ing of the manuscript. Also, thanks are due to Mr. H. Yoshidomi for making the programming.

REFERENCES

[1] H. Freeman, "On the encoding of arbitrary geometric configurations," *IRE Trans. Electron. Comput.*, vol. EC-10, pp. 260–268, June 1961.

[2] S. Murakami, M. Okada, and T. Tsuchiya, "A study on a method of inputting line drawings," *Trans. IECE Japan*, vol. J59-D, pp. 117–124, Feb. 1976.

[3] F. de Coulon and O. Johnsen, "Adaptive block scheme for source coding of black-and-white facsimile," *Electron. Lett.*, vol. 12, pp. 61–62, Feb. 1976.

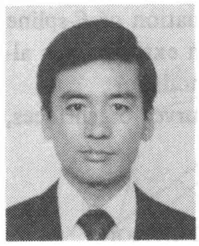
[4] O. Johnsen, "Etude de strategies adaptives pour la transmission d'images facsimile a deux niveaux," *AGEN Mittelungen*, no. 20, June 1976.

[5] T. S. Huang, "Coding of two-tone images," *IEEE Trans. Commun.*, vol. COM-25, pp. 1406–1424, Nov. 1977.

[6] H. G. Musmann and D. Preuss, "Comparison of redundancy reducing codes for facsimile transmission of documents," *IEEE Trans. Commun.*, vol. COM-25, pp. 1425–1433, Nov. 1977.

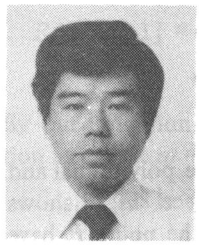
[7] K. Kubota, "Facsimile," *J. IECE Japan*, vol. 57, pp. 1370–1376, Nov. 1974.

[8] M. Takagi and T. Tsuda, "Band-width compression for facsimile using two-dimensional prediction," *Trans. IECE Japan*, vol. 56-D, pp. 170–177, Mar. 1973.



Eiji Kawaguchi was born in Japan, on November 27, 1940. He received the B.E., M.E., and D.E. degrees in 1964, 1966, and 1971, respectively, from Kyushu University, Fukuoka, Japan.

Currently he is working as an Associate Professor at the Faculty of Engineering, Kyushu University. His main research interest pertains to speech recognition systems and digital processing of pictures.



Tsutomu Endo was born in Japan, on April 30, 1949. He received the B.E. and M.E. degrees in 1972 and 1974, respectively, from Kyushu University, Fukuoka, Japan.

Currently he is working as a Research Assistant at the Faculty of Engineering, Kyushu University. His main research interest pertains to document retrieval systems and digital processing of pictures.

A Theoretical Development for the Computer Generation and Display of Piecewise Polynomial Surfaces

JEFFREY M. LANE AND RICHARD F. RIESENFELD

***Abstract*—Two algorithms for parametric piecewise polynomial evaluation and generation are described. The mathematical development of these algorithms is shown to generalize to new algorithms for obtaining curve and surface intersections and for the computer display of parametric curves and surfaces.**

***Index Terms*—*B*-splines, computer-aided geometric design, computer graphics, subdivision.**

I. INTRODUCTION

RECENTLY there has been increasing attention given to the modeling of physical objects within a digital computer for automated design and manufacture [1], [2]. Potentially, the computer can free the designer from the limitations of traditional drafting techniques while enhancing and accelerating the production process. Working interactively at a graphics terminal or a numerically controlled drafting machine, a designer can modify or manipulate an existing model or he can design an object *ab initio*, using trial and error techniques to produce an acceptable design. The computer system can then generate the necessary information for numerically controlled manufacture and production.

The realization of such a system for interactive curve and surface design imposes certain constraints on the computer-based model. The model must accurately represent a variety of shapes, be amenable to analysis and manipulation, and must take into account the capabilities and limitations of both the computer and the designer.

Two methods recently proposed for curve and surface design are vector-valued *B*-spline and Bernstein approximation [3], [4]. Although these methods have already proven successful in several design systems [5]–[7], their performance in interactive design is hampered by the relatively slow algorithms for their evaluation and manipulation. In this paper we present

Manuscript received July 12, 1978; revised January 30, 1979. The work of R. F. Riesenfeld was supported by NSF Grant MCS-78-01966.

J. M. Lane is with the Boeing Commercial Airplane Company, Seattle, WA 98124.

R. F. Riesenfeld is with the Department of Computer Science, University of Utah, Salt Lake City, UT 84112.

two fast subdivision algorithms for the evaluation of B-spline and Bernstein curves and surfaces. We then extend these algorithms in a straightforward manner to methods for:

1) finding the intersection of any two curves or surfaces, and
2) surface display.

II. Piecewise Bernstein Approximation

In this method curves are vector-valued piecewise polynomials where the curve is given on each piece by

$$B(t) = \sum_{i=0}^{m} f_{m,i}(t) P_i \qquad P_i \in R^3 \tag{2.1}$$

and

$$f_{m,i}(t) = \binom{m}{i} (t-a)^i (b-t)^{m-i} / (b-a)^m$$

for $t \in R$ (real), where m is the degree of the polynomial and the P_i are the appropriate vertices for that piece. Fig. 1 shows a piecewise Bernstein approximation where the points P_i have been associated with the vertices of a polygon, in the manner of Bézier [4].

Surface pieces are generated by taking the tensor product of the basis functions with respect to two orthogonal directions. That is,

$$B(u, v) = \sum_{i=0}^{m} \frac{\binom{m}{i}}{(b-a)^m} (u^i - a)(b-u)^{m-i} \cdot \left(\sum_{j=0}^{n} \frac{\binom{n}{j}}{(d-c)^n} (v^j - c)(d-v)^{n-j} \right) P_{ij} \tag{2.2}$$

where $(u, v) \in R^2$, m, n are the degrees of the polynomial with respect to the u and v directions and the P_{ij} are the control points. Fig. 2 shows a bicubic Bernstein polynomial, where we now associate the control points P_{ij} with the vertices of a defining rectilinear network. We will find the following properties of Bernstein approximation important in the formulation of the subdivision algorithms.

Lemma 2.1 (Convex Hull Property): The Bernstein basis functions are nonnegative on $[a, b]$ and sum identically to 1, i.e.,

$$\sum_{i=0}^{m} f_{m,i}(t) \equiv 1$$

and $f_{m,i}(t) \geq 0$ for all $t \in [a, b]$, where $i \in (0, 1, \cdots, m)$, and m is a positive integer.

Proof: The basis functions are evidently nonnegative on $[a, b]$. From the binomial expansion theorem we have

$$\sum_{i=0}^{m} \frac{\binom{m}{i}}{(b-a)^m} (t-a)^i (b-t)^{m-i} = \frac{(t-a+b-t)^m}{(b-a)^m} = 1.$$

Lemma 2.2: $f_{m,i}(t) = f_{m,m-i}(1-t)$.

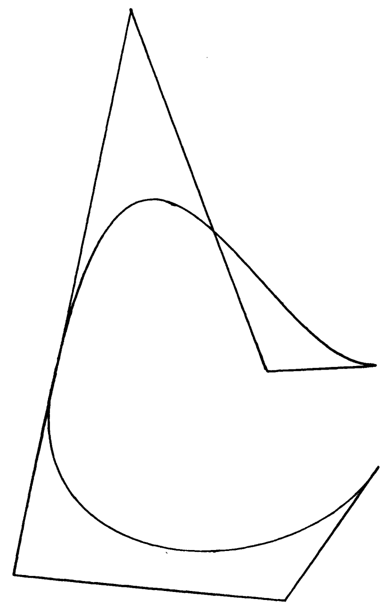


Fig. 1. Piecewise Bernstein approximation.

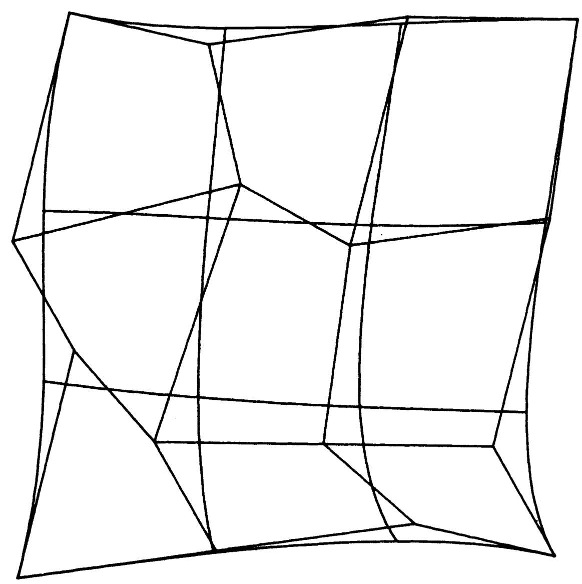


Fig. 2. Bicubic Bernstein polynomial.

