp. 309-319.

61. Butterfield, Kenneth R. "The Computation of All the Derivatives of a B-spline Basis", J. Inst. Maths. Applics., Vol. 17, 1976, pp. 15-25.

62. Butterfield, Kenneth R. Introductory Theory Associated with Splines and the General Intersection of Three Spline Surfaces, Ph.D. Thesis, Brunel University, Uxbridge, Middlesex, 1978.

63. Carasso, C. and Laurent, P. J. "On the Numerical Construction and the Practical Use of Interpolating Spline-Functions". In Proceedings of the IFIP Congress, Information Processing '68, Edinburgh, Vol. 1, Mathematics, Software, North-Holland Publishing Company, Amsterdam, 1969, pp. 86-89.

64. Carlson, R. E. and Hall, C. A. "On Piecewise Polynomial Interpolation in Rectangular Polygons", Journal of Approximation Theory, Vol. 4, 1971, pp. 37-53. Also WAPD-T-2160, Westinghouse Bettis Nuclear Power Station, November 1969.

65. Catmull, Edwin E. and Clark, James H. "Recursively Generated B-spline Surfaces on Arbitrary Topological Meshes", Computer-Aided Design, Vol. 10, No. 6, November 1978, pp. 350-355.

66. Catmull, Edwin E. and Rom, Raphael J. "A Class of Local Interpolating Splines". In Computer Aided Geometric Design, edited by Barnhill, Robert E. and Riesenfeld, Richard F., Academic Press, New York, 1974, pp. 317-326.

67. Chaikin, George M. "An Algorithm for High-Speed Curve Generation", Computer Graphics and Image Processing, Vol. 3, 1974, pp. 346-349.

68. Cheney, E. W. and Schurer, F. "A Note on the Operators Arising in Spline Approximation", Journal of Approximation Theory, Vol. 1, 1968, pp. 94-102.

69. Clark, James H. 3-D Design of Free-Form B-spline Surfaces, Ph.D. Thesis, University of Utah, Salt Lake City, Utah, September 1974. Also Tech. Report No. UTEC-CSc-74-120, Department of Computer Science, University of Utah.

70. Clark, James H. "Some Properties of B-splines". In Proceedings of the 2nd USA-Japan Computer Conference, AFIPS, Montvale, N.J., 26-28 August 1975, pp. 542-545.

71. Clark, James H. "Designing Surfaces in 3-D", Communications of the ACM, Vol. 19, No. 8, August 1976, pp. 454-460.

72. Cline, Alan K. "Scalar- and Planar-Valued Curve Fitting Using Splines Under Tension", Communications of the ACM, Vol. 17, No. 4, April 1974, pp. 218-220. Also in Atmos. Tech., No. 3, 1973, pp. 60-65.

73. Cohen, Elaine; Lyche, Tom; and Riesenfeld, Richard F. "Discrete B-splines and Subdivision Techniques in Computer-Aided Geometric Design and Computer Graphics", Computer Graphics and Image Processing, Vol. 14, No. 2, October 1980, pp. 87-111. Also Tech. Report No. UUCS-79-117, Department of Computer Science, University of Utah, October 1979.

74. Cohen, Elaine and Riesenfeld, Richard F. "General Matrix Representations for Bezier and B-spline Curves", Computers in Industry. Special Steven A. Coons Memorial Issue, to appear.

75. Coons, Steven A. "Surfaces for Computer-Aided Design of Space Forms", Tech. Report No. MAC-TR-41, Project MAC, M.I.T., Cambridge, Massachusetts, June 1967. Available as AD-663 504 from NTIS, Springfield, Virginia.

76. Coons, Steven A. "Surface Patches and B-spline Curves". In Computer Aided Geometric Design, edited by Barnhill, Robert E. and Riesenfeld, Richard F., Academic Press, New York, 1974, pp. 1-16.

77. Coons, Steven A. "Modification of the Shape of Piecewise Curves", Computer-Aided Design, Vol. 9, No. 3, July 1977, pp. 178-180.

78. Cox, Morris G. "The Numerical Evaluation of B-splines", Report No. NPL-DNACS-4, Division of Numerical Analysis and Computing, National Physical Laboratory, Teddington, Middlesex, England, August 1971. Also in J. Inst. Maths. Applics., Vol. 10, 1972, pp. 134-149.

79. Cox, Morris G. "An Algorithm for Approximating Convex Functions by Means of First-Degree Splines", The Computer Journal, Vol. 14, No. 3, 1971, pp. 272-275.

80. Cox, Morris G. "Cubic-Spline Fitting with Convexity and Concavity Constraints", Report No. NPL-DNACS-23, National Physical Laboratory, Teddington, Middlesex, England, 1973.

81. Cox, Morris G. "An Algorithm for Spline Interpolation", J. Inst. Maths. Applics, 1975, pp. 95-108.

82. Cox, Morris G. "Direct versus Iterative Methods for Multivariate Spline Fitting Problems", Report No. NPL-DNACS-37/80, Division of Numerical Analysis and Computing, National Physical Laboratory, Teddington, Middlesex, England, October 1980.

