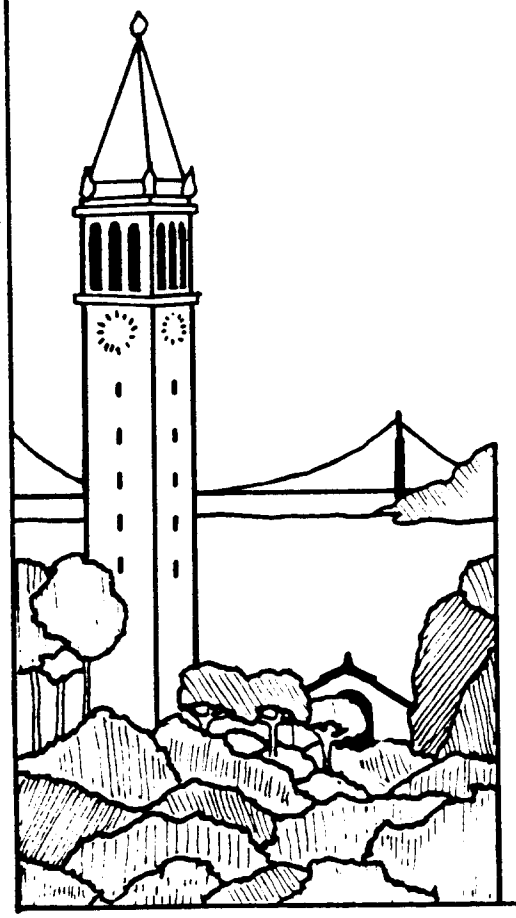
Arbitrary Subdivision of Bézier Curves

Brian A. Barsky



Report No. UCB/CSD 86/265

November 1985

Computer Science Division (EECS)
University of California
Berkeley, California 94720

Arbitrary Subdivision of Bézier Curves

Brian A. Barsky

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

ABSTRACT

Subdivision is a powerful technique that has many useful applications. The fundamental concept is the splitting of a curve or surface into smaller pieces whose union is identical to the original curve or surface. Standard Bézier subdivision splits the curve at the midpoint of the curve, in parametric space.

This paper generalizes *midpoint subdivision* to *arbitrary subdivision*, enabling the subdivision to be performed at any parametric value, not solely at the midpoint. This allows for subdivision that would adapt to regions of varying curvature or correlate with the curve length in geometric space.

After explaining the original development of Bézier curves, the mathematical theory for arbitrary subdivision is developed, and finally an illustration of the subdivision process that shows the recursive procedure in a step-by-step manner is given.

This work was supported in part by an IBM Faculty Development Award.

1. Introduction

Subdivision is a powerful technique that has many useful applications. The fundamental concept is the splitting of a curve or surface into smaller pieces whose union is identical to the original curve or surface. This corresponds to determining new *control polygons*, that is, ordered sequences of *control vertices*, each of which defines one of the pieces. Bézier curves have several properties that make them particularly attractive for subdivision.[1,3,4,5]

A Bézier curve is confined to the *convex hull* of the control polygon that defines it. The convex hull of a control polygon can be found by imagining a rubber band stretched around its vertices. The area within the boundaries of the rubber band defines the convex hull. The new subdivided pieces are smaller than the original curve and the new control polygon lies much closer to the curve than did the original one. Consequently, if the subdivision process is performed recursively, the sequence of new control polygons generated at each step will converge to the curve itself. Using this approach, a piecewise linear approximation to the curve can be generated to within any desired tolerance without ever doing explicit curve evaluation.

A Bézier curve does not generally *interpolate*, or pass through, the control vertices that define it; however, it will indeed interpolate the vertex at each end. It follows immediately from this fact that the point on the curve at which the split occurs is exactly one of the new control vertices. In fact, this point is actually both the terminal vertex of one of the pieces and the initial vertex of the other. In this way, a single level of subdivision immediately yields a point on the curve itself. Since the determination of the new control vertices can be interpreted geometrically, this provides a *geometric construction* for a point on the curve.

In addition to the use of subdivision for display, the new vertices that are generated can be used for more precise control of shape in the design process. The vertices can provide a top-down design environment where a design can be refined by subdividing to engender additional vertices to be used for shape modification.[8]

Bézier (and uniform B-spline) subdivision was analyzed thoroughly by Lane and Riesenfeld in.[7] However, they handle only the case of *midpoint subdivision*, that is, where the subdivision point corresponds to the midpoint of the curve, in parametric space. For more flexibility, it is desirable to be able to subdivide at any point on the curve, called *arbitrary subdivision*. For example, if a curve is flat at one end but curved at the other, it would be beneficial to have the subdivision point be closer to the more curved part. Another situation where arbitrary subdivision would be useful occurs when the edges of the control polygon have disparate lengths; in this case, the value of the subdivision point in parametric space can be computed to take into account the length of the polygon edges in geometric space. This paper generalizes the work in[7] for arbitrary subdivision of Bézier curves.

2. Development of Bézier Curves

The original development by Pierre Bézier [5,6] constructed a twisted curve inside a rectangular parallelpiped in three-space. As shown in Figure 1, the curve starts at one corner (**A**) of the parallelpiped and finishes at the diagonally opposite corner (**B**). (The initial point is related to the coordinate system by the vector $\mathbf{A}_0$.) The beginning of the curve is tangent to one of the edges of the parallelpiped incident at **A** and the end of the curve is tangent to one of the edges incident at **B**. These edges are denoted $\mathbf{A}_1$ and $\mathbf{A}_3$, respectively. At **A**, the curve lies in the plane defined by $\mathbf{A}_1$ and $\mathbf{A}_2$, and at **B**, the curve lies in the plane defined by $\mathbf{A}_2$ and $\mathbf{A}_3$.

From this, a point on the curve is given by

$$\mathbf{Q}(u) = \mathbf{A} + \sum_{i=1}^{3} f_i(u)\mathbf{A}_i \qquad 0 \le u \le 1 \tag{1}$$

The parametric functions $f_i(u)$, $i=1,2,3$, can be determined from the relation between the curve and the parallelpiped given above.

The beginning of the curve, $\mathbf{Q}(0)$, is the point **A**. This can be satisfied by equation (1) if

$$f_i(0) = 0, \quad i = 1, 2, 3 \tag{2}$$

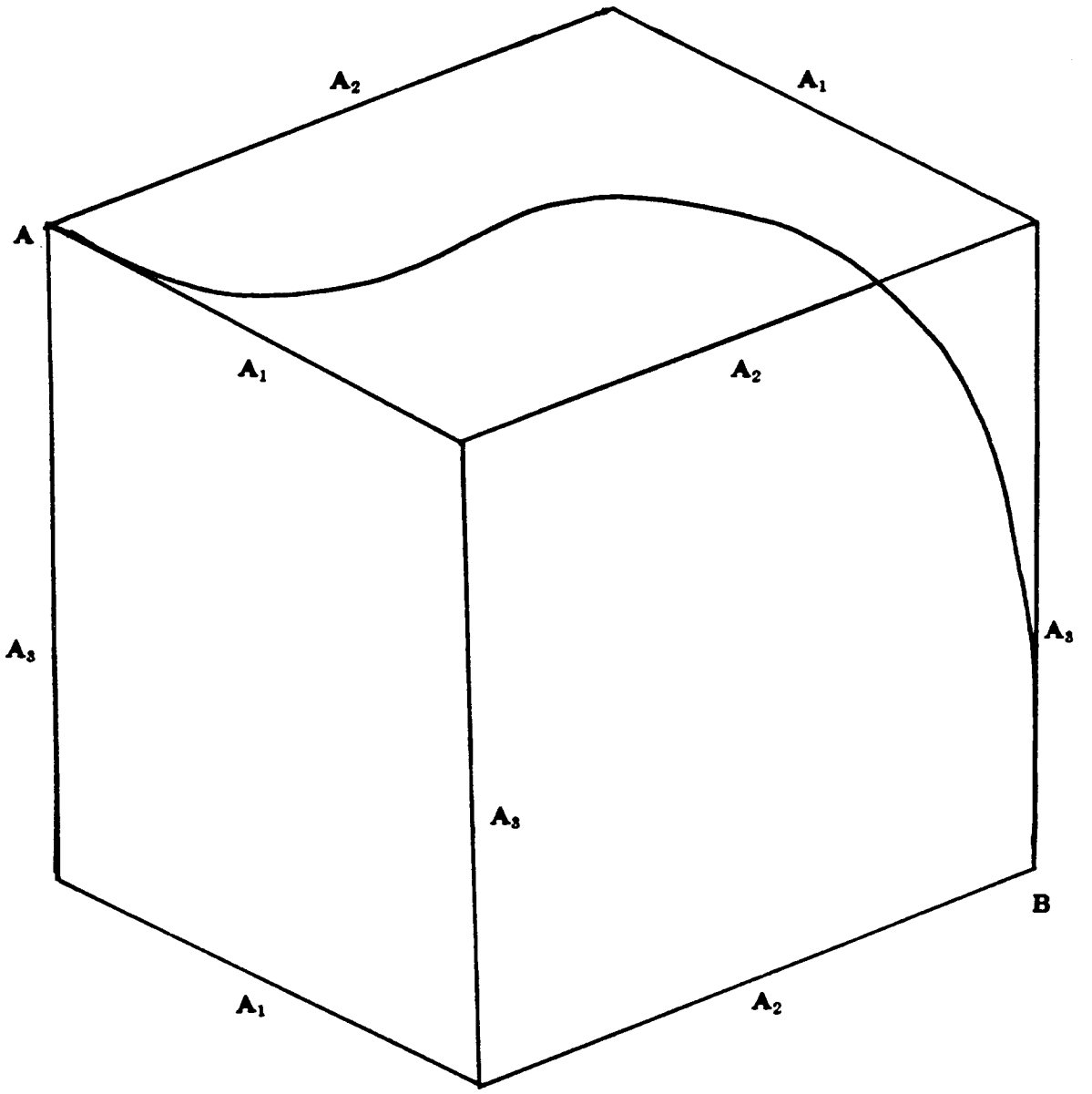


Figure 1. Twisted curve inside a rectangular parallelpiped in three-space.