Proof:

$$\begin{aligned} f_{m,i}(t) &= \binom{m}{i} t^i (1-t)^{m-i} \\ &= \binom{m}{m-i} t^i (1-t)^{m-i} \\ &= \binom{m}{m-i} (1-t)^{m-i} t^{m-(m-i)} \\ &= f_{m,m-i}(1-t). \end{aligned}$$

Lemma 2.3: Let $B_m[P; a, b] = B_m[P_0, P_1, \cdots, P_m; a, b]$ be the Bernstein polynomial of degree m to the polygon P on $[a, b]$. That is,

$$B_m[P;a,b] = \sum_{i=0}^{m} P_i \binom{m}{i} \frac{(t-a)^i(b-t)^{m-i}}{(b-a)^m},$$

$$t \in R, P \equiv [P_0, \cdots, P_m].$$

Then

$$\begin{aligned}B_m&[P_0,\cdots,P_m;a,b]\\ &= ((b-t)B_{m-1}[P_0,\cdots,P_{m-1};a,b]\\ &\quad + (t-a)B_{m-1}[P_1,\cdots,P_m;a,b])1/(b-a). \qquad (2.3)\end{aligned}$$

It suffices to consider the normalized interval [0, 1].

Proof: Simple algebraic manipulation yields

$$\begin{aligned}B_m[P;0,1] &= \sum_{i=0}^{m}\binom{m}{i} t^i(1-t)^{m-i}P_i\\ &= \sum_{i=0}^{m-1}\binom{m}{i}\frac{m-i}{m} t^i(1-t)^{m-i}P_i\\ &\quad + \sum_{i=1}^{m}\binom{m}{i}\frac{i}{m} t^i(1-t)^{m-i}P_i\\ &= \sum_{i=0}^{m-1}\binom{m-1}{i} t^i(1-t)^{m-i}P_i\\ &\quad + \sum_{i=0}^{m-1}\binom{m-1}{i} t^{i+1}(1-t)^{m-1-i}P_{i+1}\\ &= (1-t)[B_{m-1}P_0,\cdots,P_{m-1};0,1]\\ &\quad + t[B_{m-1}P_1,\cdots,P_m;0,1].\end{aligned}$$

Theorem 2.1 (Subdivision Theorem): Let $B_m[P;a,b]$ be defined as above. Then

$$B_m[P;0,1] = B_m[P_0^0,\cdots,P_m^m;0,\tfrac{1}{2}] \qquad (2.4)$$

and

$$B_m[P;0,1] = B_m[P_m^m, P_m^{m-1},\cdots,P_m^0;\tfrac{1}{2},1] \qquad (2.5)$$

where

$$P_i^k = \begin{cases}(P_{i-1}^{k-1} + P_i^{k-1})/2 & k = 1,\cdots,m\\ P_i & k = 0.\end{cases} \qquad (2.6)$$

It suffices to prove the first equality (2.4), since Lemma 2.2 implies that the symmetric relationship (2.5) must hold. (See Fig. 3.)

Proof [8] (By induction on m): Let $m = 1$. Then we have

$$\begin{aligned}B_1[P;0,1] &= (1-t)P_0 + tP_1\\ &= (1-2t)P_0 + t(P_0+P_1)\\ &= 2\left(\frac{1}{2} - t\right)P_0 + 2t\left(\frac{P_0+P_1}{2}\right)\\ &= B_1\left[P_0^0, P_1^1; 0, \frac{1}{2}\right].\end{aligned}$$

Now assume the theorem holds for all $k < m$. We have by Lemma 2.3 that

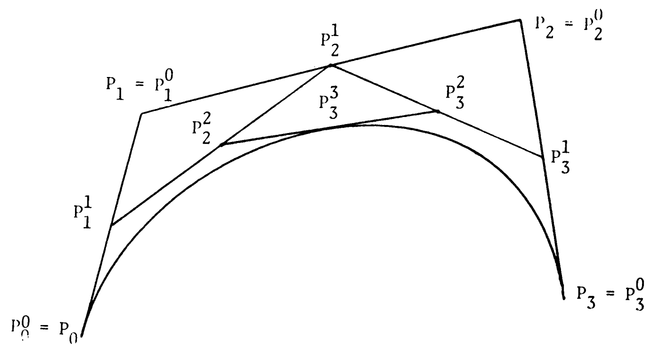


Fig. 3. Subdivision of cubic Bernstein polynomial.

$$\begin{aligned}B_m[P;0,1] &= (1-t)B_{m-1}[P_0,\cdots,P_{m-1};0,1]\\ &\quad + tB_{m-1}[P_1,\cdots,P_m;0,1]. \qquad (2.7)\end{aligned}$$

By our induction assumption, and similar algebraic manipulation as in case $m = 1$, (2.7) reduces to

$$\begin{aligned}B_m&[P;0,1]\\ &= 2(\tfrac{1}{2} - t)B_{m-1}[P_0^0,\cdots,P_{m-1}^{m-1};0,\tfrac{1}{2}]\\ &\quad + 2t(B_{m-1}[P_0^0,\cdots,P_{m-1}^{m-1};0,\tfrac{1}{2}]\\ &\quad + B_{m-1}[P_1^0,\cdots,P_m^{m-1};0,\tfrac{1}{2}])/2.\end{aligned}$$

Since the Bernstein operator is linear, we have

$$\begin{aligned}B_m&[P;0,1]\\ &= 2(\tfrac{1}{2} - t)B_{m-1}[P_0^0,\cdots,P_{m-1}^{m-1};0,\tfrac{1}{2}]\\ &\quad + 2tB_{m-1}[(P_0^0+P_1^0)/2,\cdots,(P_{m-1}^{m-1}+P_m^{m-1})/2;0,\tfrac{1}{2}]\\ &= 2(\tfrac{1}{2} - t)B_{m-1}[P_0^0,\cdots,P_{m-1}^{m-1};0,\tfrac{1}{2}]\\ &\quad + 2tB_{m-1}[P_1^1,\cdots,P_m^m;0,\tfrac{1}{2}].\end{aligned}$$

But we can apply Lemma 2.3 to yield

$$B_m[P;0,1] = B_m[P_0^0,\cdots,P_m^m;0,\tfrac{1}{2}].$$

Lemma 2.4: Let P_i^i be defined as in (2.6). Then

$$|P_i^i - P_{i+1}^{i+1}| \leq (\tfrac{1}{2})M$$

where $i = 0, 1, \cdots, m-1$ and $M = \max|P_{i+1} - P_i|$, and $|\ |$ means Euclidean distance.

Proof: The conclusion is immediate from these easily deduced relationships (see Fig. 3):

$$|P_i^i - P_{i+1}^{i+1}| = \tfrac{1}{2}|P_i^i - P_{i+1}^i|, \qquad i = 0,\cdots,m, m \geq 1 \qquad (2.8)$$

and

$$|P_i^i - P_{i+1}^i| \leq \tfrac{1}{2}(|P_i^{i-1} - P_{i+1}^{i-1}| + |P_{i+1}^{i-1} - P_i^{i-1}|)$$

$$i = 0,\cdots,m, m \geq 2.$$

Again, in view of Lemma 2.2, a symmetric condition holds on the polygon $[P_m^{m-1}, P_m^{m-2},\cdots,P_m^0]$. If we apply this splitting construction (2.5) in turn to each of the polygons $[P_0^0,\cdots,P_m^m]$ and $[P_m^m, P_m^{m-1},\cdots,P_m^0]$, we generate four polygons, which when concatenated form a polygon ψ_m^2 of $4m+1$ vertices. Defining $\psi_m^k[P]$ as the polygon derived after k iterations of this algorithm (ψ_m^k would have $2^k m + 1$ vertices), we shall prove

$$\lim_{k\to\infty} \psi_m^k[P] = B_m[P].$$

Lemma 2.5: Define $\psi_m^k[P]$ as above. Then for any two consecutive vertices Q and R of ψ_m^k we have