83. Craven, P. and Wahba, G. "Smoothing Noisy Data with Spline Functions; Estimating the Correct Degree of Smoothing by the Method of Generalized Cross-Validation", TR No. 445, Department of Statistics, University of Wisconsin, Madison, Wisconsin, October 1977.

84. Creutz, Guenter and Schubert, Christian. "An Interactive Line Creation Method Using B-splines", Computers and Graphics, Vol. 5, No. 2-4, 1980, pp. 69-78.

85. Curry, H. B. and Schoenberg, I. J. "On Spline Distributions and their Limits: The Polya Distribution Functions, Abstract 380t", Bulletin of the American Mathematical Society, Vol. 53, 1947, pp. 1114.

86. Curry, H. B. and Schoenberg, I. J. "On Polya Frequency Functions IV: The Fundamental Spline Functions and their Limits", Journal d'Analyse Mathematique, Vol. 17, 1966, pp. 71-107.

87. Dahmen, Wolfgang. "On Multivariate B-splines", SIAM Journal on Numerical Analysis, Vol. 17, No. 2, April 1980, pp. 179-191.

88. Davis, Philip J. Interpolation and Approximation, Ginn-Blaisdell, New York, 1963. Republished by Dover Publ., 1975.

89. Deimel, L. E. Jr.; Doss, C. L. II; Fornaro, R. J.; McAllister, D. F.; and Roulier, J. A. "Application of Shape-Preserving Spline Interpolation to Interactive Editing of Photogrammetric Data". In Proceedings of SIGGRAPH '78, ACM, August 1978, pp. 93-99.

90. Denman, H. H. "Smooth Cubic Spline Interpolation Functions", Industrial Mathematics, J. Ind. Math. Soc., Vol. 21, No. 2, 1971, pp. 55-75.

91. DeVore, R. and Richards, F. "Saturation and Inverse Theorems for Spline Approximation". In Spline Functions and Approximation Theory, edited by Meir, A. and Sharma, A., Birkhauser Verlag, Basel, 1973, pp. 73-82. ISNM, Vol. 21.

92. Dimsdale, B. "Bicubic Patch Bounds", Comp. & Maths. with Appls., Vol. 3, No. 2, 1977, pp. 95-104.

93. Dimsdale, B. "Convex Cubic Splines", IBM J. Res. and Develop., Vol. 22, No. 2, March 1978, pp. 168-178.

94. Dimsdale, B. and Burkley, R. M. "Bicubic Patch Surfaces for High-Speed Numerical Control Processing", IBM J. Res. and Develop., 1976, pp. 358-367.

95. Dodson, D. S. Optimal Order Approximation by Polynomial Spline Functions, Ph.D. Thesis, Purdue University, West Lafayette, Indiana, August 1972.

96. Done, G. T. S. "Interpolation of Mode Shapes: A Matrix Scheme Using Two-Way Spline Curves", The Aeronautical Quarterly, November 1965, pp. 333-349.

97. Doo, D. W. H. "A Subdivision Algorithm for Smoothing Down Irregular Shaped Polyhedrons". In Proceedings of the International Conference Interactive Techniques in Computer Aided Design, IEEE Computer Society, Bologna, 21 - 23 September 1978, pp. 157-165.

98. Doo, D. W. H. and Sabin, M. A. "Behaviour of Recursive Division Surfaces Near Extraordinary Points", Computer-Aided Design, Vol. 10, No. 6, November 1978, pp. 356-360.

99. Dube, R. Peter. Local Schemes for Computer Aided Geometric Design, Ph.D. Thesis, University of Utah, Salt Lake City, Utah, June 1975.

100. Dube, R. Peter. "Tension in a Bicubic Surface Patch", Computer Graphics and Image Processing, Vol. 5, December 1976, pp. 496-502.

101. Dube, R. Peter. "Univariate Blending Functions and Alternatives", Computer Graphics and Image Processing, Vol. 6, 1977, pp. 394-408.

102. Dube, R. Peter. "Preliminary Specification of Spline Curves", IEEE Transactions on Computers, Vol. C-28, No. 4, April 1979, pp. 286-290.

103. Dube, R. Peter; Herron, Gary J.; Little, Frank F.; and Riesenfeld, Richard F. "SURFED -- An Interactive Editor for Free-Form Surfaces", Computer-Aided Design, Vol. 10, No. 2, March 1978, pp. 111-115.

104. Farin, Gerald. Subsplines Ueber Dreiecken, Ph.D. Thesis, Technische Universitaet Braunschweig, Braunschweig, West Germany, 1979.

105. Faux, Ivor D. and Pratt, Michael J. Computational Geometry for Design and Manufacture, Ellis Horwood Ltd., 1979.

106. Forrest, A. Robin. Curves and Surfaces for Computer-Aided Design, Ph.D. Thesis, Cambridge University, Cambridge, England, July 1968.

107. Forrest, A. Robin. "The Computation of Bicubic Twist Terms", CAD Group Document No. 20, Computer Aided Design Group, Cambridge University, January 1969.

108. Forrest, A. Robin. "Interactive Interpolation and Approximation by Bezier Polynomials", The Computer Journal, Vol. 15, No. 1, January 1972, pp. 71-79. Also CAD Group Document No. 45, UML, Cambridge University, October 1970.