The end of the curve, $\mathbf{Q(1)}$, is at the point **B**, which corresponds to the vector sum of the four $\mathbf{A}_i's$. This can be obtained by equation (1) if

$$f_i(1) = 1, \quad i = 1, 2, 3 \tag{3}$$

The beginning of the curve has a tangent direction given by $\mathbf{A}_1$, which requires

$$f_i^{(1)}(0) = 0, \quad i = 2, 3 \tag{4}$$

$$where\ f_i^{(p)}(a) = \frac{d^p}{du^p} f(u) \mid_a$$

The end of the curve has a tangent direction given by $\mathbf{A}_3$ which requires

$$f_i^{(1)}(1) = 0, \quad i = 1, 2 \tag{5}$$

The start of the curve lies in the plane whose normal vector is in the $\mathbf{A}_3$ direction; thus,

$$f_3^{(2)}(0) = 0 \tag{6}$$

The end of the curve lies in the plane whose normal vector is in the $\mathbf{A}_1$ direction; thus,

$$f_1^{(2)}(1) = 0 \tag{7}$$

These are twelve constraints on the three functions which is the appropriate number if the functions are cubic polynomials. These twelve conditions can be expressed as a system of simultaneous linear equations. Solving for the coefficients of the three cubic polynomials yields

$$\begin{aligned} f_1(u) &= u^3-3u^2+3u \\ f_2(u) &= -2u^3+3u^2 \\ f_3(u) &= u^3 \end{aligned} \tag{8}$$

These functions are shown in Figure 2.

The cubic degree of these functions arises from the initial choice of a three-dimensional parallelpiped. The same ideas can be generalized to an m-dimensional hyper-parallelpiped. This corresponds to m edges and functions of degree m. Projecting m-dimensional hyperspace onto two- or three-dimensional space yields the following curve form:

$$\mathbf{Q}_m(u) = \mathbf{A} + \sum_{i=1}^{m} f_{i,m}(u)\mathbf{A}_i \qquad 0 \le u \le 1 \tag{9}$$

where the subscript "m" has been added to $f_i(u)$ and $\mathbf{Q}(u)$ to indicate that this is the m^{th} degree case.

The functions $f_{i,m}(u)$ can be determined in a manner analogous to that for the cubic case. The conditions that were given for the cubic case are generalized for the m^{th} degree case as follows:

$$\begin{aligned} f_i(0) &= 0, \quad i = 1, 2, \ldots, m \\ f_i(1) &= 1, \quad i = 1, 2, \ldots, m \\ f_i^{(j)}(0) &= 0, \quad i = j+1, \ldots, m \text{ for } j = 1, \ldots, m-1 \\ f_i^{(j)}(1) &= 0, \quad i = 1, \ldots, m-j \text{ for } j = 1, \ldots, m-1 \end{aligned} \tag{10}$$

The four equations (10) represent $m+m+(m-1)m/2+(m-1)m/2=m(m+1)$ constraints on the m functions $f_{i,m}(u)$, $i=1, \ldots, m$, which is the appropriate number for m^{th} degree functions. These functions are

$$f_{i,m}(u) = \frac{(-u)^i}{(i-1)!} \frac{d^{i-1}}{du^{i-1}}\left[\frac{(1-u)^m-1}{u}\right], \quad i = 1, \ldots, m$$

which can be rewritten as

$$f_{i,m}(u) = \sum_{j=0}^{m}(-1)^{i+j}\begin{bmatrix} m \\ j \end{bmatrix}\begin{Bmatrix} j-1 \\ i-1 \end{Bmatrix} u^j, \quad i = 1, \ldots, m \tag{11}$$

where

$$\begin{bmatrix} r \\ s \end{bmatrix} = 0 \text{ if } r < s$$

These $f_{i,m}(u)$ functions for $m = 1, \ldots, 6$ are given in Table 1 and plotted in Figure 2.

The $f_{i,m}(u)$ functions weight the vectors $\mathbf{A}_i$. For curve design, these vectors can be two- or three-dimensional and are arranged from head to tail in the following manner: $\mathbf{A}_1$ starts at $\mathbf{A}$ and then each $\mathbf{A}_i$ is placed with its tail at the same point as the head of $\mathbf{A}_i$ for $i = 1, \ldots, m$. This is shown for the two-dimensional case in Figure 3.

m	i	$f_{i,m}(u)$
1	1	u
2	1	$-u^2+2u$
2	2	u^2
3	1	u^3-3u^2+3u
3	2	$-2u^3+3u^2$
3	3	u^3
4	1	$-u^4+4u^3-6u^2+4u$
4	2	$3u^4-8u^3+6u^2$
4	3	$-3u^4+4u^3$
4	4	u^4
5	1	$u^5-5u^4+10u^3-10u^2+5u$
5	2	$-4u^5+15u^4-20u^3+10u^2$
5	3	$6u^5-15u^4+10u^3$
5	4	$-4u^5+5u^4$
5	5	u^5
6	1	$-u^6+6u^5-15u^4+20u^3-15u^2+6u$
6	2	$5u^6-24u^5+45u^4-40u^3+15u^2$
6	3	$-10u^6+36u^5-45u^4+20u^3$
6	4	$10u^6-24u^5+15u^4$
6	5	$-5u^6+6u^5$
6	6	u^6

Table 1. The $f_{i,m}(u)$ functions for $m = 1, \ldots, 6$.

Instead of specifying the vectors $\mathbf{A}_i$, the actual points corresponding to the locations of the heads of these vectors in this coordinate system could be used. These points are called *control vertices*, and form a *control polygon*. The control vertex corresponding to the position of the head of the vector $\mathbf{A}_i$ is denoted $\mathbf{V}_i$. In this coordinate system, the vector $\mathbf{A}_i$ can be written as

$$\mathbf{A}_i = \mathbf{V}_i - \mathbf{V}_{i-1}, \quad i = 1, \ldots, m$$

$$\text{and } \mathbf{A} = \mathbf{V}_0$$

Substituting into (9)

$$\mathbf{Q}_m(u) = \mathbf{V}_0 + \sum_{i=1}^{m} f_{i,m}(u)(\mathbf{V}_i - \mathbf{V}_{i-1}) \qquad 0 \le u \le 1 \tag{12}$$

$$\mathbf{Q}_m(u) = \sum_{i=0}^{m-1} (f_{i,m}(u) - f_{i+1,m}(u))\mathbf{V}_i + f_{m,m}(u)\mathbf{V}_m \qquad 0 \le u \le 1 \tag{13}$$

where

$$f_{0,m}(u) = 1$$

Considering the difference of the functions

$$f_{i,m}(u) - f_{i+1,m}(u), \quad i = 0, \ldots, m-1$$

$$= \sum_{j=0}^{m} (-1)^{i+j} \binom{m}{j} \binom{j-1}{i-1} u^j - \sum_{j=0}^{m} (-1)^{i+j+1} \binom{m}{j} \binom{j-1}{i} u^j$$

$$= \sum_{j=0}^{m} (-1)^{i+j} \binom{m}{j} u^j \left[\binom{j-1}{i-1} + \binom{j-1}{i} \right]$$

Note that the expression in the square brackets is a combinatoric identity; thus this sum becomes

$$f_{i,m}(u) - f_{i+1,m}(u) = \sum_{j=0}^{m} (-1)^{i+j} \binom{m}{j} \binom{j}{i} u^j, \quad i = 0, \ldots, m-1$$