$$|R - Q| \leqslant (\tfrac{1}{2})^k M$$

where

$$M = \max |P_{i+1} - P_i|, \quad i = 0, \cdots, m-1.$$

Proof: It suffices to consider $\psi^k[P]$ on interval $[0, (\frac{1}{2})^k]$, since Lemma 2.2 and Theorem 2.1 imply that a symmetric relationship holds on any interval $[c, c + (\frac{1}{2})^k]$, $0 \leqslant c \leqslant 1 - (\frac{1}{2})^k$. $\psi^k[P]$ can be conveniently represented recursively on $[0, (\frac{1}{2})^k]$ with the following construction:

$$\psi[P] = [P_0^0, P_1^1, \cdots, P_m^{m-1}]$$

on $[0, \frac{1}{2}]$, and

$$\psi^k[P] = \psi[\psi^{k-1}[P]] \equiv [P_0^{0,k}, P_1^{1,k}, \cdots, P_m^{m,k}]$$

on $[0, (\frac{1}{2})^k]$, where P_i^i is defined as in (2.5). Thus, we must show that

$$|P_i^{i,k} - P_{i+1}^{i+1,k}| \leqslant (\tfrac{1}{2})^k M \tag{2.9}$$

for all $k > 0$, $i = 0, \cdots, m-1$. The result is trivial for the case $m = 1$. For the general case $m \geqslant 2$ a proof will follow by induction on k.

For $k = 1$ the conclusion is provided by Lemma 2.4. So assume the lemma holds for all $j < k$. We must prove that it holds for k. Now from (2.8) and our assumption we have

$$\begin{aligned} |P_i^{i,k} - P_{i+1}^{i+1,k}| &= \tfrac{1}{2} |P_i^{i,k} - P_{i+1}^{i,k}| \\ &\leqslant \tfrac{1}{4} (|P_i^{i-1,k} - P_{i-1}^{i-1,k}| + |P_{i+1}^{i-1,k} - P_i^{i-1,k}|) \\ &\leqslant \tfrac{1}{4} ((\tfrac{1}{2})^{k-1} M + (\tfrac{1}{2})^{k-1} M) = (\tfrac{1}{2})^k M. \end{aligned}$$

Theorem 2.2: Let $B_m[P; 0, 1]$ be the vector-valued Bernstein approximation of degree m to the polygon $P = [P_0, \cdots, P_m]$, and define the polygon $\psi^k[P]$ as in Lemma 2.5. Then

$$\lim_{k\to\infty} \psi_m^k[P] = B_m[P; 0, 1].$$

Proof: From Theorem 2.1 and simple induction on k we have

$$B_m[\psi_m^k[P]; 0, 1] = B_m[P; 0, 1]. \tag{2.10}$$

From the convex hull property (Lemma 2.1), we then have $B_m[P; 0, 1]$ restricted to $[c, c + (\frac{1}{2})^k]$, $0 \leqslant c \leqslant 1 - (\frac{1}{2})^k$, lies within the convex hull of some $2(m+1)$ vertices of ψ_m^k. But in Lemma 2.5 we have shown that the length of the convex hull must tend to zero as k increases; thus, the sequence $\psi_m^k[P]$ must converge to the curve $B_m[P; 0, 1]$.

III. Uniform *B*-Spline Approximation

With this method curves are represented by piecewise polynomials defined by

$$P(t) = \sum_{i=0}^{n} P_i N_{i,m}(t)$$

where $N_{i,m}(t)$ is the B-spline basis function of degree $m - 1$ defined by

$$\begin{aligned} N_{i,m}(t) \equiv N_{i,m}(h, t) &= \frac{1}{h} \int_{t-h/2}^{t+h/2} N_{i,m-1}(h, x)\, dx \\ &= \pi(h, t) * N_{i,m-1}(h, t) \end{aligned} \tag{3.1}$$

where $*$ represents convolution, h is the mesh size, $\pi(h, t) = N_{0,1}(h, t)/h$

$$N_{i,1}(h, t) \equiv \begin{cases} 1 & -h/2 \leqslant t - hi \leqslant h/2 \\ 0 & \text{elsewhere.} \end{cases} \tag{3.2}$$

We can readily prove from the definition (3.1) and induction that

$$\sum_i N_{i,m}(t) = 1, \quad N_{i,m} \geqslant 0.$$

Thus, the B-spline basis also possesses the convex hull property. Fig. 4 shows a cubic B-spline approximation while Fig. 5 gives the analogous bicubic surface approximation. The following theorems are analogous to those developed for piecewise Bernstein approximation.

Theorem 3.1: Let $B_m[P; 0, n]$ be the uniform B-spline approximation of order $m \geqslant 2$ and mesh size 1 to the polygon $P = [P_0, \cdots, P_n]$ on $[0, n]$, $n \geqslant m$. That is,

$$B_m[P; 0, n] = \sum_{i=0}^{n} P_i N_{i,m}(1, x) \tag{3.3}$$

where $N_{i,m}(1, x)$ is given by (3.2) and $x \in [(m-2)/2, n - (m-2)/2]$. Then

$$B_m[P; 0, n] = \sum_{i=0}^{2n-m+2} P_i^m N_{i,m}(0.5, x) \tag{3.4}$$

where P_i^m is defined recursively by

$$P_i^m = (P_i^{m-1} + P_{i+1}^{m-1})/2 \qquad i = 0, 1, \cdots, 2n + m + 2, m > 2, \tag{3.5}$$

and

$$P_i^2 = \begin{cases} P_{i/2} & i \text{ even} \\ (P_{(i-1)/2} + P_{(i+1)/2})/2 & i \text{ odd} \end{cases} \qquad i = 0, 1, \cdots, 2n.$$

That is, given the B-spline approximation of order m with integral knot spacing to the polygon P, the control points for the same curve in terms of the B-spline basis over the refined mesh $0.0, 0.5, \cdots, n - 0.5, n$ are given by (3.5). (See Fig. 6.)

Proof (By induction on degree): Let $m = 2$. We must show that

$$\sum_{i=0}^{n} P_i N_{i,2}(1, x) = \sum_{i=0}^{2n} P_i^2 N_{i,2}(0.5, x).$$

But the degree 1 (order 2) B-spline approximation is the piecewise linear interpolant to the vertices P_i. Thus, (3.5) holds, since piecewise linear interpolation preserves linear polynomials.

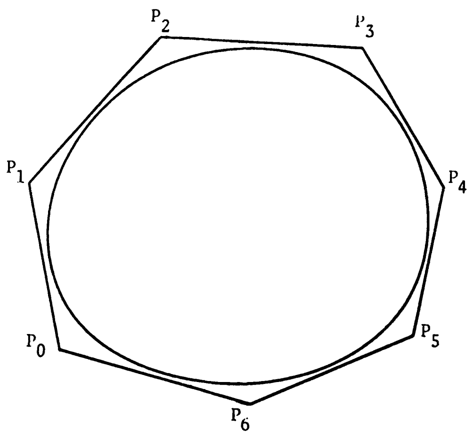


Fig. 4. Cubic B-spline approximation.

Now assume (3.5) holds for all k, $2 \leqslant k < m$. For $k = m$ we have

$$\sum_{i=0}^{n} P_i N_{i,m}(1,x) = \sum_{i=0}^{n} P_i(\pi(1,x) * N_{i,m-1}(1,x))$$

$$= \pi(1,x) * \sum_{i=0}^{n} P_i N_{i,m-1}(1,x) \tag{3.7}$$

for $x \in [(m-2)/2,\ n-(m-2)/2]$. Now by our induction hypothesis

$$\sum_{i=0}^{n} P_i N_{i,m-1}(1,x) = \sum_{i=0}^{2n-m+3} P_i^{m-1} N_{i,m-1}(0.5,x);$$

thus, (3.6) reduces algebraically to

$$\sum_{i=0}^{n} P_i N_{i,m}(1,x) = \pi(1,x) * \sum_{i=0}^{2n-m+3} P_i^{m-1} N_{i,m-1}(0.5,x)$$

$$= (\pi(0.5, x-0.25) + \pi(0.5, x+0.25))/2$$

$$* \sum_{i=0}^{2n-m+3} P_i^{m-1} N_{i,m-1}(0.5,x)$$

$$= \left(\frac{1}{2}\right) \sum_{i=0}^{2n-m+3} P_i^{m-1} \pi(0.5, x-0.25)$$

$$* N_{i,m-1}(0.5,x) + \left(\frac{1}{2}\right) \sum_{i=0}^{2n-m+3} P_i^{m-1}$$

$$\cdot \pi(0.5, x-0.25) * N_{i,m-1}(0.5,x)$$

$$= \left(\frac{1}{2}\right)\left(\sum_{i=0}^{2n-m+3} P_i^{m-1} N_{i,m}(0.5, x-0.25)\right.$$

$$\left. + \sum_{i=0}^{2n-m+3} P_i^{m-1} N_{i,m}(0.5, x+0.25)\right) \tag{3.8}$$

for $x \in [(m-2)/2,\ n-(m-2)/2]$. Now both $N_{0,m}(0.5, x-0.25)$ and $N_{2n-m+3,m}(0.5, x+0.25)$ are zero on the interval $[(m-2)/2,\ n-(m-2)/2]$, and after dropping terms concerning these basis functions and rearranging the remaining terms in (3.8) we have