109. Forrest, A. Robin. "Mathematical Principles for Curve and Surface Representation". In Proceedings of Curved Surfaces in Engineering, Churchill College, edited by Brown, I. J., IPC Science and Technology Press, Ltd., Cambridge, England, 15-17th March 1972, pp. 5-13.

110. Fritsch, F. N. and Carlson, R. E. "Monotone Piecewise Cubic Interpolation", SIAM Journal on Numerical Analysis, Vol. 17, No. 2, April 1980, pp. 238-246.

111. Godwin, A. N. "Family of Cubic Splines with One Degree of Freedom", Computer-Aided Design, Vol. 11, No. 1, January 1979, pp. 13-18.

112. Gordon, William J. "Distributive Lattices and the Approximation of Multi-Variate Functions". In Proceedings of the Symposium on Approximation with Special Emphasis on Splines, edited by Schoenberg, Isaac J., University of Wisconsin Press, Madison, Wisconsin, 1969.

113. Gordon, William J. "Spline-Blended Surface Interpolation through Curve Networks", Journal of Mathematics and Mechanics, Vol. 18, No. 10, 1969, pp. 931-952. Also Research Publication GMR-921, General Motors Research Laboratories, September 1969.

114. Gordon, William J. and Riesenfeld, Richard F. "B-spline Curves and Surfaces". In Computer Aided Geometric Design, edited by Barnhill, Robert E. and Riesenfeld, Richard F., Academic Press, New York, 1974, pp. 95-126.

115. Gordon, William J. and Riesenfeld, Richard F. "Bernstein-Bezier Methods for the Computer-Aided Design of Free Form Curves and Surfaces", Journal of the ACM, Vol. 21, No. 2, April 1974, pp. 293-310. Also Research Publication GMR-1176, General Motors Research Laboratories, March 1972.

116. Greville, T. N. E. "Interpolation by Generalized Spline Functions", University of Wisconsin, Madison, 1964.

117. Greville, T. N. E. "On the Normalisation of the B-splines and the Location of the Nodes for the case of Unequally Spaced Knots". In Inequalities, edited by Shisha, Oved, Academic Press, New York, 1967. Supplement to the paper "On Spline Functions" by Isaac J. Schoenberg.

118. Greville, T. N. E. (editor). Theory and Applications of Spline Functions, Academic Press, New York, 1969.

119. Greville, T. N. E. "Introduction to Spline Functions". In Theory and Applications of Spline Functions, edited by Greville, T. N. E., Academic Press, New York, 1969.

120. Hall, C. A. "On Error Bounds for Spline Interpolation", Journal of Approximation Theory, Vol. 1, No. 2, September 1968, pp. 209-218.

121. Hall, C. A. "Bicubic Interpolation Over Triangles", J. Math and Mech., Vol. 19, 1969, pp. 1-11.

122. Hall, C. A. and Meyer, W. W. "Optimal Error Bounds for Cubic Spline Interpolation", Journal of Approximation Theory, Vol. 16, 1976, pp. 105-122.

123. Hanson, Richard J. "Constrained Least Squares Curve Fitting to Discrete Data Using B-splines: A User's Guide", Numerical Mathematics Divsion, Sandia Laboratories, Albuquerque, New Mexico, February 1979.

124. Harris, M. "SPLIFIT: An Interactive Graphics Curve Fitting Computer Program Utilizing Spline Functions", Report No. PA-TR-4677, Picatinny Arsenal, Dover, N.J., January 1974.

125. Hartley, P. J. and Judd, C. J. "Parametrisation of Bezier-Type B-spline Curves and Surfaces", Computer-Aided Design, Vol. 10, No. 2, March 1978, pp. 130-134.

126. Hartley, P. J. and Judd, C. J. "Parametrization and Shape of B-spline Curves for CAD", Computer-Aided Design, Vol. 12, No. 5, September 1980, pp. 235-238.

127. Hartley, P. J. and Judd, C. J. "Curve and Surface Representations for Bezier B-spline Systmes". In Proceedings of the CAD80 Conference, IPC Science and Technology Press, Ltd., Guildford, England, March 1980, pp. 226-236.

128. Hayes, Jeffrey G. "New Shapes from Bicubic Splines", Report No. V.58, Division of Numerical Analysis and Computing, National Physical Laboratory, Teddington, Middlesex, England, September 1974.

129. Hayes, J. G. and Halliday, J. "The Least-Squares Fitting of Cubic Spline Surfaces to General Data Sets", J. Inst. Maths. Applics., Vol. 14, 1974, pp. 89-103.

130. Hazony, Y. "Algorithms for Parallel Processing Curve and Surface Definition with Q-splines", Computers and Graphics, Vol. 4, No. 3-4, 1979, pp. 165-76.

131. Hedstrom, G. W. and Varga, R. S. "Application of Besov Spaces to Spline Approximation", Journal of Approximation Theory, Vol. 4, 1971, pp. 295-327.

132. Herron, Gary J. Triangular and Multisided Patch Schemes, Ph.D. Thesis, University of Utah, Salt Lake City, Utah, 1979.

133. Higdon, Archie; Ohlsen, Edward H.; Stiles, William B.; and Weese, John A. Mechanics of Materials, John Wiley and Sons, Inc., New York, 1967.