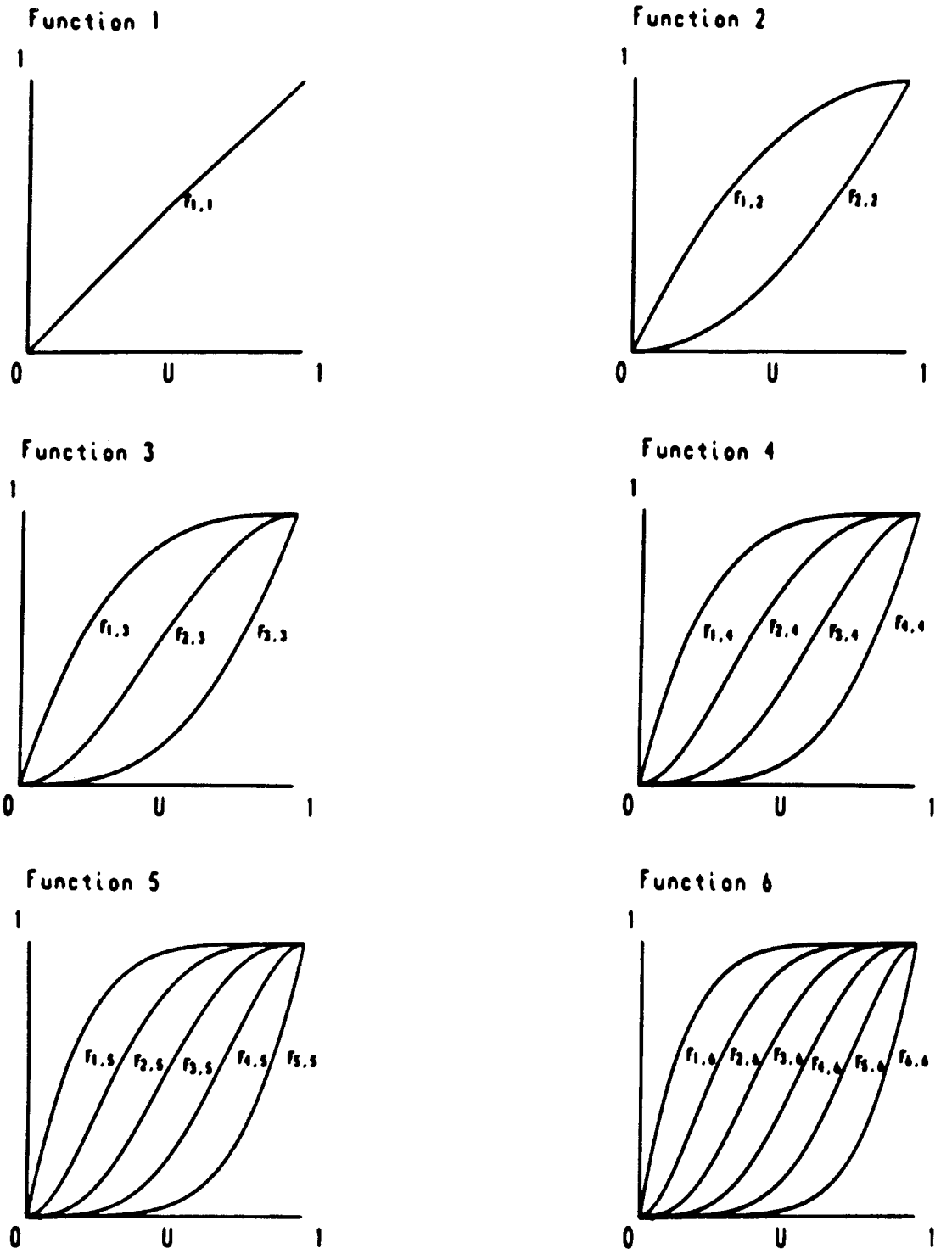


Figure 2. The parametric functions $f_{i,m}(u)$ for $m = 1, \ldots, 6$.

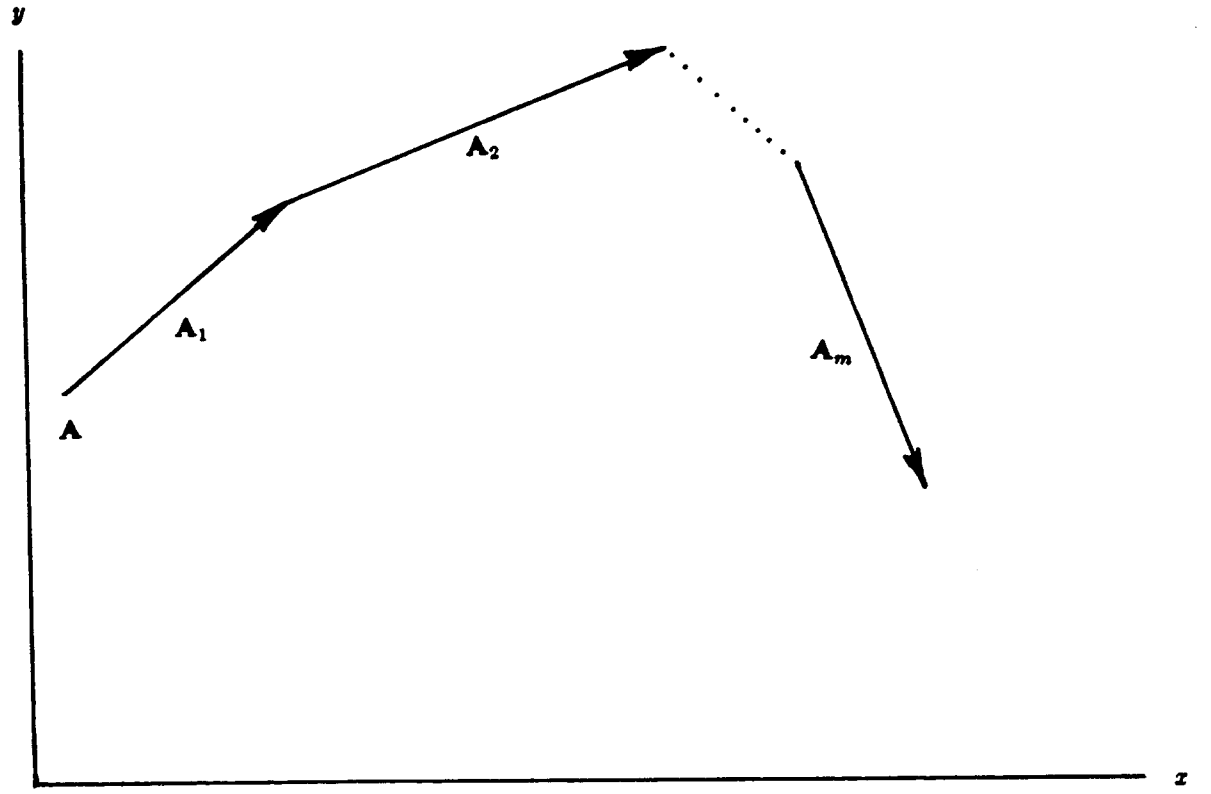


Figure 3. The vectors $\mathbf{A}_i$ arranged from head to tail in the two-dimensional case.

Noting that $\binom{m}{j}\binom{j}{i}=\binom{m}{i}\binom{m-i}{j-i}$ and $(-1)^{i+j} = (-1)^{j-i}$, the sum can be rewritten as

$$= \binom{m}{i}\sum_{j=0}^{m}(-1)^{j-i}\binom{m-i}{j-i}u^j, \quad i = 0, \ldots, m-1$$

Changing the lower limit on the summation to eliminate zero terms and simplifying,

$$= \binom{m}{i}u^i\sum_{j=i}^{m}(-1)^{j-i}\binom{m-i}{j-i}u^{j-i}, \quad i = 0, \ldots, m-1$$

which is the binomial expansion of

$$= \binom{m}{i}u^i(1-u)^{m-i} \tag{14}$$

that is, the Bernstein polynomial

$$= B_{i,m}(u), \quad i = 0, \ldots, m-1 \tag{15}$$

Substituting into (13),

$$\mathbf{Q}_m(u) = \sum_{i=0}^{m-1}B_{i,m}(u)\mathbf{V}_i + f_{m,m}(u)\mathbf{V}_m \qquad 0\leq u\leq 1 \tag{16}$$

This expression (16) has the $\mathbf{V}_m$ term separated from the rest of the summation. This can be remedied by considering the expression for $f_{m,m}(u)$. Evaluating equation (11) for $i=m$ yields all the terms of the summation equal to zero except at $j=m$; hence $f_{m,m}(u)=u^m$. However, this is

exactly $B_{m,m}(u)$; thus, equation (16) can be rewritten in the form of a single summation as follows:

$$\mathbf{Q}_m(u) = \sum_{i=0}^{m} B_{i,m}(u)\mathbf{V}_i \qquad 0 \le u \le 1 \tag{17}$$

The Bernstein polynomials, $B_{i,m}(u)$, are plotted for $m = 1, \ldots, 6$ in Figure 4.