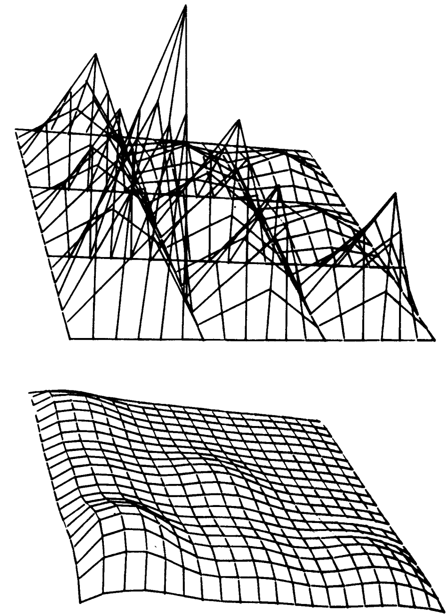


Fig. 5. Bicubic B-spline surface and associated rectilinear network of points.

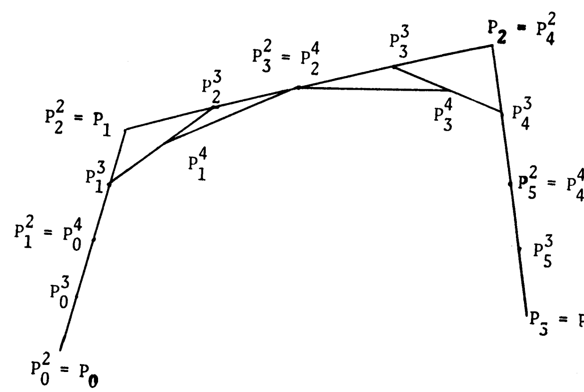


Fig. 6. Subdivision of cubic B-spline curve segment.

$$\sum_{i=0}^{n} P_i N_{i,m}(1,x) = \sum_{i=0}^{2n-m+2} ((P_i^{m-1} + P_i^{m-1})/2) N_{i,m}(0.5,x)$$

$$= \sum_{i=0}^{2n-m+2} P_i^m N_{i,m}(0.5,x)$$

which proves the theorem.

Theorem 3.2: Let $B_m[P;0,n] = \sum_{i=0}^{n} P_i N_{i,m}(1,x)$ as in (3.8) and define

$$\psi^0[P] = [P_0^m, P_1^m, \cdots, P_{2n-m+2}^m]$$

where P_i^m is given by (3.5) and

$$\psi^k[P] = \psi[\psi^{k-1}[P]] \equiv [P_0^{m,k}, \cdots, P_{r(k,m)}^{m,k}] \tag{3.9}$$

for $r(k,m) = 2^{k+1}(n-m+2) + m - 2$, $k \geqslant 1$. Then

$$\lim_{k\to\infty} \psi^k[P] = B_m[P;0,n].$$

Riesenfeld [12] has recently given a proof of Theorem 3.2 for the case $m = 3$. Here we use a quite different approach for the proof for arbitrary m.

Proof: By Theorem 3.1 and induction on k we have

$$B_m[\psi^k[P]; 0, n] = B_m[P; 0, n].$$

Now let $M = \max |P_{i+1} - P_i|$. It is easily seen that

$$|P^m_{i+1} - P^m_i| \leq \tfrac{1}{2}M$$

and therefore

$$|P^{m,k}_{i+1} - P^{m,k}_i| \leq (\tfrac{1}{2})^{k+1}M. \tag{3.10}$$

That is,

$$\lim_{k\to\infty} |P^{m,k}_{i+1} - P^{m,k}_i| = 0.$$

But then

$$\lim_{k\to\infty} |P^{m,k}_{i+j} - P^{m,k}_i| = 0 \tag{3.11}$$

for arbitrary $i \in \{1, \cdots, r(k,m)\}$, $j = 1, \cdots, m$, $i + j \leq r(k,m)$.

From the convex hull property and Theorem 3.2 we know $B_m[P; 0, n](x_0)$, $x_0 \in ((m-2)/2,\ n - (m-2)/2)$ lies within the convex hull of $P^{m,k}_i, P^{m,k}_{i+1}, \cdots, P^{m,k}_{i+m}$ for some i. But with (3.10) we can then conclude

$$\lim_{k\to\infty} \psi^k[P] = B_m[P; 0, n].$$

IV. The Subdivision Algorithms

The subdivision algorithms for polynomial and spline evaluation are based directly on Theorems 2.2 and 3.2. Since the polygons constructed in (2.9) and (3.9) converge to their respective curves, they form a natural means of approximation to the curves. Further, the methods are stable, involving at each level of recursion nothing but addition and 1-bit shifts. Given that the curve coefficients are n-bit integers, at most n interations of the algorithm would be necessary for convergence. Convergence will in fact be much faster, as (2.9) and (3.10) indicate. The algorithms for generation of an entire curve or surface can be recursive and can be either implemented with the use of a buffer or through a stack mechanism. Here we give algorithms for the Bernstein polynomials only, the B-spline algorithms being very similar.

Algorithm I (Curve Splitting)

Given the polynomial coefficients $P \equiv [P_0, P_1, \cdots, P_n]$ in terms of the Bernstein basis on $[0, 1]$ compute $Q \equiv [Q_0, Q_1, \cdots, Q_n]$ and $R \equiv [R_0, R_1, \cdots, R_n]$ such that

$$B(s) = \sum_{i=0}^{n} \binom{n}{i} s^i(1-s)^{n-i}P_i = \sum_{i=0}^{n} \binom{n}{i} 2^n s^i \left(\frac{1}{2} - s\right)^{n-i}$$

$$Q_i = \sum_{i=0}^{n} \binom{n}{i} 2^n \left(s - \frac{1}{2}\right)^i (1-s)^{n-i} R_i.$$

PROCEDURE CURVESPLIT (P, Q, R, n)

Step 1. [Initialize] $Q \leftarrow P$. $R_n \leftarrow Q_n$.

Step 2. [Compute coefficients in double loop].

```
For j = 1 to n
begin
QTMP2 ← Q_{j-1}
for k = j to n
   begin
   QTMP1 ← QTMP2
   QTMP2 ← (Q_{k-1} + Q_k)/2
   Q_{k-1}  ← QTMP1
   end
Q_n ← QTMP2
R_{n-j} ← QTMP2
end.
```

Algorithm I can be used to develop the surface splitting routine, since the surfaces are the tensor products of the curve methods.