134. Hulme, B. L. Piecewise Bicubic Methods for Plate Bending Problems, Ph.D. Thesis, Harvard University, Cambridge, Massachusetts, April 1969.

135. Hyman, James M. "Accurate Monotonicity Preserving Cubic Interpolation", Tech. Report No. LA-UR 81-841, Los Alamos Scientific Laboratory, 1981.

136. Inselberg, A. "Cubic Splines with Infinite Derivatives at Some Knots", IBM J. Res. and Develop., September 1976, pp. 430-436.

137. Jerome, J. W. and Schumaker, L. L. "On L-g Splines", Journal of Approximation Theory, Vol. 2, 1969, pp. 29-49.

138. Jerome, J. W. and Varga, R. S. "Generalizations of Spline Functions and Applications to Nonlinear Boundary Value and Eigenvalue Problems". In Theory and Applications of Spline Functions, edited by Greville, T. N. E., Academic Press, New York, 1969, pp. 103-155.

139. Jiachang, S. "A Class of Matrix Methods for Surface Representation". In Proceedings of the CAD80 Conference, IPC Science and Technology Press, Ltd., Guildford, England, March 1980, pp. 251-254.

140. Jupp, David L. B. "Approximation to Data by Splines with Free Knots", SIAM Journal on Numerical Analysis, Vol. 15, No. 2, April 1978, pp. 328-343.

141. Karlin, S. and Ziegler, Z. "Tschebysheffian Spline Functions", SIAM Journal on Numerical Analysis, No. 3, 1966, pp. 514-543.

142. Karlin, Samuel; Micchelli, Charles A.; Pinkus, Allan; and Schoenberg, Isaac J. Studies in Spline Functions and Approximation Theory, Academic Press, New York, 1976.

143. Knapp, Lewis C. A Design Scheme Using Coons Surfaces with Nonuniform B-spline Curves, Ph.D. Thesis, Syracuse University, Syracuse, N.Y., September 1979.

144. Kojima, T. "Some Geometric Relations between the Cubic B-spline Curve and its Associated Characteristic Polygon", Memo No. CGP78/2, School of Computing Studies and Accountancy, Computational Geometry Project, University of East Anglia, June 1978.

145. Kojima, T. "Representation of a Curve Using a Spline Function", Trans. Soc. Instrum. and Control Eng. (Japan), Vol. 14, No. 2, April 1978, pp. 144-148.

146. Lane, Jeffrey M. and Riesenfeld, Richard F. "A Theoretical Development for the Computer Generation of Piecewise Polynomial Surfaces", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-2, No. 1, January 1980, pp. 35-46.

147. Lane, Jeffrey M. and Riesenfeld, Richard F. "A Geometric Proof of the Variation Diminishing Property of B-spline Approximation", Journal of Approximation Theory. To appear.

148. Lucas, T. R. "A Theory of Generalized Splines", Numerische Mathematik, Vol. 15, 1970, pp. 359-370.

149. Lyche, Tom. Discrete Polynomial Spline Approximation Methods, Ph.D. Thesis, University of Texas at Austin, 1975. For a summary, see Spline Functions, Karlsruhe 1975, edited by Bohmer, K.; Meinandus, G.; and Schimpp, W., Lecture Notes in Mathematics No. 501, Springer-Verlag, pp. 144-176, 1976.

150. Lyche, Tom. "Discrete Spline Interpolation", BIT, Vol. 16, 1976, pp. 281-290.

151. Lyche, Tom and Schumaker, Larry L. "Computation of Smoothing and Interpolating Natural Splines via Local Bases", Center of Numerical Analysis, University of Texas, April 1971.

152. Lyche, Tom and Schumaker, Larry L. "Local Spline Approximation Methods", Journal of Approximation Theory, No. 15, 1975, pp. 294-325.

153. Malcolm, M. A. "On the Computation of Nonlinear Spline Functions", SIAM Journal on Numerical Analysis, Vol. 14, No. 2, 1977, pp. 254-282.

154. Mangasarian, Olvi L. and Schumaker, Larry L. "Discrete Splines via Mathematical Programming", SIAM Journal on Control, Vol. 9, 1971, pp. 174-183.

155. Mangasarian, Olvi L. and Schumaker, Larry L. "Best Summation Formula and Discrete Splines", SIAM Journal on Numerical Analysis, Vol. 10, 1973, pp. 448-459.

156. Manning, J. R. "Continuity Conditions for Spline Curves", The Computer Journal, Vol. 17, No. 2, May 1974, pp. 181-186.

157. Marlow, S. and Powell, M. J. D. "A FORTRAN Subroutine for Plotting a Cubic Spline Function", Reprot No. AERE-R-7470, UKAEA, Harwell, Berks., England, July 1973.

158. Marsden, Martin J. "An Identity for Spline Functions with Approximations to Variation-Diminishing Spline Approximation", Journal of Approximation Theory, Vol. 3, 1970, pp. 7-49.

159. Marsden, Martin J. "Quadratic Spline Interpolation", Bulletin of the American Mathematical Society, No. 80, 1974, pp. 903-906.