3. Convex Combination of Lower Degree Curves

The notation for an m^{th} degree Bézier curve is now extended to include a list of the control vertices that define it:

$$\mathbf{Q}_m(\mathbf{V}_0, \mathbf{V}_1, \ldots, \mathbf{V}_m; u)$$

Before deriving the equations governing the arbitrary subdivision of Bézier curves, it will be shown that an m^{th} degree Bézier curve is a *convex combination* of a pair of $(m-1)^{st}$ degree curves. In particular, the curve for $[\mathbf{V}_0, \ldots, \mathbf{V}_m]$ is a convex combination of the curves for $[\mathbf{V}_0, \ldots, \mathbf{V}_{m-1}]$ and $[\mathbf{V}_1, \ldots, \mathbf{V}_m]$ with convex combination coefficients u and $1-u$ (Figure 5):

$$\mathbf{Q}_m(\mathbf{V}_0, \mathbf{V}_1, \ldots, \mathbf{V}_m; u) = (1-u)\mathbf{Q}_{m-1}(\mathbf{V}_0, \mathbf{V}_1, \ldots, \mathbf{V}_{m-1}; u) + u\,\mathbf{Q}_{m-1}(\mathbf{V}_1, \mathbf{V}_2, \ldots, \mathbf{V}_m; u) \tag{18}$$

To show this,[7] recall the expression (17) for the curve $\mathbf{Q}_m(\mathbf{V}_0, \mathbf{V}_1, \ldots, \mathbf{V}_m; u)$:

$$\mathbf{Q}_m(\mathbf{V}_0, \mathbf{V}_1, \ldots, \mathbf{V}_m; u) = \sum_{i=0}^{m} B_{i,m}(u)\mathbf{V}_i$$

Substituting the expression (14) for the Bernstein basis function $B_{i,m}(u)$

$$= \sum_{i=0}^{m} \binom{m}{i} u^i (1-u)^{m-i} \mathbf{V}_i$$

expanding using a combinatoric identity,

$$= \sum_{i=0}^{m} \left[\binom{m-1}{i} + \binom{m-1}{i-1} \right] u^i (1-u)^{m-i} \mathbf{V}_i$$

and separating into two sums,

$$= \sum_{i=0}^{m} \binom{m-1}{i} u^i (1-u)^{m-i} \mathbf{V}_i + \sum_{i=0}^{m} \binom{m-1}{i-1} u^i (1-u)^{m-i} \mathbf{V}_i$$

Each of the summations has a term where the binomial coefficient is zero; specifically, $\binom{m-1}{m} = 0$ and $\binom{m-1}{-1} = 0$. When these terms are removed, the limits of the sums are changed as follows:

$$= \sum_{i=0}^{m-1} \binom{m-1}{i} u^i (1-u)^{m-i} \mathbf{V}_i + \sum_{i=1}^{m} \binom{m-1}{i-1} u^i (1-u)^{m-i} \mathbf{V}_i \tag{19}$$

Now, the second summation can be rewritten by changing the index of summation. Replacing i with $i' = i-1$ $(i = i'+1)$ in this summation yields:

$$\sum_{i=1}^{m} \binom{m-1}{i-1} u^i (1-u)^{m-i} \mathbf{V}_i = \sum_{i'=0}^{m-1} \binom{m-1}{i'} u^{i'+1} (1-u)^{m-1-i'} \mathbf{V}_{i'+1} \tag{20}$$

Then, substituting this new expression (20) for the second summation in equation (19) yields:

$$\mathbf{Q}_m(\mathbf{V}_0, \mathbf{V}_1, \ldots, \mathbf{V}_m; u) = \sum_{i=0}^{m-1} \binom{m-1}{i} u^i (1-u)^{m-i} \mathbf{V}_i + \sum_{i'=0}^{m-1} \binom{m-1}{i'} u^{i'+1} (1-u)^{m-1-i'} \mathbf{V}_{i'+1}$$

$$= (1-u) \sum_{i=0}^{m-1} \binom{m-1}{i} u^i (1-u)^{m-1-i} \mathbf{V}_i + u \sum_{i'=0}^{m-1} \binom{m-1}{i'} u^{i'} (1-u)^{m-1-i'} \mathbf{V}_{i'+1} \tag{21}$$

which is exactly equation (18).

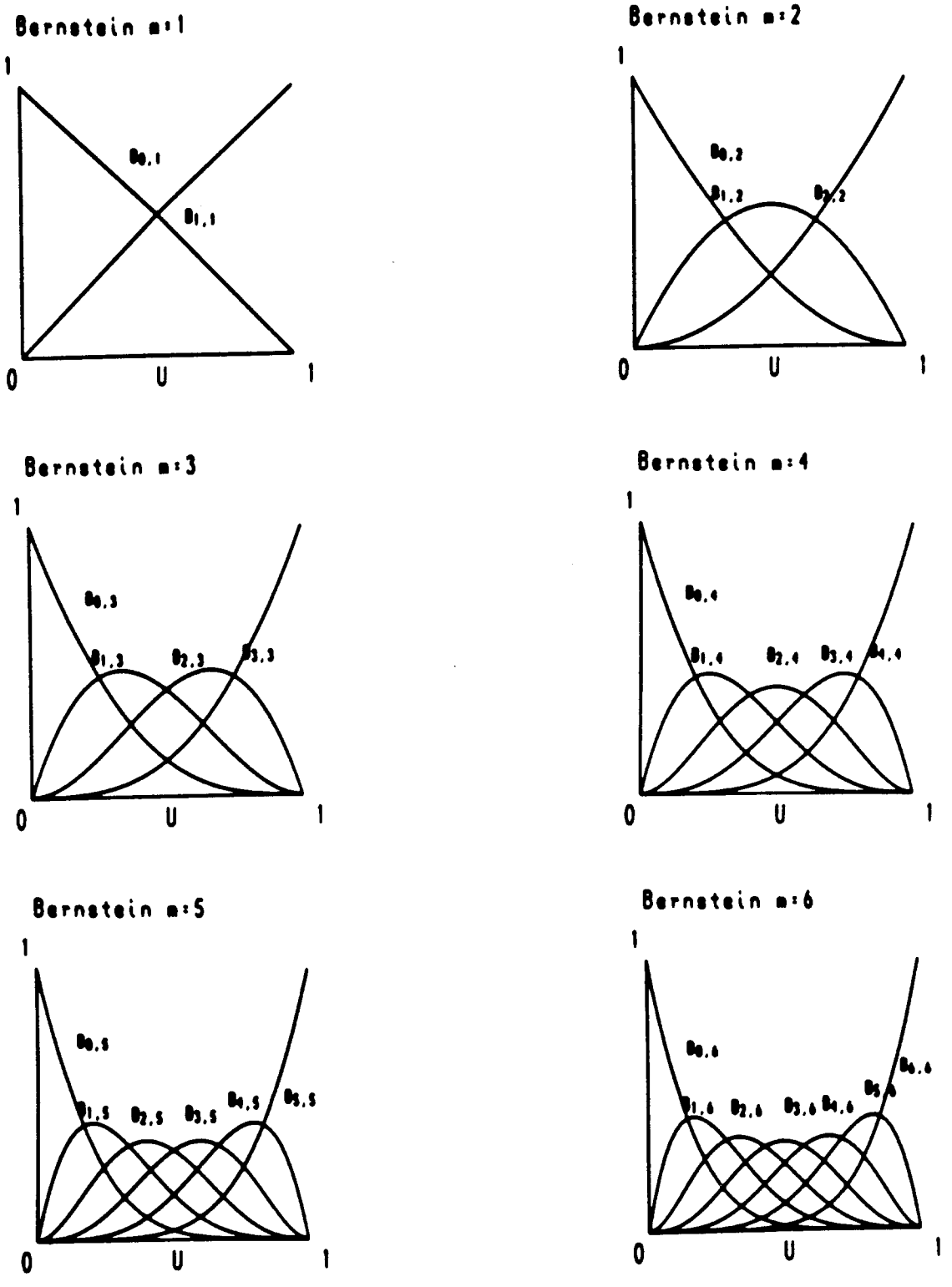


Figure 4. The Bernstein polynomials for $m = 1, \ldots, 6$.

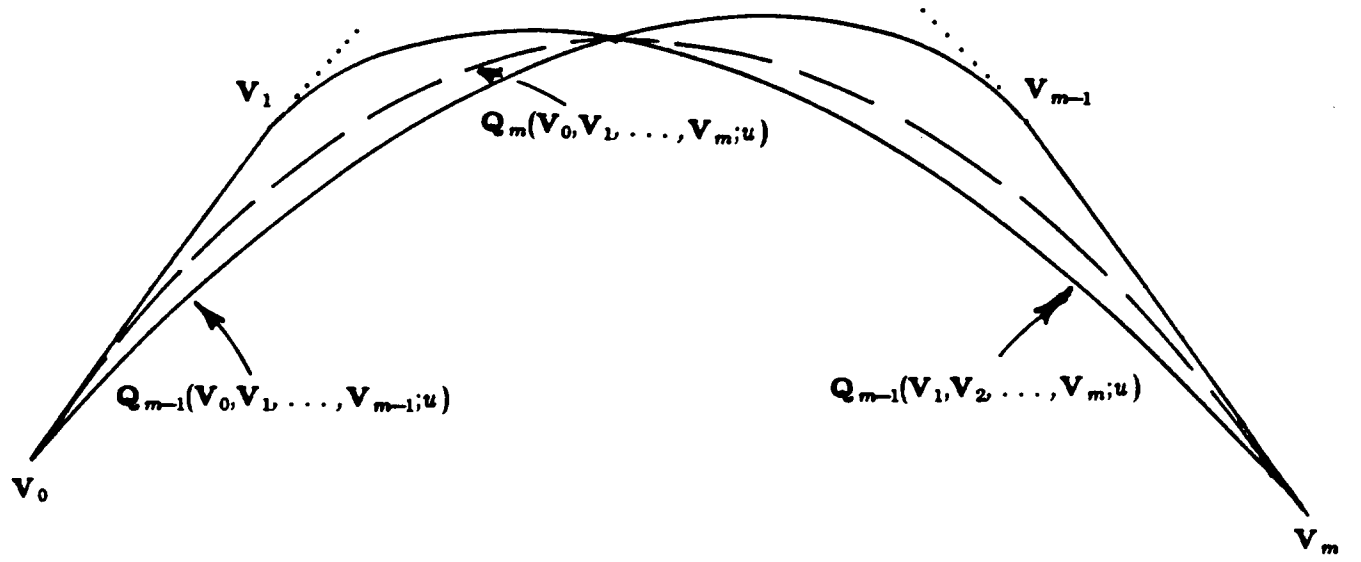


Figure 5. An m^{th} degree Bézier curve is a convex combination of a pair of $(m-1)^{st}$ degree curves.