Algorithm II (Surface Splitting)

Given the polynomial coefficients $P \equiv [P_{ij}]$, $i = 0, 1, \cdots, m$; $j = 0, 1, \cdots, n$ in terms of the Bernstein basis on $[0, 1] \times [0, 1]$ compute

$$Q \equiv [Q_{ij}], R \equiv [R_{ij}], S \equiv [S_{ij}], \text{ and } T \equiv [T_{ij}],$$
$$i = 0, 1, \cdots, m; j = 0, 1, \cdots, n$$

such that

$$\begin{aligned}
B(u,v) &= \sum_{j=0}^{n} \binom{n}{j} v^j(1-v)^{n-j} \sum_{i=0}^{m} \binom{m}{i} u^i(1-u)^{m-i}P_{ij} \\
&= \sum_{j=0}^{n} \binom{n}{j} 2^n \left(v - \frac{1}{2}\right)^j (1-v)^{n-j} \sum_{i=0}^{m} \binom{m}{i} \\
&\quad \cdot 2^n \left(u - \frac{1}{2}\right)^i (1-u)^{m-i} Q_{ij} \\
&= \sum_{j=0}^{n} \binom{n}{j} 2^n \left(v - \frac{1}{2}\right)^j (1-v)^{n-j} \sum_{i=0}^{m} \binom{m}{i} \\
&\quad \cdot 2^n u^i \left(\frac{1}{2} - u\right)^{m-i} R_{ij} \\
&= \sum_{j=0}^{n} \binom{n}{j} 2^n v^j \left(\frac{1}{2} - v\right)^{n-j} \sum_{i=0}^{m} \binom{m}{i} \\
&\quad \cdot 2^n \left(u - \frac{1}{2}\right)^i (1-u)^{m-i} S_{ij} \\
&= \sum_{j=0}^{n} \binom{n}{j} 2^n v^j \left(\frac{1}{2} - v\right)^{n-j} \sum_{i=0}^{m} \binom{m}{i} \\
&\quad \cdot 2^n u^i \left(\frac{1}{2} - u\right)^{m-i} T_{ij}.
\end{aligned}$$

PROCEDURE SURFACE SPLIT (P, Q, R, S, T, m, n)

Step 1. [Initialize] Set $Q \leftarrow P$.

Step 2. [Split in u direction]
For $k = 0$ to n
begin
$R_{m,k} \leftarrow Q_{m,k}$
for $p = 1$ to m
begin
$QTMP2 \leftarrow Q_{p-1,k}$
for $q = p$ to m
begin
$QTMP1 \leftarrow QTMP2$
$QTMP2 \leftarrow (Q_{q-1,k} + Q_{q,k})/2$
$Q_{q-1,k} \leftarrow QTMP1$
end
$Q_{m,k} \leftarrow QTMP2$
$R_{m-j,k} \leftarrow QTMP2$
end
end

Step 3. [Split Q in v direction].
For $k = 0$ to m
begin
$S_{k,n} \leftarrow Q_{k,n}$
for $p = 1$ to n
begin
$QTMP2 \leftarrow Q_{k,p-1}$
for $q = p$ to n
begin
$QTMP1 \leftarrow QTMP2$
$QTMP2 \leftarrow (Q_{k,q-1} + Q_{k,q})/2$
$Q_{k,q-1} \leftarrow QTMP1$
end
$S_{k,n-j} \leftarrow QTMP2$
end
end.

Step 4. [Split R in v direction].
For $k = 0$ to m
begin
$T_{k,n} \leftarrow R_{k,n}$
for $p = 1$ to n
begin
$RTMP2 \leftarrow R_{k,p-1}$
for $q = p$ to n
begin
$RTMP1 \leftarrow RTMP2$
$RTMP2 \leftarrow (R_{k,q-1} + R_{k,q})/2$
$R_{k,q-1} \leftarrow RTMP1$
end
$T_{k,n-j} \leftarrow RTMP2$
end
end
Return.

Algorithm III (Polynomial Evaluation)

Given the polynomial coefficients $P_0, \cdots, P_n$ in terms of the Bernstein basis and $s \in [0, 1]$ compute

$$P(s) = \sum_{i=0}^{n} \binom{n}{i} s^i (1-s)^{n-i} P_i.$$

Step 1. [Get next bit] Set SBIT $\leftarrow \lfloor 2s \rfloor$
$s \leftarrow (2s) -$ SBIT.

Step 2. [Subdivide] If SBIT $= 0$ set $P_i \leftarrow P_i^i, i = 0, \cdots, n$
else set $P_i \leftarrow P_n^{n-i}, i = 0, \cdots, n$, where

$$P_i^k \leftarrow \begin{cases} (P_{i-1}^{k-1} + P_i^{k-1})/2, & k = 1, \cdots, n \\ P_i & k = 0. \end{cases}$$

Step 3. [Convergence Test] If max $|P_i - P_{i+1}| <$ Tolerance for all i then output P_0 and halt, else go to Step 1.

Algorithm IV (Curve Generation)

Let $P \equiv [P_0, \cdots, P_n]$ be the Bernstein coefficients for the curve. This algorithm draws the entire curve, making use of an auxiliary stack A.

Step 1. [Initialize] Set stack A empty and move to P_0.

Step 2. [Convergence Test] If $|P_i, P_{i+1}| <$ Tolerance for $i = 0, \cdots, n$ go to Step 4.

Step 3. [Push P] Push P onto Stack A. Set $P_i \leftarrow P_i^i, i = 0, 1. \cdots, n$ where

$$P_i^k \leftarrow \begin{cases} (P_{i-1}^{k-1} + P_i^{k-1})/2, & k = 1, 2, \cdots, n \\ P_i & k = 0. \end{cases}$$

Go to Step 2.

Step 4. [Pop P] Draw to P_n. If stack A is empty, the algorithm terminates; set $P \leftarrow A$. Set $P_i \leftarrow P_n^{n-i}, i = 0, 1, \cdots, n$, where P_n^{n-i} is defined as in Step 3 and go to Step 2.

Theorem 2.2 guarantees the convergence of both algorithms. The formulation of the parameterization of the curve in terms of the left-right splitting action is an extension of an earlier formulation of a similar curve method described by Riesenfeld [12]. It is noteworthy that Algorithm II represents the traversal of a binary tree in postorder. The convergence test given in both algorithms checks for "constant" form, that is, it tests to see if all points of the segment are within a tolerance of being equal. However, this test can be avoided, since (2.9) gives us the capability to determine *a priori* an iteration level at which convergence must take place. Thus, as an alternative to subdivision, we could also determine *a priori* a parameter step which can be used in forming a difference table for forward difference generation of the curve [18]. A further generalization would be not to test for convergence to "constant" form in Algorithm II, but to test for the amount of deviation from a straight line, therefore computing a piecewise linear approximation to the curve to within any tolerance desired. In this case Step 2 in Algorithm II would be as follows.

> Step 2. [Convergence Test] Let $l(P_0, P_n)$ be the line segment from P_0 to P_n and let $d(\cdot, -)$ be the Euclidean distance. Then if $d(P_i, l(P_0, P_n)) <$ Tolerance for all $i = 2, \cdots, n - 1$ go to Step 4.

This convergence criteria was used in making the curves in the Figs. 7-10.

The curve generation algorithm extends directly to a surface generation algorithm. For surfaces instead of outputting line segments we output quadrilaterals, where the quadrilaterals are fit arbitrarily close to the surface. The quadrilaterals outputted

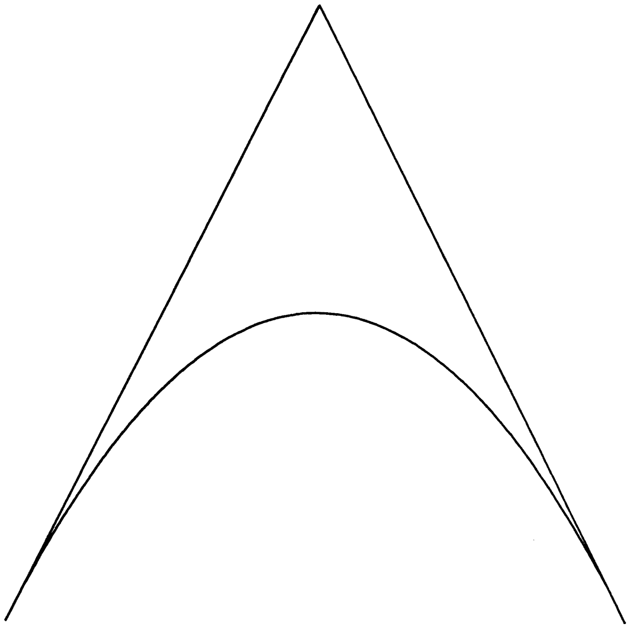


Fig. 7. Quadratic polynomial.

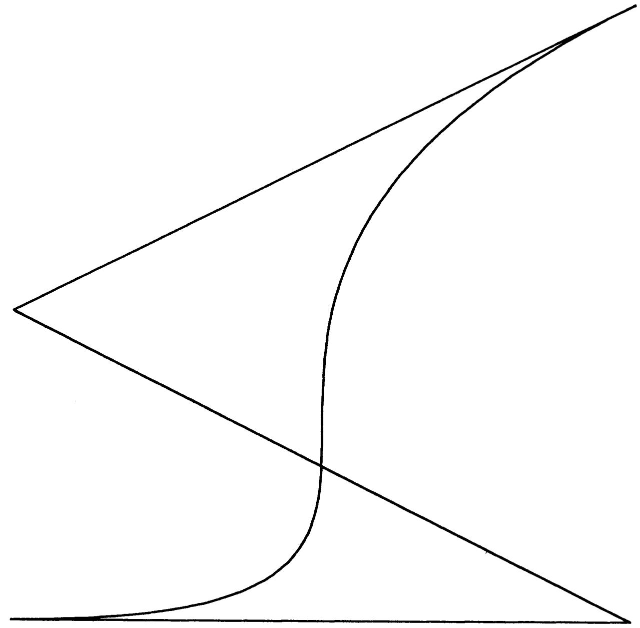


Fig. 8. Cubic polynomial.