160. Marsden, Martin J. "On Uniform Spline Approximation", Journal of Approximation Theory, No. 6, 1977, pp. 254-282.

161. McAllister, David F.; Passow, Eli; and Roulier, John A. "Algorithms for Computing Shape Preserving Spline Interpolation to Data", Mathematics of Computation, Vol. 31, No. 139, July 1977, pp. 717-725.

162. McAllister, David F. and Roulier, John A. "Interpolation by Convex Quadratic Splines", Mathematics of Computation, Vol. 32, No. 144, October 1978, pp. 1154-1162.

163. McLeod, R. J. Y. and Mitchell, A. R. "A Piecewise Parabolic C^1 Approximation Technique for Curved Boundaries", Comput. and Math. with Appl., Vol. 5, No. 4, 1979, pp. 277-284.

164. Mehlum, Even. "Nonlinear Splines". In Computer Aided Geometric Design, edited by Barnhill, Robert E. and Riesenfeld, Richard F., Academic Press, New York, 1974, pp. 173-205.

165. Meinardus, G. "Algebraische Formulierung von Spline-Interpolationen". In Moderne Methoden der Numerischen Mathematik, International Series in Numerical Mathematics, Birkhauser Verlag, Basel and Stuttgart, 1976, pp. 125-138.

166. Meir, A. and Sharma, A. "Convergence of a Class of Interpolatory Splines", Journal of Approximation Theory, Vol. 1, 1968, pp. 243-250.

167. Meir, A. and Sharma, A. "On Uniform Approximation by Cubic Splines", Journal of Approximation Theory, Vol. 2, 1969, pp. 270-274.

168. Micchelli, Charles A. "On a Numerically Efficient Method for Computing Multivariate B-splines". In Multivariate Aproximation Theory, ISNM 30, edited by Schempp, Walter and Zeller, Karl, Birkhaeuser Verlag, Basel, 1979, pp. 211-248. Also Research Report No. RC 7716 (#33432), IBM T. J. Watson Research Center, Yorktown Heights, N.Y.

169. Micchelli, Charles A. "A Constructive Approach to Kergin Interpolation in R^k: Multivariate B-splines and Lagrange Interpolation", Rocky Mountain Journal of Mathematics. To appear. Also Math. Res. Center Report No. 1895, University of Wisconsin, Madison.

170. Munchmeyer, Fredrick C. "On the Interactive Design of Smooth Patched Surfaces". In Proceedings of the International Conference Interactive Techniques in Computer Aided Design, IEEE Computer Society, Bologna, 21 - 23 September 1978, pp. 393-403.

171. Munchmeyer, Fredrick C.; Schubert, Christian; and Nowacki, Horst. "Interactive Design of Fair Hull Surfaces Using B-splines", Computers in Industry, Vol. 1, No. 2, December 1979, pp. 77-86.

172. Munteanu, M. J. and Schumaker, L. L. "On a Method of Carasso and Laurent for Constructing Interpolating Splines", Mathematics of Computation, No. 27, 1973, pp. 317-325.

173. Myers, Diane-Claire and Roulier, John A. "Markov-Type Inequalities and the Degree of Convex Spline Interpolation", Journal of Approximation Theory, Vol. 28, No. 3, March 1980, pp. 267-272.

174. Napolitano, L. G. and Losito,V. "The Closed Spline Functions", Comput. Methods Appl. Mech. and Eng., Vol. 13, No. 3, March 1978, pp. 335-350.

175. Newman, William M. and Sproull, Robert F. Principles of Interactive Computer Graphics, McGraw-Hill, 1979. Second edition.

176. Nielson, Gregory M. Surface Approximation and Data Smoothing Using Generalized Spline Functions, Ph.D. Thesis, University of Utah, Salt Lake City, Utah, 1970.

177. Nielson, Gregory M. "Methods for Constructing Combined Smoothing and Interpolating Spline Functions", Tech. Memo No. 9, Department of Mathematics, Arizona State University, September 1973.

178. Nielson, Gregory M. "Some Piecewise Polynomial Alternatives to Splines under Tension". In Computer Aided Geometric Design, edited by Barnhill, Robert E. and Riesenfeld, Richard F., Academic Press, New York, 1974, pp. 209-235.

179. Nielson, Gregory M. "Computation of NU-splines", Tech. Report No. 044-433-11, Department of Mathematics, Arizona State University, June 1974.

180. Nielson, Gregory M. "Multivariate Smoothing and Interpolating Splines", SIAM Journal on Numerical Analysis, Vol. 11, No. 2, April 1974, pp. 435.

181. Nord, S. "Approximation Properties of the Spline Fit", BIT, Vol. 7, 1967, pp. 132-144.

182. Nutbourne, A. W. "A Cubic Spline Package. Part 2 -- The Mathematics.", Computer-Aided Design, Vol. 5, No. 1, 1973, pp. 7-13.

183. Overhauser, Albert W. "Analytic Definition of Curves and Surfaces by Parabolic Blending", Tech. Report No. SL68-40, Scientific Research Staff Publications, Ford Motor Company, May 1968.

184. Pal, T. K. "Intrinsic Spline Curve with Local Control", Computer-Aided Design, Vol. 10, No. 1, January 1978, pp. 19-29.