4. New Vertices and Geometric Construction

Considering equation (21) and combining the summations

$$\mathbf{Q}_m(\mathbf{V}_0,\mathbf{V}_1,\ldots,\mathbf{V}_m;u) = \sum_{i=0}^{m-1} B_{i,m-1}(u)[(1-u)\mathbf{V}_i + u\mathbf{V}_{i+1}] \tag{22}$$

Equation (22) can be rewritten as

$$\mathbf{Q}_m(\mathbf{V}_0,\mathbf{V}_1,\ldots,\mathbf{V}_m;u) = \sum_{i=0}^{m-1} B_{i,m-1}(u)\mathbf{V}_i^{[1]}(u)$$

where

$$\mathbf{V}_i^{[1]}(u) = (1-u)\mathbf{V}_i + u\mathbf{V}_{i+1}$$

Repeating this process recursively k times yields

$$\mathbf{Q}_m(\mathbf{V}_0,\mathbf{V}_1,\ldots,\mathbf{V}_m;u) = \sum_{i=0}^{m-k} B_{i,m-k}(u)\mathbf{V}_i^{[k]}(u) \tag{23}$$

where

$$\mathbf{V}_i^{[k]}(u) = \left\{ \begin{matrix} (1-u)\mathbf{V}_i^{[k-1]}(u) + u\mathbf{V}_{i+1}^{[k-1]}(u), \\ \mathbf{V}_i \end{matrix} \quad \begin{matrix} k=1,\ldots,m \\ k=0 \end{matrix} \right\} \tag{24}$$

In the case of $k=m$, equation (23) becomes

$$\mathbf{Q}_m(\mathbf{V}_0,\mathbf{V}_1,\ldots,\mathbf{V}_m;u) = B_{0,0}(u)\mathbf{V}_0^{[m]}(u)$$

which is

$$\mathbf{Q}_m(\mathbf{V}_0,\mathbf{V}_1,\ldots,\mathbf{V}_m;u) = \mathbf{V}_0^{[m]}(u)$$

Thus, for a given parametric value u^*, the point on the curve at u^* is $\mathbf{V}_0^{[m]}(u^*)$, as defined in equation (24). This yields another way to compute a point on the curve; simply recursively compute this vertex using a ratio equal to the parametric value of the desired point. This idea can be used to geometrically construct a Bézier curve. To compute the point $\mathbf{Q}(u^*)$, each edge of the control polygon is divided in $u^*:1-u^*$ ratios and these points are connected in succession to form $m-1$ edges. This process is continued, and after m iterations, a single point results which is the desired value. Figure 6 illustrates the computation of a point on a cubic Bézier curve. This

same process can be performed for various values of the parameter u, and then these points can be connected to generate a piecewise linear approximation to the curve. An example of this process showing a cubic curve is given in Section 6 and illustrated in Figure 7.

In addition, $\mathbf{V}_0^{[m]}(u^*)$ is the common vertex between two subdivided curves, each with its own control polygon. This is derived in the next section and is illustrated in Figure 6 for degree $m=3$.

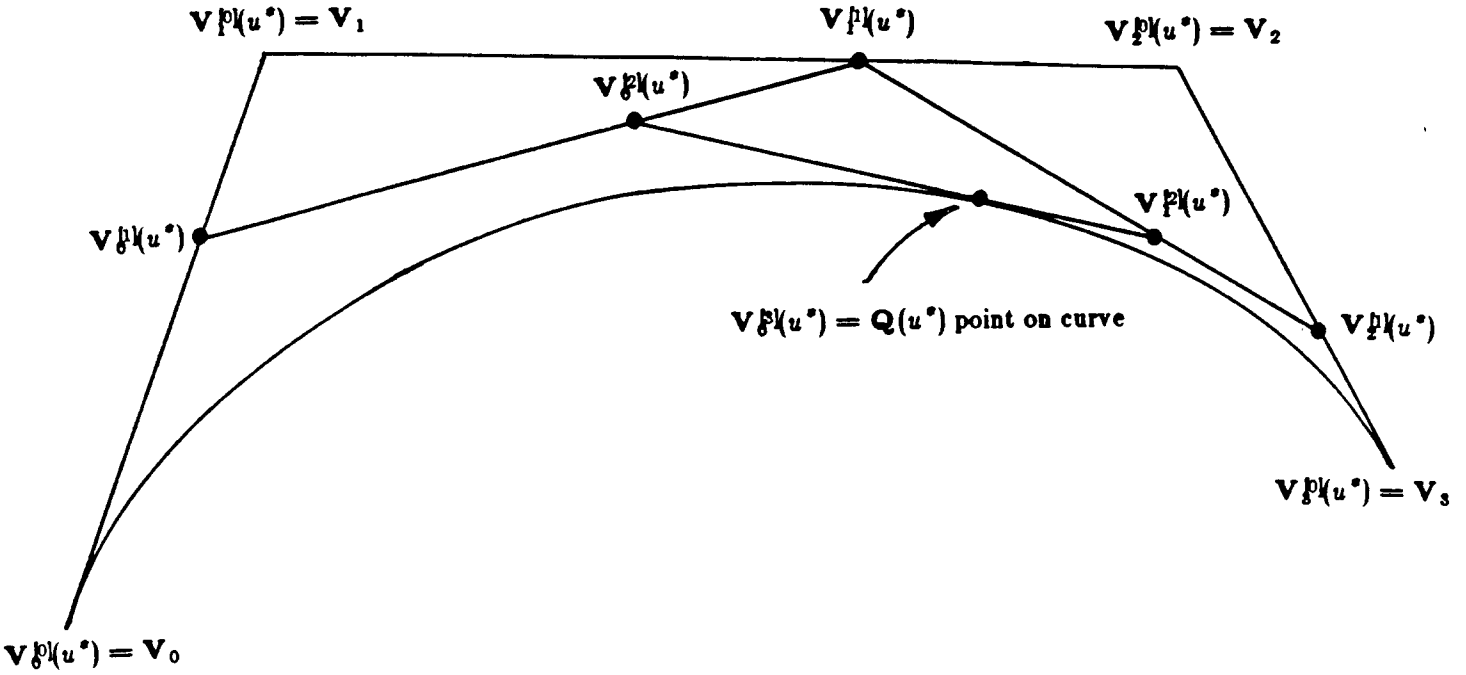


Figure 6. $\mathbf{V}_0^{[3]}(u^*)$ is the common vertex between the two subdivided curves for degree $m=3$.

5. Mathematical Theory of Arbitrary Subdivision

The notation for a curve will now be extended to include the parametric limits corresponding to the initial and terminal points of the curve. (Note that these parametric limits are not necessarily the same as the parametric limits defining the interval over which the parameter is allowed to vary.) In this manner, the curve defined by the $m+1$ control vertices $\mathbf{V}_0, \ldots, \mathbf{V}_m$ and beginning at $u=a$ and ending at $u=b$ is given by

$$\mathbf{Q}_m(\mathbf{V}_0, \mathbf{V}_1, \ldots, \mathbf{V}_m; a, b; u)$$

The two subdivided curves are then

$$\mathbf{Q}_m(\mathbf{V}_0^{[0]}(u^*), \mathbf{V}_0^{[1]}(u^*), \ldots, \mathbf{V}_0^{[m]}(u^*); 0, u^*; \frac{u}{u^*})$$

and

$$\mathbf{Q}_m(\mathbf{V}_0^{[m]}(u^*), \mathbf{V}_1^{[m-1]}(u^*), \ldots, \mathbf{V}_m^{[0]}(u^*); u^*, 1; \frac{u-u^*}{1-u^*})$$

Figure 6 illustrates these two curves for $m=3$.