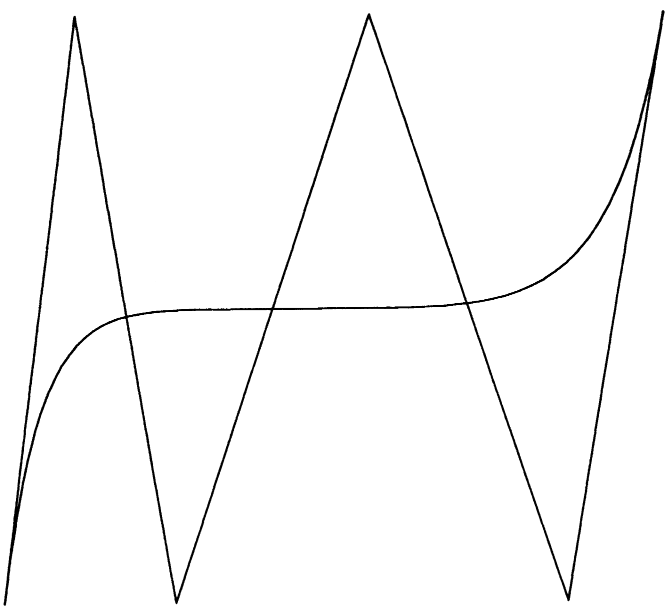


Fig. 9. Quintic polynomial.

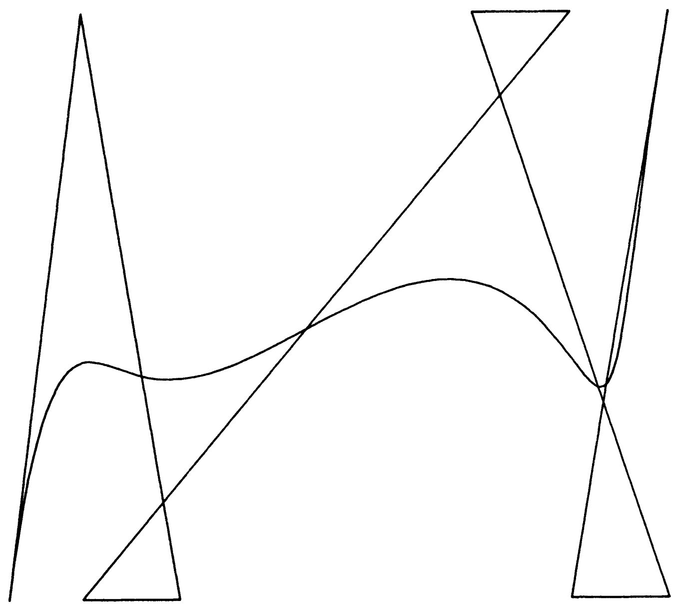


Fig. 10. Eighth-degree polynomial.

can go to a line or surface display algorithm. Below we give the recursive form of this algorithm.

Algorithm V (Surface Generation)

Given $P \equiv [P_{ij}]$, $i = 0, 1, \cdots, m$, $j = 0, 1, \cdots, n$ the Bernstein coefficients on $[0, 1] \times [0, 1]$ for the surface, this algorithm draws the entire surface.

PROCEDURE SURFGEN (P, m, n)
Step 1. [Convergence Test] Let $l(P_{0,0}, P_{m,0}, P_{0,n})$ be the plane through $P_{0,0}$, $P_{m,0}$, and $P_{0,n}$ and for any two points A and B let $l(A, B)$ be the line segment from A to B and $d(A, B)$ be the Euclidean distance. Then if

$$d(l(P_{0,0}, P_{m,0}, P_{0,n}), P_{ij}) < \text{Tolerance}$$

$$d(l(P_{0,0}, P_{m,0}), P_{i,0}) < \text{Tolerance}$$

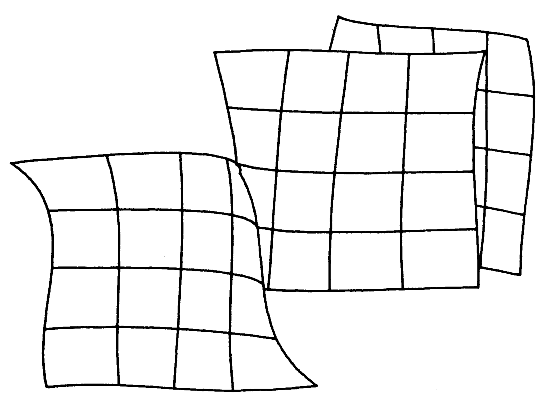


Fig. 11. Three bicubic patches represented by curves of constant parameter, hidden lines removed.

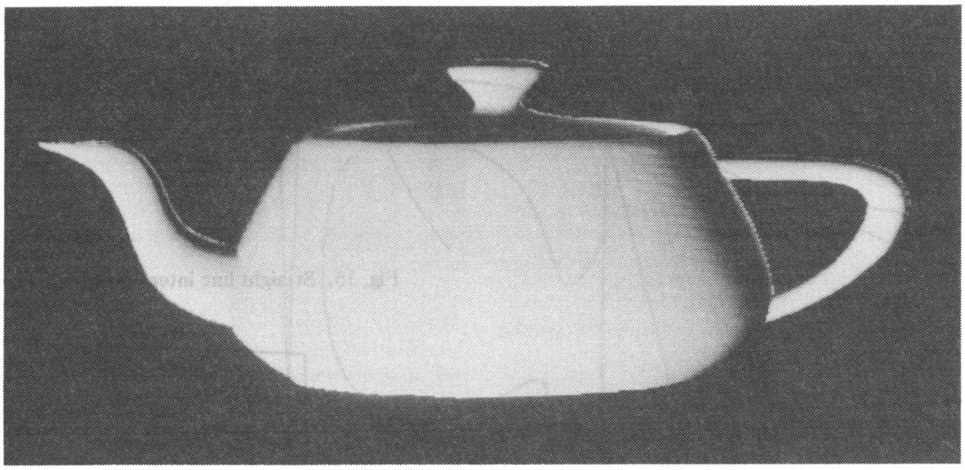


Fig. 12. A teapot modeled with 28 bicubic patches, hidden surfaces removed.

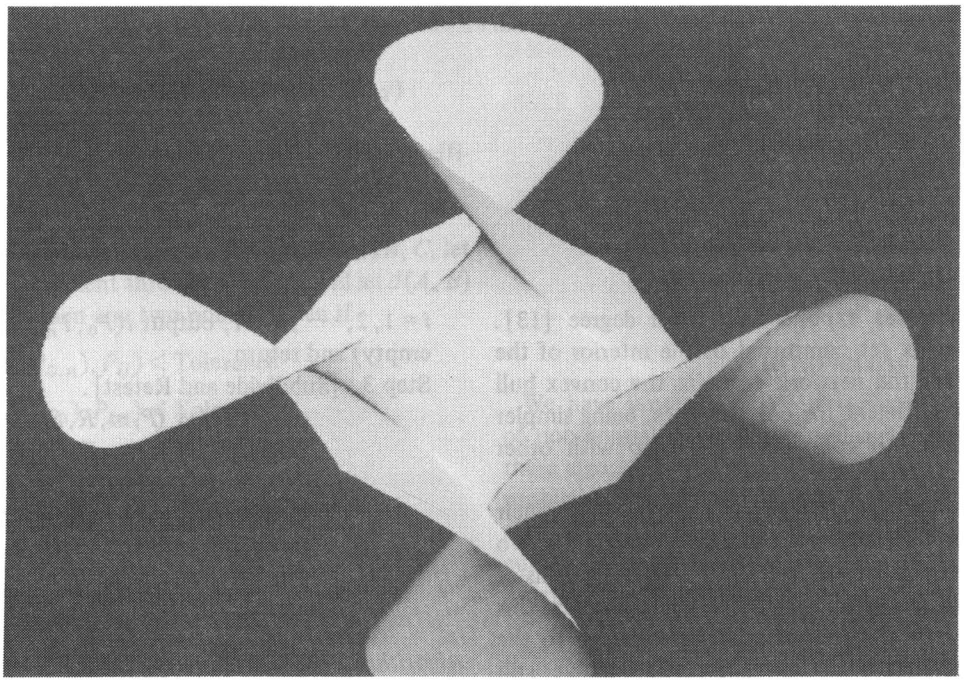


Fig. 13. One pathological bicubic patch, hidden surfaces removed.

$$d(l(P_{0,n}, P_{m,n}), P_{i,n}) < \text{Tolerance}$$

$$d(l(P_{0,0}, P_{0,n}), P_{0,j}) < \text{Tolerance}$$

$$d(l(P_{m,0}, P_{m,n}), P_{mj}) < \text{Tolerance}$$

$i = 0, 1, \cdots, m$; $j = 0, 1, \cdots, n$ then output quadrilateral $(P_{0,0}, P_{m,0}, P_{m,n}, P_{0,n})$ and return.