185. Passow, Eli. "Piecewise Monotone Spline Interpolation", Journal of Approximation Theory, Vol. 12, No. 3, November 1974, pp. 240-241.

186. Passow, Eli and Roulier, John A. "Monotone and Convex Spline Interpolation", SIAM Journal on Numerical Analysis, Vol. 14, No. 5, October 1977, pp. 904-909.

187. Payne, P. J. "A Contouring Program for Joined Surface Patches", CAD Group Document No. 58, Computer Aided Design Group, Cambridge University, June 1971.

188. Pilcher, David T. Smooth Approximation of Parametric Curves and Surfaces, Ph.D. Thesis, University of Utah, Salt Lake City, Utah, August 1973.

189. PorterGoff, R. F. D. "The Representation of Surfaces by Coons' Patch Technique", Tech. Report No. 69-3, University of Leicester, England, February 1969.

190. Prenter, Paddy M. "Piecewise L-splines", Numerische Mathematik, Vol. 18, No. 3, December 1971, pp. 243-253.

191. Prenter, Paddy M. Splines and Variational Methods, John Wiley and Sons, Inc., New York, 1975.

192. Pruess, Steven. "Properties of Splines in Tension", Journal of Approximation Theory, Vol. 17, No. 1, May 1976, pp. 86-96.

193. Pruess, Steven. "An Algorithm for Computing Smoothing Splines in Tension", Computing, Vol. 19, 1978, pp. 365-373.

194. Pruess, Steven. "Alternatives to the Exponential Spline in Tension", Mathematics of Computation, Vol. 33, No. 148, 1979, pp. 1273-1281.

195. Reinsch, C. H. "Smoothing by Spline Functions", Numerische Mathematik, Vol. 10, 1967, pp. 177-183.

196. Reinsch, C. H. "Smoothing by Spline Functions II", Numerische Mathematik, Vol. 16, 1971, pp. 451-454.

197. Rice, John R. "Characterization of Chebyshev Approximation by Splines", SIAM Journal on Numerical Analysis, No. 4, 1967, pp. 557-565.

198. Rice, John R. "On the Degree of Convergence of Nonlinear Spline Approximation". In Approximation with Special Emphasis on Spline Functions, edited by Schoenberg, Isaac J., Academic Press, New York, 1969, pp. 349-365.

199. Riesenfeld, Richard F. Applications of B-spline Approximation to Geometric Problems of Computer-Aided Design, Ph.D. Thesis, Syracuse University, Syracuse, N.Y., May 1973. Available as Tech. Report No. UTEC-CSc-73-126, Department of Computer Science, University of Utah.

200. Riesenfeld, Richard F. "Nonuniform B-spline Curves". In Proceedings of the 2nd USA-Japan Computer Conference, AFIPS, Montvale, N.J., 26-28 August 1975, pp. 551-555.

201. Riesenfeld, Richard F. "Aspects of Modelling in Computer Aided Geometric Design". In Proceedings of the National Computer Conference, AFIPS, May 1975, pp. 597-602.

202. Riesenfeld, Richard F. "On Chaikin's Algorithm", Computer Graphics and Image Processing, Vol. 4, No. 3, September 1975, pp. 304-310.

203. Riesenfeld, Richard F.; Cohen, Elaine; Fish, Russell D.; Thomas, Spencer W.; Cobb, Elizabeth S.; Barsky, Brian A.; Schweitzer, Dino L.; and Lane, Jeffrey M. "Using the Oslo Algorithm as a Basis for CAD/CAM Geometric Modelling". In Proceedings of the Second Annual NCGA National Conference, National Computer Graphics Association, Inc., Baltimore, 14-18 June 1981. To appear.

204. Rogers, David F. and Adams, J. Alan. Mathematical Elements for Computer Graphics, McGraw-Hill, New York, 1976.

205. Rogers, David F. and Satterfield, Steven G. "B-spline Surfaces for Ship Hull Design". In Proceedings of SIGGRAPH '80, ACM, July 1980, pp. 211-217.

206. van Rooij, P. L. J. and Schurer, F. "A Bibliography on Spline Functions". In Tagung ueber Spline-Funktionen, edited by Boehmer, K; Meinardus, G.; and Schempp, W., Bibliographisches Institut, Mannheim, 1974, pp. 315-415.

207. Sabin, Malcolm A. "Spline Surfaces", Report No. VTO/MS/156, Dynamics and Mathematical Services Dept., British Aircraft Corporation, Weybridge, June 1968.

208. Sabin, Malcolm A. "A 16-Point Bicubic Formulation Suitable for Multipatch Surfaces", Report No. VTO/MS/155, Dynamics and Mathematical Services Dept., British Aircraft Corporation, Weybridge, March 1969.

209. Sabin, Malcolm A. "Parametric Splines in Tension", No. VTO/MS/160, British Aircraft Corporation, Weybridge, Surrey, England, 23 July 1970.

210. Sabin, Malcolm A. The Use of Piecewise Forms for the Numerical Representation of Shape, Ph.D. Thesis, Budapest, 1976.