It will now be shown that these two subdivided curves are coincident with the original curve, for the same parametric values; that is:

$$\mathbf{Q}_m(\mathbf{V}_0, \mathbf{V}_1, \ldots, \mathbf{V}_m; 0, 1; u) = \begin{cases} \mathbf{Q}_m(\mathbf{V}_0^{[0]}(u^*), \mathbf{V}_0^{[1]}(u^*), \ldots, \mathbf{V}_0^{[m]}(u^*); 0, u^*; \frac{u}{u^*}) \text{ for } 0 \le u \le u^* \\ \mathbf{Q}_m(\mathbf{V}_0^{[m]}(u^*), \mathbf{V}_1^{[m-1]}(u^*), \ldots, \mathbf{V}_m^{[0]}(u^*); u^*, 1; \frac{u-u^*}{1-u^*}) \text{ for } u^* \le u \le 1 \end{cases} \quad (25)$$

Equation (25) will now be proven for $0 \leq u \leq u^*$; the case of $u^* \leq u \leq 1$ can be accomplished in an analogous fashion. To prove this by induction on m, let $m=1$. Then from equation (18) with $m=1$, the righthand side can be rewritten as

$$(1-\frac{u}{u^*})\mathbf{V}_0^{[0]}(u^*) + \frac{u}{u^*}\mathbf{V}_0^{[1]}(u^*)$$

$$= (1-\frac{u}{u^*})\mathbf{V}_0 + \frac{u}{u^*}((1-u^*)\mathbf{V}_0 + u^*\mathbf{V}_1)$$

$$= (1-u)\mathbf{V}_0 + u\mathbf{V}_1$$

$$= \mathbf{Q}_1(\mathbf{V}_0,\mathbf{V}_1;0,1;u)$$

which is the lefthand side and hence establishes the basis case.

Now, for the induction step, assume that the equation (25) is true for $k=1, \ldots, m-1$. Recall that

$$\mathbf{Q}_m(\mathbf{V}_0, \ldots, \mathbf{V}_m;0,1;u)$$

can be written as

$$(1-u)\,\mathbf{Q}_{m-1}(\mathbf{V}_0, \ldots, \mathbf{V}_{m-1};0,1;u) + u\,\mathbf{Q}_{m-1}(\mathbf{V}_1, \ldots, \mathbf{V}_m;0,1;u)$$

By the induction hypothesis (25), this becomes

$$(1-u)\,\mathbf{Q}_{m-1}(\mathbf{V}_0^{[0]}(u^*),\mathbf{V}_0^{[1]}(u^*), \ldots, \mathbf{V}_0^{[m-1]}(u^*);0,u^*;\frac{u}{u^*}) +$$

$$u\,\mathbf{Q}_{m-1}(\mathbf{V}_1^{[0]}(u^*),\mathbf{V}_1^{[1]}(u^*), \ldots, \mathbf{V}_1^{[m-1]}(u^*);0,u^*;\frac{u}{u^*})$$

This can be expanded as follows:

$$(1-\frac{u}{u^*})\mathbf{Q}_{m-1}(\mathbf{V}_0^{[0]}(u^*),\mathbf{V}_0^{[1]}(u^*), \ldots, \mathbf{V}_0^{[m-1]}(u^*);0,u^*;\frac{u}{u^*}) +$$

$$\frac{u}{u^*}[u^*\mathbf{Q}_{m-1}(\mathbf{V}_1^{[0]}(u^*),\mathbf{V}_1^{[1]}(u^*), \ldots, \mathbf{V}_1^{[m-1]}(u^*);0,u^*;\frac{u}{u^*}) +$$

$$(1-u^*)\mathbf{Q}_{m-1}(\mathbf{V}_0^{[0]}(u^*),\mathbf{V}_0^{[1]}(u^*), \ldots, \mathbf{V}_0^{[m-1]}(u^*);0,u^*;\frac{u}{u^*})]$$

Regrouping, this is

$$(1-\frac{u}{u^*})\mathbf{Q}_{m-1}(\mathbf{V}_0^{[0]}(u^*),\mathbf{V}_0^{[1]}(u^*), \ldots, \mathbf{V}_0^{[m-1]}(u^*);0,u^*;\frac{u}{u^*}) +$$

$$\frac{u}{u^*}\mathbf{Q}_{m-1}((1-u^*)\mathbf{V}_0^{[0]} + u^*\mathbf{V}_1^{[0]}(u^*),$$

$$(1-u^*)\mathbf{V}_0^{[1]} + u^*\mathbf{V}_1^{[1]}(u^*), \ldots,$$

$$(1-u^*)\mathbf{V}_0^{[m-1]} + u^*\mathbf{V}_1^{[m-1]}(u^*);0,u^*;\frac{u}{u^*})$$

By the definition (24) of $\mathbf{V}_1^{[k]}(u^*)$, this becomes

$$(1-\frac{u}{u^*})\mathbf{Q}_{m-1}(\mathbf{V}_0^{[0]}(u^*),\mathbf{V}_0^{[1]}(u^*), \ldots, \mathbf{V}_0^{[m-1]}(u^*);0,u^*;\frac{u}{u^*}) +$$

$$\frac{u}{u^*}\mathbf{Q}_{m-1}(\mathbf{V}_0^{[1]}(u^*),\mathbf{V}_0^{[2]}(u^*), \ldots, \mathbf{V}_0^{[m]}(u^*);0,u^*;\frac{u}{u^*})$$

But from (18), this is just

$$\mathbf{Q}_m(\mathbf{V}_0^{[0]}(u^*),\mathbf{V}_0^{[1]}(u^*), \ldots, \mathbf{V}_0^{[m]}(u^*);0,u^*;\frac{u}{u^*})$$

which substantiates the claim (25) for $0 \leq u \leq u^*$. A similar analysis will verify the claim for $u^* \leq u \leq 1$. Thus, the two subdivided curves defined by the new control vertices given in (24) together form a curve that is coincident with the original curve.

6. Illustration of Arbitrary Subdivision

An example illustrating the arbitrary subdivision process is given in Figure 7. In this example, the original control polygon consists of four vertices, as shown in Figure 7(i). First, the geometric construction is applied to this polygon.

The parametric value u^*, at which the subdivision is to occur, reflects the lengths of the polygon edges in geometric space. Specifically, it is ratio of the length of the "left" half of the control polygon to its total length; that is,

$$u^* = \frac{|V_0V_1| + |V_1V_2|/2}{|V_0V_1| + |V_1V_2| + |V_2V_3|} \tag{26}$$

The parameter *step* describes the construction process. At step k, the vertices $\mathbf{V}_j^{[k]}(u^*)$, $j=0, \ldots, 3-k$ are determined. Figure 7(ii) shows step 0, where the vertices are the four original control vertices. Then, step 1 yields three new vertices (Figure 7 (ii)) and step 2 yields two new vertices (Figure 7 (iii)). Finally, Figure 7 (iv) shows the point on the curve, $\mathbf{V}_0^{[3]}(u^*)$, which is determined at step 3. This completes the geometric construction for the point on the curve from the original control polygon. Now, the same process is performed recursively on the new control vertices. For this construction, a new value of u^* will be computed in terms of each new set of control vertices.

This recursive subdivision corresponds to a preorder traversal of a tree. For this reason, the subdivision stages can be labeled in correspondence with the nodes in the tree. For each *subdivision* in sequence, *level* and *branch* define the node as follows: level is the depth of the node in the tree and branch indicates which specific node at that level by a left/right code (except at level 0 where branch is undefined).

The depth of the tree determines the precision of the subdivision; the deeper the tree, the closer the new control vertices will be to the true curve. *Flatness* is a natural geometric criterion for quantifying the precision of the approximation.[2] For the simplicity of this example, however, the depth of the tree will be cut off at 2 across the entire tree. Note that in general, the depth of the tree would not be the same at all the leaves. In fact, this arbitrary cutoff yields a very poor final approximation (Figure 7 (xxvii)) to the curve. Use of a reasonable stopping criterion, such as flatness, would remedy this defect.

Since the depth of the tree is 2, it has seven nodes; hence the subdivision numbers range from 0 to 6, the level from 0 to 2, and branch is L or R at level 1, and LL, LR, RL, or RR at level 2. The specific values of each of these parameters, in the order of tree traversal, are given in Table 2, and the tree is shown in Figure 8.

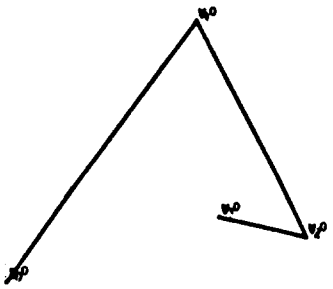
Subdivision	Level	Branch	Figures
0	0	-	7(i)-7(iv)
1	1	L	7(v)-7(vii)
2	2	LL	7(viii)-7(xi)
3	2	LR	7(xii)-7(xvi)
4	1	R	7(xvii)-7(xix)
5	2	RL	7(xx)-7(xxiii)
6	2	RR	7(xxiv)-7(xxvii)

Table 2. Subdivision number, level, and branch values for the example.