Step 2. [Subdivide and retest].

SURFACESPLIT (P, Q, R, S, T, m, n)
SURFGEN (Q, m, n)
SURFGEN (R, m, n)
SURFGEN (S, m, n)
SURFGEN (T, m, n)
return.

The quadrilateral polygons were passed to a line drawing algorithm in making Fig. 11 (3 bicubic patches) a z-buffer algorithm for hidden surface display in Fig. 12 (28 bicubic patches), and a scan line algorithm for polygon display [17] in Fig. 13 (1 bicubic patch).

V. The Intersection Algorithms

Of particular importance in CAD/CAM applications of curve and surface representations is the ability to interrogate the models for arbitrary intersections among them. Below we give algorithms for finding all the intersections of two arbitrary polynomial or spline curves or surfaces. As with the subdivision algorithms, the extensions to surfaces is straightforward.

We know that each curve segment lies within the convex hull of its defining polygon. So if we can prove that the convex hulls do not intersect, then we can conclude that the curves cannot intersect. The computation of the convex hull for polynomial networks is not difficult, but the computational complexity of the problem for three-dimensional and higher

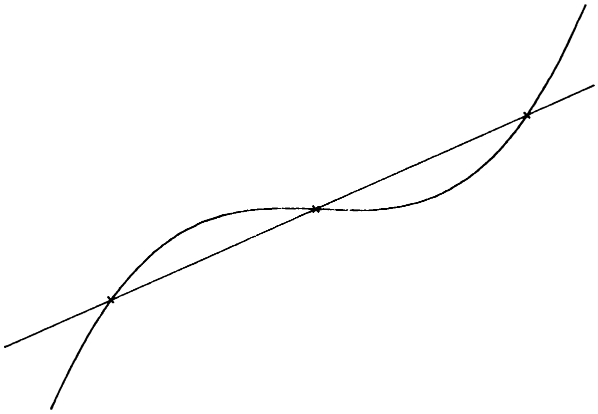


Fig. 14. Intersection of a straight line and a cubic.

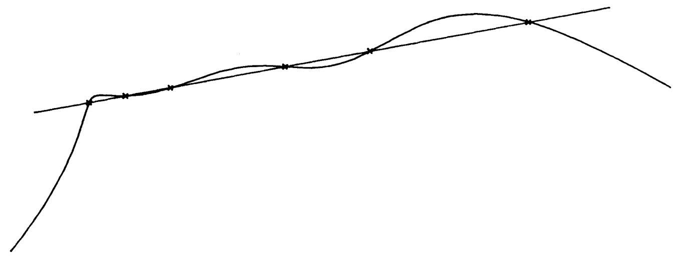


Fig. 15. Straight line intersected with degree 6 polynomial.

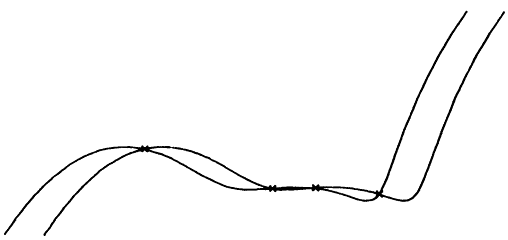


Fig. 16. Two intersecting seventh degree polynomials.

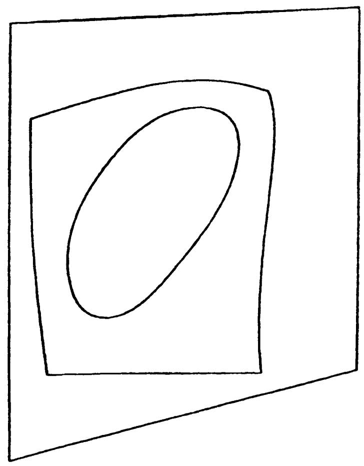


Fig. 17. A bilinear surface intersected with a biquintic.

degree networks increases exponentially with degree [13]. Fortunately, the convex set comprised of the interior of the minmax box [14] for the network contains the convex hull of the points and is sufficient for our purposes, being simpler to compute and easier to compare for overlap with other boxes.

If we cannot prove that the curves do not intersect, i.e., their minmax boxes do intersect, then we simply subdivide the two curves and reapply the test to all appropriate combinations of these two segments. Theorems 2.2 and 3.2 guarantee that this process converges since the minmax boxes converge to the curves. When two minmax boxes are sufficiently small yet still overlap we declare that a point of intersection has been found. Below we give recursive versions for this algorithm for finding a point of intersection between two curves or two surfaces.

Algorithm V (Curve Intersections)

Given $P \equiv [P_0, \cdots, P_m]$ and $Q \equiv [Q_0, \cdots, Q_n]$ the coefficients of two Bernstein polynomials, on $[0, 1]$, find the points of intersection (if any) of the two curves. This algorithm is given in its recursive form.

PROCEDURE CURVE INTERSECT (P, m, Q, n)

Step 1. [Test For Overlap] If $\max(P) < \min(Q)$ or $\min(P) > \max(Q)$ return.

Step 2. [Test For Convergence]. Let $l(A, B)$ be the line segment from point A to point B and let $d(A, B)$ be the distance between the two points. Then if

$d(l(P_0, P_m), P_i) < \text{Tolerance}$

$i = 1, 2, \cdots, m - 1$, and

$d(l(Q_0, Q_n), Q_i) < \text{Tolerance}$

$i = 1, 2, \cdots, n - 1$, output $l(P_0, P_m) \wedge l(Q_0, Q_n)$ (may be empty) and return.

Step 3. [Subdivide and Retest].

CURVESPLIT (P, m, R, S)
CURVESPLIT (Q, n, T, U)
CURVE INTERSECT (R, m, T, n)
CURVE INTERSECT (R, m, U, n)
CURVE INTERSECT (S, m, T, n)
CURVE INTERSECT (S, m, U, n)
return.

Algorithm VI (Surface Intersection)

Given $P \equiv [P_{ij}]\, i = 0, 1, \cdots, m;\ j = 0, 1, \cdots, n$ and $Q \equiv [Q_{ij}]\, i = 0, 1, \cdots, p;\ j = 0, 1, \cdots, q$ the coefficients of two nondegenerate Bernstein polynomial surfaces on $[0, 1] \times [0, 1]$, find the curves of intersection (if any) of the two surfaces.

Surface intersection proceeds as with the curve intersection algorithm, except that instead of producing points as the intersection of line segments, the surface algorithm line segments as the intersection of planar segments. Since these line segments are found in a somewhat arbitrary order, they must be sorted by endpoint matchup to form a connected curve. In all probability this sorting can be done on (xyz) coefficients alone, but since parametric surfaces can be multivalued the proper sorting key is the parametric values associated with each endpoint. These parametric values are easily tracked during the subdivision process; in fact, if we extend P and Q to P' and Q', respectively, such that $P'_{ij} = (P_{ij}, i/m, j/n)$ $i = 0, 1, \cdots, m;\ j = 0, 1, \cdots, n$, and $Q'_{ij} = (Q_{ij}, i/p, j/q)$, $i = 0, 1, \cdots, p, j = 0, 1, \cdots, q$, we need only subdivide on the extended coefficients to compute the proper parametric values associated with each new coefficient with respect to the original surface.