211. Sablonniere, P. "Spline and Bezier Polygons Associated with a Polynomial Spline Curve", Computer-Aided Design, Vol. 10, No. 4, July 1978, pp. 257-261.

212. Sard, A. and Weintraub, S. A Book of Splines, John Wiley and Sons, Inc., New York, 1971.

213. Schaback, R. "Calculation of Best Approximations by Rational Splines". In Approximation Theory, edited by Lorentz, G. G.; Chui, C. K.; and Schumaker, L. L., Academic Press, New York, 1976, pp. 533-539.

214. Schechter, A. "Synthesis of 2D Curves by Blending Piecewise Linear Curvature Profiles", Computer-Aided Design, Vol. 10, No. 1, January 1978, pp. 8-18.

215. Schoenberg, Isaac J. "Contributions to the Problem of Approximation of Equidistant Data by Analytic Functions", Quarterly Applied Math., Vol. 4, No. 1, 1946, pp. 45-99 and 112-141.

216. Schoenberg, Isaac J. "On Variation Diminishing Approximation Methods". In On Numerical Approximation, edited by Langer, R. E., University of Wisconsin Press, Madison, Wisconsin, 1959, pp. 249-274.

217. Schoenberg, Isaac J. "Spline Functions and the Problem of Graduation". In Proceedings of the National Academy of Sciences, 1964, pp. 947-950.

218. Schoenberg, Isaac J. "On Polya Frequency Functions IV: The Fundamental Splines and Their Limits", Math. Res. Center Report No. 567, University of Wisconsin, Madison, 1965.

219. Schoenberg, Isaac J. "On Hermite-Birkhoff Interpolation", Math. Res. Center Technical Report No. 659, University of Wisconsin, Madison, May 1966.

220. Schoenberg, Isaac J. "On Spline Functions". In Inequalities, edited by Shisha, Oved, Academic Press, New York, 1967, pp. 255-291. With supplement by T. N. E. Greville. Also Math. Res. Center Report 625, University of Wisconsin, Madison, May 1966.

221. Schoenberg, Isaac J. "Spline Functions and the Higher Derivatives", Math. Res. Center Report No. 761, University of Wisconsin, Madison, July 1967.

222. Schoenberg, Isaac J. Cardinal Spline Interpolation, SIAM, Philadelphia, Penn., 1973.

223. Schoenberg, I. J. and Whitney, A. "On Polya Frequency Functions III: The Positivity of Translation Determinants with an Application to the Interpolation Problem by Spline Curves", Transactions of the American Mathematical Society, Vol. 74, 1953, pp. 246-259.

224. Schultz, Martin H. "Error Bounds for Polynomial Spline Interpolation", Mathematics of Computation, Vol. 24, 1970, pp. 507-515.

225. Schultz, Martin H. Spline Analysis, Prentice-Hall, Inc., Series in Automatic Computation, Englewood Cliffs, N.J., 1973.

226. Schultz, Martin H. "Computing Piecewise Hermite Interpolates Without Derivatives", Yale University.

227. Schultz, Martin H. "Computing Spline Interpolates Without Derivatives", Yale University.

228. Schultz, Martin H. and Varga, Richard S. "L-splines", Numerische Mathematik, Vol. 10, 1967, pp. 345-369.

229. Schultz, Martin H. and Varga, Richard S. "L-splines", Numerische Mathematik, Vol. 10, 1967, pp. 345-369.

230. Schumaker, Larry L. "Uniform Approximation by Tchebycheffian Spline Functions", J. Math. Mech., No. 18, 1968, pp. 369-377.

231. Schumaker, Larry L. "Uniform Approximation by Chebyshev Spline Functions, II: Free Knots", SIAM Journal on Numerical Analysis, No. 5, 1968, pp. 647-656.

232. Schumaker, Larry L. "Some Algorithms for the Computation of Interpolating and Approximating Spline Functions". In Theory and Applications of Spline Functions, edited by Greville, T. N. E., Academic Press, New York, 1969.

233. Schumaker, Larry L. "Constructive Aspects of Discrete Polynomial Spline Functions". In Approximation Theory, edited by Lorentz, G. G., Academic Press, New York, 1973, pp. 469-476.

234. Schumaker, Larry L. "Work on Splines". University of Texas.

235. Schurer, F. "A Note on Interpolating Periodic Quintic Splines with Equally Spaced Nodes", Journal of Approximation Theory, Vol. 1, 1968, pp. 493-500.

236. Schweikert, Daniel G. "An Interpolation Curve Using a Spline in Tension", Journal of Mathematics and Physics, Vol. 45, 1966, pp. 312-317.

237. Sharma, A. and Meir, A. "Degree of Approximation of Spline Interpolation", J. Math. Mech, No. 15, 1966, pp. 759-768.

238. Soanes, R. W. Jr. "VP-splines, an Extension of Twice Differentiable Interpolation". In Proceedings of the 1976 Army Numerical Analysis and Computer Conference (ARO Report 76-3), U.S. Army Research Office, 1976, pp. 141-152.

239. Spaeth, H. "Exponential Spline Interpolation", Computing, Vol. 4, 1969, pp. 225-233.

240. Spaeth, H. Spline-Algorithmen zur Konstruktion glatter Kurven und Flaechen, R. Oldenburg Verlag, Munich, West Germany, 1973. English translation by Hoskins, W. D. and Sager, H. W. Spline Algorithms for Curves and Surfaces, Utilitas Mathematica Publ., Inc., Winnipeg, Manitoba, Canada, 1974.