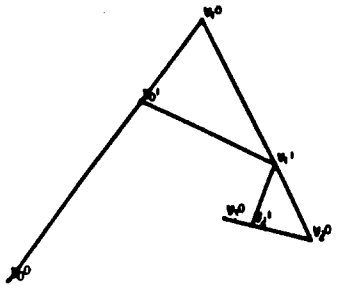
The images in Figure 7 show the complete subdivision process. After the geometric construction is completed for the original control polygon, the construction is performed on the

"left" control polygon, $\mathbf{V}_\theta^{[0]}(u^*), \mathbf{V}_\theta^{[1]}(u^*), \ldots, \mathbf{V}_\theta^{[3]}(u^*)$, as shown in Figures 7(v) through 7(vii). This process is repeated for subdivision 2 in Figures 7(viii) through 7(x); Figure 7(xi) shows the addition of the approximation to the curve. Subdivision 3 is shown in Figures 7(xii) through 7(xiv) with the curve drawn in Figure 7(xv). Figure 7(xvi) shows just the approximation of the curve without the display of the intermediate vertices. Figures 7(xvii) through 7(xix) show subdivision 4. Subdivision 5 is shown in Figures 7(xx) through 7(xxii) and the approximation to the curve is added in Figure 7(xxiii). Finally, Figures 7(xxiv) through 7(xxvi) show subdivision 6 and Figure 7(xxvii) adds the approximation to the curve. The final curve without the display of intermediate control vertices is shown in Figure 7(xxviii).

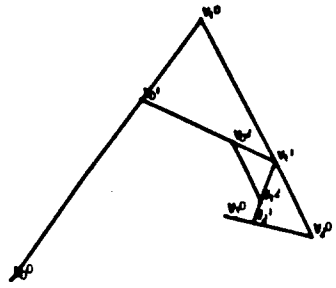
(i)
Subdivision 0
Level 0
Branch -
Step 0



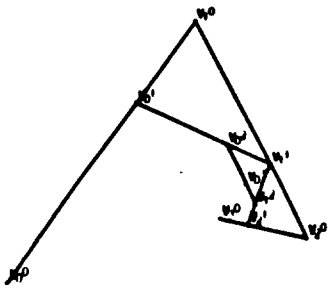
(ii)
Subdivision 0
Level 0
Branch -
Step 1



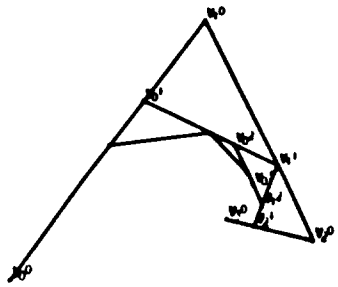
(iii)
Subdivision 0
Level 0
Branch -
Step 2



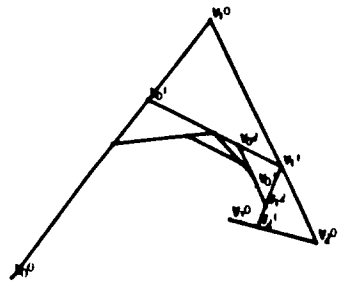
(iv)
Subdivision 0
Level 0
Branch -
Step 3



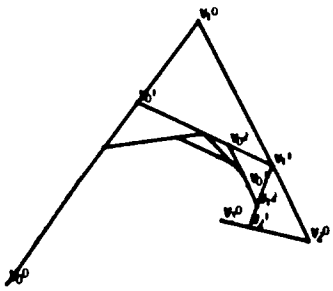
(v)
Subdivision 1
Level 1
Branch L
Step 1



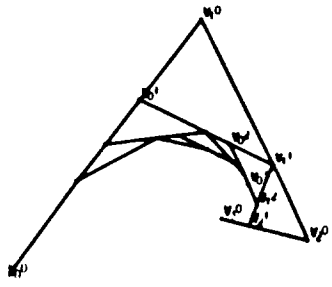
(vi)
Subdivision
Level 1
Branch L
Step 2



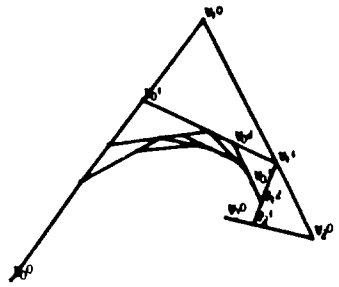
(vii)
Subdivision 1
Level 1
Branch L
Step 3



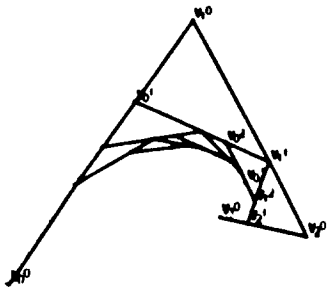
(viii)
Subdivision 2
Level 2
Branch LL
Step 1



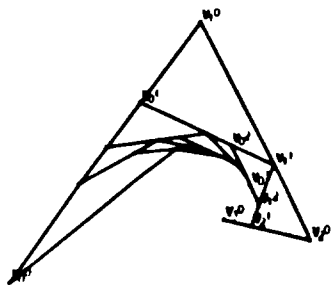
(ix)
Subdivision 2
Level 2
Branch LL
Step 2



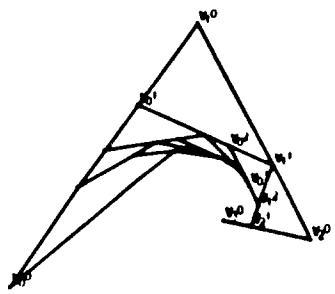
(x)
Subdivision 2
Level 2
Branch LL
Step 3



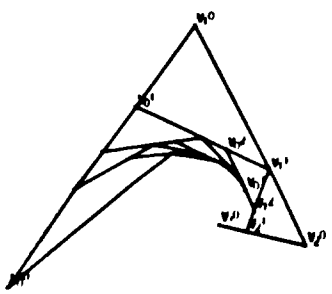
(xi)
Subdivision 2
Level 2
Branch LL
Step 3
Curve Drawn



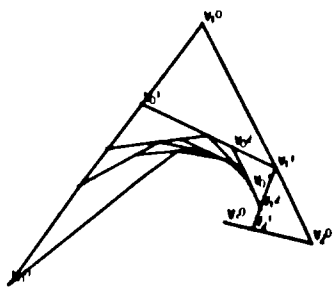
(xii)
Subdivision 3
Level 2
Branch LR
Step 1



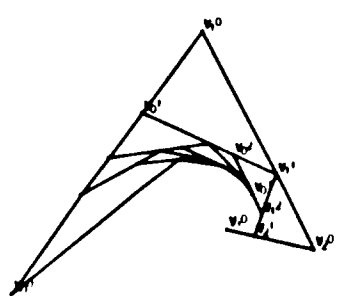
(xiii)
Subdivision 3
Level 2
Branch LR
Step 2



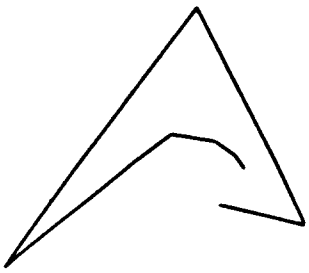
(xiv)
Subdivision 3
Level 2
Branch LR
Step 3



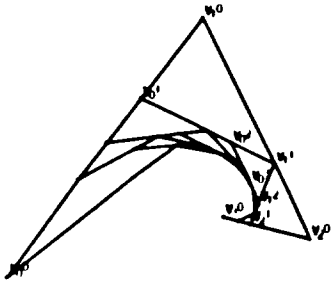
(xv)
Subdivision 3
Level 2
Branch LR
Step 3
Curve Drawn



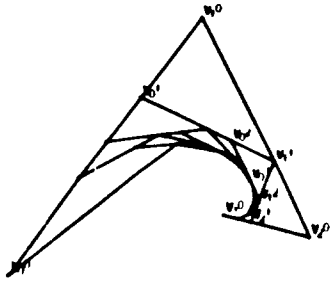
(xvi)
Subdivision 3
Level 2
Branch LR
Step 3
Curve Only



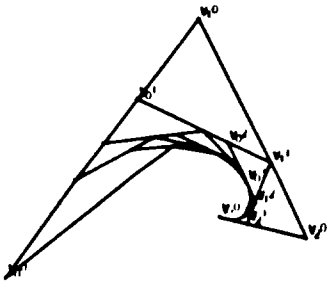
(xvii)
Subdivision 4
Level 1
Branch R
Step 1



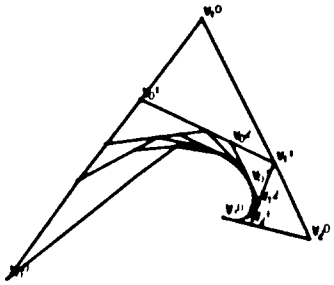
(xviii)
Subdivision
Level 1
Branch R
Step 2



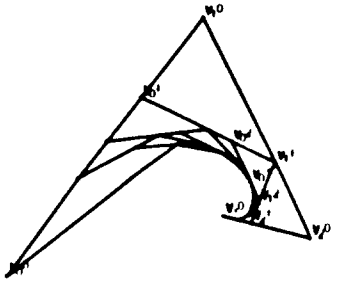
(xix)
Subdivision 4
Level 1
Branch R
Step 3



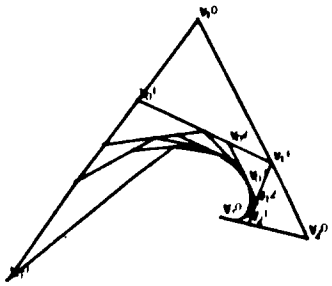
(xx)
Subdivision 5
Level 2
Branch RL
Step 1