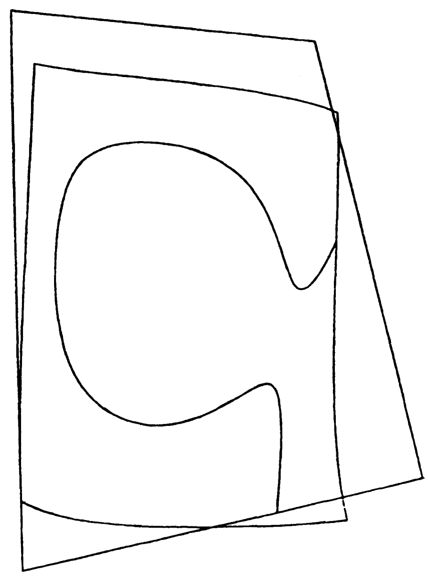


Fig. 18. A bilinear surface intersected with a bicubic.

PROCEDURE SURFACEINTERSECT (P', m, n, Q', p, q)
Step 1. [Test for Overlap].
Project P' onto P, Q' onto Q. (Drop parametric coefficients.) If $\max P < \min Q$ or $\min P > \max Q$ return.
Step 2. [Test for Convergence].
Let $l(A, B, C)$ be the plane through three points A, B, C, let $l(A, B)$ be the line segment through A and B, and let $d(A, B)$ be the distance between any two points. Then if

$$d(l(P_{0,0}, P_{m,0}, P_{0,n}), P_{ij}) < \text{Tolerance}$$
$$d(l(P_{0,0}, P_{m,0}), P_{i,0}) < \text{Tolerance}$$
$$d(l(P_{0,n}, P_{m,n}), P_{i,n}) < \text{Tolerance}$$
$$d(l(P_{0,0}, P_{0,n}), P_{0,j}) < \text{Tolerance}$$
$$d(l(P_{m,0}, P_{mn}), P_{m,j}) < \text{Tolerance}$$

$i = 0, 1, \cdots, m; j = 0, 1, \cdots, n$, and

$$d(l(Q_{0,0}, Q_{p,0}, Q_{0,q}), Q_{ij}) < \text{Tolerance}$$
$$d(l(Q_{0,0}, Q_{p,0}), Q_{i,0}) < \text{Tolerance}$$
$$d(l(Q_{0,q}, Q_{p,q}), Q_{i,q}) < \text{Tolerance}$$
$$d(l(Q_{0,0}, Q_{0,q}), Q_{0,j}) < \text{Tolerance}$$
$$d(l(Q_{p,0}, Q_{p,q}), Q_{pj}) < \text{Tolerance}$$

then output $l(P'_{0,0}, P'_{m,0}, P'_{0,n}) \Lambda < (Q'_{0,0}, Q'_{p,0}, Q'_{0,q})$ (may be empty) and return.
Step 3. [Subdivide and Retest for all Intersections].
SURFSPLIT (P', R, S, T, U, m, n)
SURFSPLIT (Q', V, W, X, Y, p, q)
SURFACEINTERSECT (R, m, n, V, p, q)
SURFACEINTERSECT (R, m, n, W, p, q)
SURFACEINTERSECT (R, m, n, X, p, q)
SURFACEINTERSECT (R, m, n, Y, p, q)
SURFACEINTERSECT (S, m, n, V, p, q)
SURFACEINTERSECT (S, m, n, W, p, q)
SURFACEINTERSECT (S, m, n, X, p, q)
SURFACEINTERSECT (S, m, n, Y, p, q)
SURFACEINTERSECT (T, m, n, V, p, q)
SURFACEINTERSECT (T, m, n, W, p, q)
SURFACEINTERSECT (T, m, n, X, p, q)
SURFACEINTERSECT (T, m, n, Y, p, q)
SURFACEINTERSECT (U, m, n, V, p, q)
SURFACEINTERSECT (U, m, n, W, p, q)
SURFACEINTERSECT (U, m, n, X, p, q)
SURFACEINTERSECT (U, m, n, Y, p, q)
return.

VI. Conclusion

We have presented some new algorithms for the evaluation of polynomial spline curves and surfaces, and have shown how these algorithms can be useful in solving many of the geometric problems encountered in CAD/CAM applications. We have applied the subdivision technique to piecewise polynomial curves and surfaces but the approach is equally valid for other function spaces and for disciplines other than CAD/CAM.

Acknowledgment

The authors are indebted to A. R. Forrest and E. Catmull for their early recognition of the importance of subdivision methods, and to R. F. Sproull for many useful suggestions on a previous draft. We also wish to thank J. Bloomenthal and L. Carpenter for their assistance in the generation and production of the smooth-shaded illustrations. K. S. Fu has been an especially helpful communicating editor for this paper.

References

[1] R. E. Barnhill and R. F. Riesenfeld, Eds., *Computer Aided Geometric Design*. New York: Academic, 1974.
[2] J. M. Lane and R. F. Riesenfeld, "The application of total positivity to computer aided curve and surface design," Univ. of Utah, Tech. Rep. UUCS-79-115, 1979.
[3] W. J. Gordon and R. F. Riesenfeld, "B-spline curves and surfaces," in *Computer Aided Geometric Design*, R. Barnhill and R. F. Riesenfeld, Eds. New York: Academic, 1974.
[4] P. Bézier, *Numerical Control-Mathematics and Applications*. London, England: Wiley, 1972.
[5] R. P. Dube, G. J. Herron, F. F. Little, and R. F. Riesenfeld, "Surfed–A system for curve and design," *CAD*, Mar. 1978.
[6] P. Bézier, "Mathematical and practical possibilities of unisurf,"

in *Computer Aided Geometric Design*. New York: Academic, 1974.

[7] S.-C. Wu, J. F. Abel, and D. P. Greenberg, "An interactive computer graphics approach surface representation," *Commun. Ass. Comput. Mach.*, Oct. 1977.

[8] F. F. Little, private communication.

[9] C. deBoor, "On calculating with *B*-splines," *J. Approx. Theory*, vol. 6, pp. 50-62, 1972.

[10] M. G. Cox, "The numerical evaluations of *B*-splines," *J. Inst. Math. Appl.*, vol. 10, pp. 134-149, 1972.

[11] H. B. Curry and I. J. Schoenberg, "On polya frequency functions IV: The fundamental spline functions and their limits," *J. d'Anal. Math.*, vol. 17, pp. 71-107, 1966.

[12] R. F. Riesenfeld, "On Chaikins algorithm," *Comput. Graphics and Image Processing*, vol. 4, pp. 304-310, 1975.

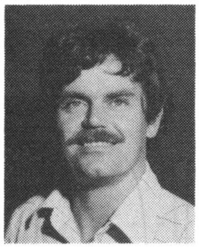
[13] M. I. Shamos, "Computational geometry," Ph.D. dissertation, Yale Univ., New Haven, CT, 1975.

[14] R. A. Schumaker, R. F. Sproull, and I. E. Sutherland, "A characterization of ten hidden-surface algorithms," *Computing Surveys*, vol. 6, Mar. 1974.

[15] D. Knuth, *The Art of Computer Programming*, vol. 1. Reading, MA: Addison-Wesley, 1969.

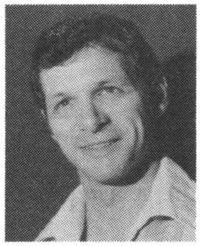
[16] E. Catmull, "A subdivision algorithm for computer display of curved surfaces," Ph.D. dissertation, Univ. of Utah, 1974.

[17] J. Lane, L. Carpenter, T. Whitted, and J. Blinn, "Scanline methods for displaying parametrically defined surfaces," *Commun. Ass. Comput. Mach.*, to be published.



Jeffrey M. Lane was born in Marblehead, MA, on March 19, 1948. He received the B.Sc. degree in mathematics from Syracuse University, Syracuse, NY, in 1970, the M.S. degree in computer science from the University of Oregon, Eugene, in 1973, and the Ph.D. degree in computer science from the University of Utah, Salt Lake City, in 1977.

He is now with the Boeing Company, Seattle, WA. His research interests are in approximation theory, computer graphics, computer-aided design, and computational geometry.



Richard F. Riesenfeld received the Ph.D. degree from the School of Computer and Information Science at Syracuse University, Syracuse, NY, in 1973.

He is currently an Associate Professor of Computer Science at the University of Utah, where he joined the faculty in 1972. Computer-aided geometric design and computer graphics are his areas of specialty in research.