241. Stechkin, S. B. and Subbotin, Yu N. Splines in Numerical Mathematics, Izd. 'Nauka', Moscow, 1976.

242. Strasser, Wolfgang. Schnelle Kurven- und Flaechendarstellung auf grafischen Sichtgeraeten, Ph.D. Thesis, Technische Universitaet Berlin, Berlin, September 1974.

243. Subbotin, Yu N. "Piecewise Polynomial Interpolation", Math. Zametki, Vol. 1, 1967.

244. Sun, Jia-Chang. "The Spline Interpolation for Space Curves in Local Coordinates", Acta Math. App. Sin. (China), Vol. 2, No. 4, November 1979, pp. 340-343.

245. Swartz, Blair K. "$O(h^{2n+2-1})$ Bounds on some Spline Interpolation Errors", Bulletin of the American Mathematical Society, Vol. 74, 1968, pp. 1072-1078.

246. Swartz, Blair K. "$O(h^{k-j}(D^k f,h))$ Bounds on Some Spline Interpolation Errors", Tech. Report No. LA-4477, Los Alamos Scientific Laboratory, 1970.

247. Swartz, Blair K. and Varga, Richard S. "Error Bounds for Spline and L-spline Interpolation", Journal of Approximation Theory, Vol. 6, No. 1, July 1972, pp. 6-49.

248. Swartz, K. D. and Varga, R. S. "Bounds for Some Spline Interpolation Errors". To appear.

249. Thomas, D. H. "Pseudospline Interpolation for Space Curves", Mathematics of Computation, Vol. 30, No. 133, 1976, pp. 58-67.

250. Tilove, Robert B. "B-splines for the Uninitiated", Memo No. CGP78/9, School of Computing Studies and Accountancy, Computational Geometry Project, University of East Anglia, August 1978.

251. Varah, J. M. "On the Condition Number of Local Bases for Piecewise Cubic Polynomials", Mathematics of Computation, Vol. 31, No. 137, 1977, pp. 37-44.

252. Varga, Richard S. "Error Bounds for Spline Interpolation". In Approximations with Special Emphasis on Spline Functions, edited by Schoenberg, Isaac J., Academic Press, New York, 1969, pp. 367-388.

253. Veron, M.; Ris, G.; and Musse, J.-P. "Continuity of Biparametric Surface Patches", Computer-Aided Design, Vol. 8, No. 4, 1976, pp. 267-273.

254. Versprille, Kenneth J. Computer-Aided Design Applications of the Rational B-spline Approximation Form, Ph.D. Thesis, Syracuse University, Syracuse, N.Y., February 1975.

255. Walsh, J. H.; Ahlberg, J. H.; and Nilson, E. H. "Best Approximation Properties of the Spline Fit", J. Math. Mech., Vol. 11, 1962, pp. 225-234.

256. Weinert, H. L.; Byrd, R. H.; and Sidhu, G. S. "A Stochastic Framework for Recursive Computation of Spline Functions. II. Smoothing Splines", J. Optimiz. Theory and Appl., Vol. 30, No. 2, February 1980, pp. 255-268.

257. Weinert, H. L. and Sidhu, G. S. "A Stochastic Framework for Recursive Computation of Spline Functions. I. Interpolating Splines", IEEE Transactions on Information Theory, Vol. IT-24, No. 1, June 1978, pp. 45-50.

258. Wielinga, R. F. "Constrained Interpolation using Bezier Curves as a New Tool in Computer-Aided Geometric Design". In Computer Aided Geometric Design, edited by Barnhill, Robert E. and Riesenfeld, Richard F., Academic Press, New York, 1974.

259. Wu, Sheng-Chuan and Abel, John F. "Representation and Discretization of Arbitrary Surfaces for Finite Element Shell Analysis", International Journal for Numerical Methods in Engineering, Vol. 14, No. 6, 1979, pp. 813-836.

260. Wu, Sheng-Chuan; Abel, John F.; and Greenberg, Donald P. "An Interactive Computer Graphics Approach to Surface Representation", Communications of the ACM, Vol. 20, No. 10, October 1977, pp. 703-712.

261. Xiong, Zhen-Xiang. "The Spline Functions of (2n+1)th Degree with the Coefficients Expressed by Even Order Derivatives". In Proceedings of the CAD80 Conference, IPC Science and Technology Press, Ltd., Guildford, England, March 1980, pp. 237-242.

262. Yamaguchi, Fujio. "A New Curve Fitting Method Using a CRT Computer Display", Computer Graphics and Image Processing, Vol. 7, No. 3, June 1978, pp. 425-437.

263. York, Bryant W. "A Primer on Polynomial Interpolation and Splines", COINS Technical Report No. 79-5, Computer and Information Science, University of Massachusetts at Amherst, 1979.

264. Young, Sam W. "Piecewise Monotone Spline Interpolation", Bulletin of the American Mathematical Society, Vol. 73, No. 5, September 1967, pp. 642-643.