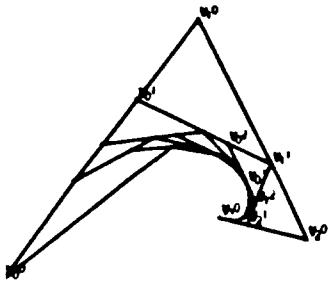
(xxi)
Subdivision
Level 2
Branch RL
Step 2



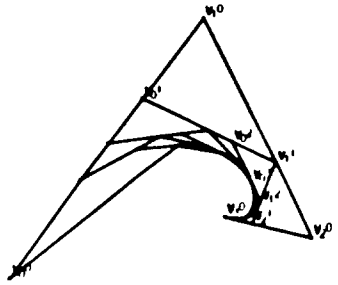
(xxii)
Subdivision 5
Level 2
Branch RL
Step 3



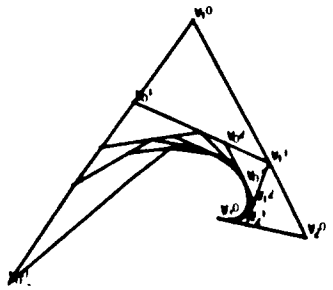
(xxiii)
Subdivision 5
Level 2
Branch RL
Step 3
Curve Drawn



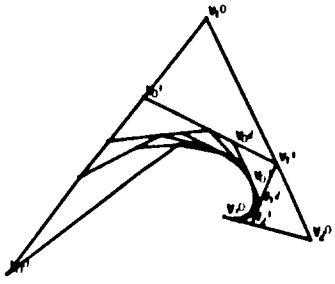
(xxiv)
Subdivision 6
Level 2
Branch RR
Step 1



(xxv)
Subdivision 6
Level 2
Branch RR
Step 2



(xxvi)
Subdivision 6
Level 2
Branch RR
Step 3



(xxvii)
Subdivision 6
Level 2
Branch RR
Step 3
Curve Drawn

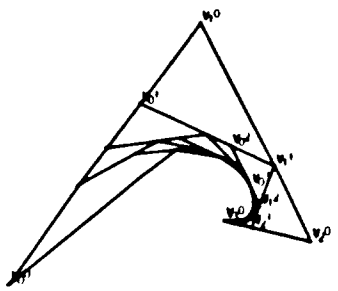
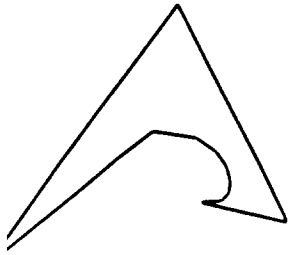


Figure 7. Example of arbitrary subdivision.

(xxviii)
Subdivision
Level 2
Branch RR
Step 3
Curve Only



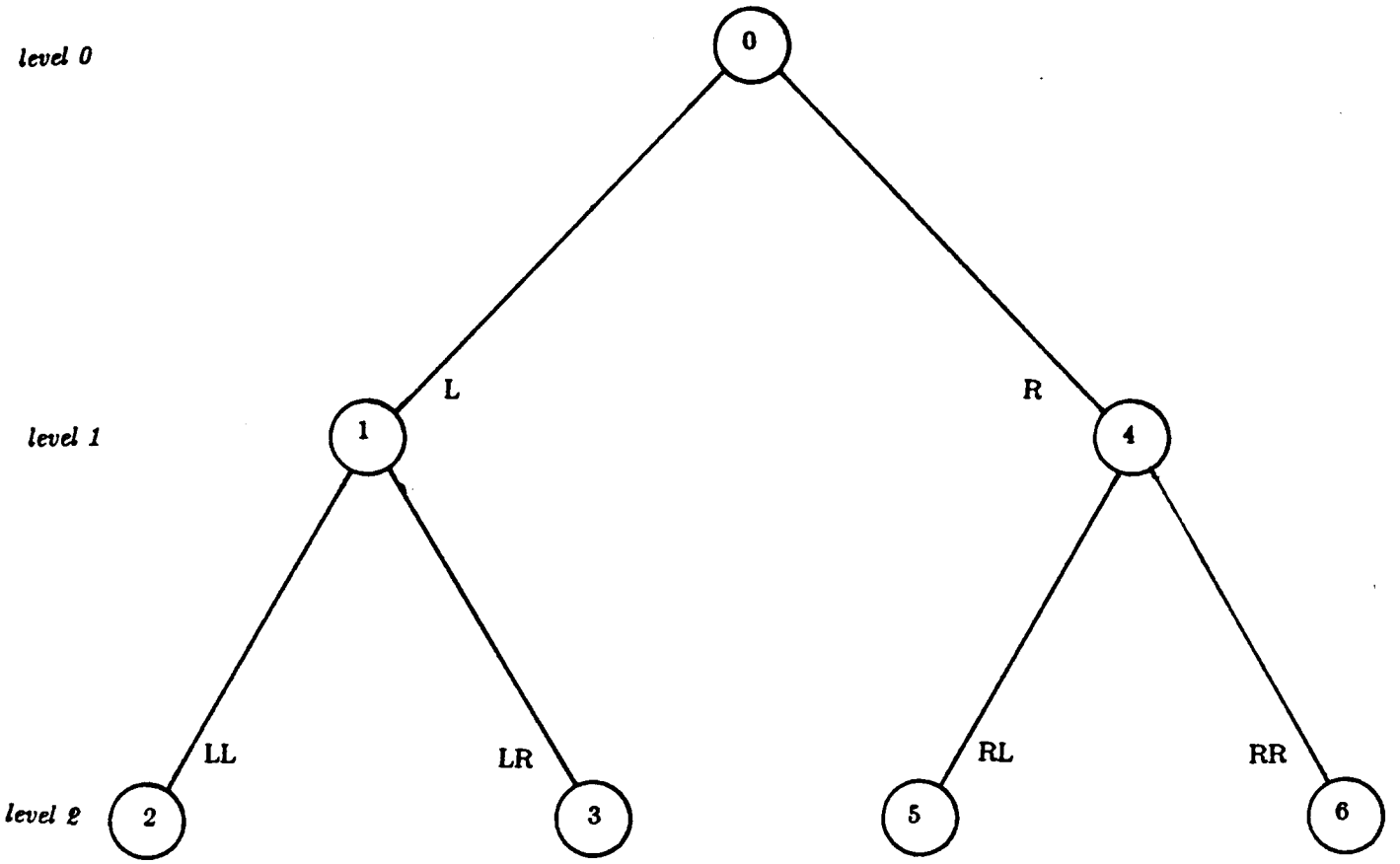


Figure 8. Subdivision tree for the example.

7. Conclusion

Midpoint subdivision was generalized to *arbitrary subdivision*; this allows subdivision to be performed at any parametric value without restricting it to be the midpoint. This enables the subdivision point in parametric space to take into account the curvature of the curve or the curve length in geometric space.

After explaining the original development of Bézier curves, the mathematical theory for arbitrary subdivision was developed, and finally an illustration of the subdivision process that shows the recursive procedure in a step-by-step manner was given.

It should be mentioned that although this paper only concerns itself with Bézier *curves*, the results could be extended for *surfaces* by treating the control vertices in each parametric direction as a control polygon for a curve, and applying the recurrence relation accordingly.

Acknowledgements

The author is indebted to Jane P. Wilhelms of the Berkeley Computer Graphics Laboratory for her assistance in the preparation of this paper, especially in the programming of the computer-generated figures. Appreciation is also expressed to Professor Pierre Bézier for his helpful comments.

References

1. Brian A. Barsky, "A Description and Evaluation of Various 3-D Models," *IEEE Computer Graphics and Applications,* Vol. 4, No. 1, January, 1984, pp. 38-52. Earlier version published in *Proceedings of InterGraphics '83,* Japan Management Association, Tokyo, 11-14 April 1983, pp. (B2-5) 1 to 21.
2. Brian A. Barsky, Tony D. DeRose, and Mark D. Dippé, "An Adaptive Subdivision Method with Crack Prevention for Rendering Beta-spline Objects." In preparation.
3. Pierre E. Bézier, *Emploi des machines à commande numérique,* 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.
4. Pierre E. Bézier, "Mathematical and Practical Possibilities of UNISURF," pp. 127-152 in *Computer Aided Geometric Design,* ed. Barnhill, Robert E. and Riesenfeld, Richard F.,Academic Press,New York(1974).
5. Pierre E. Bézier, *Essai de définition numérique des courbes et des surfaces expérimentales,* Ph.D. Thesis, l'Université Pierre et Marie Curie, Paris (February, 1977).
6. Pierre E. Bézier, private communication. July 1983.
7. Jeffrey M. Lane and Richard F. Riesenfeld, "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.
8. Richard F. Riesenfeld, Elaine Cohen, Russell D. Fish, Spencer W. Thomas, Elizabeth S. Cobb, Brian A. Barsky, Dino L. Schweitzer, and Jeffrey M. Lane, "Using the Oslo Algorithm as a Basis for CAD/CAM Geometric Modelling," pp. 345-356 in *Proceedings of the Second Annual NCGA National Conference,* National Computer Graphics Association, Inc.,Baltimore(14-18 June, 1981).