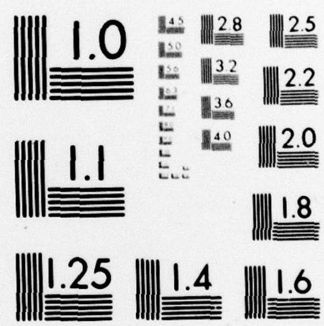
AD-A073 422 UTAH UNIV SALT LAKE CITY DEPT OF COMPUTER SCIENCE F/G 9/2
SURFACE REPRESENTATION-SYMBOLIC COMPUTATION.(U)
APR 78 R E BARNHILL, R F RIESENFELD N00014-77-C-0157

UNCLASSIFIED NL

1 OF 1
AD
A073422

END
DATE
FILMED
10-79
DDC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

LEVEL II (12)

MIL-STD-847A
31 January 1973

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) (6) SURFACE REPRESENTATION - SYMBOLIC COMPUTATION.		5. TYPE OF REPORT & PERIOD COVERED (9) Technical Rept.,
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) (10) R. E. BARNHILL, R. F. RIESENFELD		8. CONTRACT OR GRANT NUMBER(s) (15) N00014-77-C-0157, NSF-MCS74-13017
9. PERFORMING ORGANIZATION NAME AND ADDRESS University of Utah, Computer Science Department, University of Utah, Salt Lake City 84112		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research, Dept. of Navy, Arlington, Virginia 22217 (Denicoff)		12. REPORT DATE Apr. 78
		13. NUMBER OF PAGES 7
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Office of Naval Research Resident Representative, Univ. of California, 553 Evans Hall, Berkeley, CA 94720		15. SECURITY CLASS. (of this report)
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

(12) 8 p.

This document has been approved for public release and sale; its distribution is unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

(11) Apr 78

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

79 08 29 101

404 949

Abstract

Recent research in Computer Aided Geometric Design is surveyed in this paper. A demonstration of the feasibility and utility of some of the new methods of curve and surface representation for interactive design is given in a movie which will be shown during the presentation.

PART I

by R. E. Barnhill

SURFACE REPRESENTATION

The representation of surfaces is needed for computer aided ship design and for the finite element analysis of ship hulls.

This module of our paper is a very broad survey of recent methods for representing and approximating surfaces. For more details consult the survey by Barnhill (1977), which also contains the relevant references.

There are three geometric units for surfaces: rectangles, triangles, and points. The approximation used depends upon which of these three correspond to the geometry of the data.

Rectangular Methods

Data are often given along orthogonal lines, either all along the lines or at the meshpoints of the corresponding rectangular grid. Let $F = F(u,v)$ be a "general coordinate", i.e., F stands for x, y, and z successively. The simplest Coons patch interpolates to the data $F(0,v)$, $F(1,v)$, $F(u,0)$, and $F(u,1)$ and is the following:

$$(1.1)\quad B(u,v) = [1-u \;\; u]\begin{pmatrix} F(0,v) \\ F(1,v) \end{pmatrix} + [F(u,0) \;\; F(u,1)]\begin{pmatrix} 1-v \\ v \end{pmatrix} - [1-u \;\; u]\begin{pmatrix} F(0,0) & F(0,1) \\ F(1,0) & F(1,1) \end{pmatrix}\begin{pmatrix} 1-v \\ v \end{pmatrix}$$

$$0 \le u, v \le 1$$

The bilinearly blended Coons patch defined by (1.1), considered over a mesh of rectangles, provides a continuous, locally defined interpolant. Useful surfaces usually must be at least continuously differentiable, i.e., C^1. The bicubically blended Coons patch is a start on the problem of defining a C^1 surface and it is given by the following:

$$(1.2)\quad PF(u,v)$$

$$= [h_0(u) \;\; h_1(u) \;\; \bar{h}_0(u) \;\; \bar{h}_1(u)]\begin{pmatrix} F(0,v) \\ F(1,v) \\ F_{1,0}(0,v) \\ F_{1,0}(1,v) \end{pmatrix} + [F(u,0) \;\; F(u,1) \;\; F_{0,1}(u,0) \;\; F_{0,1}(u,1)]\begin{pmatrix} h_0(v) \\ h_1(v) \\ \bar{h}_0(v) \\ \bar{h}_1(v) \end{pmatrix} - [h_0(u) \;\; h_1(u) \;\; \bar{h}_0(u) \;\; \bar{h}_1(u)]\, B \begin{pmatrix} h_0(v) \\ h_1(v) \\ \bar{h}_0(v) \\ \bar{h}_1(v) \end{pmatrix}$$

where

$$(1.3)\quad B = \begin{pmatrix} F(0,0) & F(0,1) & F_{0,1}(0,0) & F_{0,1}(0,1) \\ F(1,0) & F(1,1) & F_{0,1}(1,0) & F_{0,1}(1,1) \\ F_{1,0}(0,0) & F_{1,0}(0,1) & \frac{\partial}{\partial u}F_{0,1}(0,0) & \frac{\partial}{\partial u}F_{0,1}(0,1) \\ F_{1,0}(1,0) & F_{1,0}(1,1) & \frac{\partial}{\partial u}F_{0,1}(1,0) & \frac{\partial}{\partial u}F_{0,1}(1,1) \end{pmatrix}$$

The h_i and $\bar{h}_i$ are the univariate cubic Hermite basis functions

(1.4)

$$h_0(t) = (1-t)^2(2t+1) \quad \bar{h}_0(t) = t(1-t)^2$$
$$h_1(t) = t^2(-2t+3) \quad \bar{h}_1(t) = t^2(t-1).$$

For some time people thought that (1.2) provided a C^1 interpolant to the data, F and the normal derivative $\frac{\partial F}{\partial n}$, all around the boundary of $\{(u,v): 0 \leq u, v \leq 1\}$. However, $\bar{P}F$ is C^1 but fails to interpolate to $F_{0,1}(u,0)$ or to $F_{0,1}(u,1)$, in general. A "compatibly corrected" Coons patch that does interpolate to the data is (1.2) with the twist partition (the lower right partition) of B replaced by:

(1.5)

$$\frac{u\frac{\partial^2 F}{\partial v \partial u}(0,0) + v\frac{\partial^2 F}{\partial u \partial v}(0,0)}{u+v}$$

$$\frac{-u\frac{\partial^2 F}{\partial v \partial u}(0,1) + (v-1)\frac{\partial^2 F}{\partial u \partial v}(0,1)}{-u+v-1}$$

$$\frac{(1-u)\frac{\partial^2 F}{\partial v \partial u}(1,0) + v\frac{\partial^2 F}{\partial u \partial v}(1,0)}{1-u+v}$$

$$\frac{(u-1)\frac{\partial^2 F}{\partial v \partial u}(1,1) + (v-1)\frac{\partial^2 F}{\partial u \partial v}(1,1)}{u+v-2}$$

(enumerated row-wise)

All of these Coons patches can be discretized to form finite dimensional interpolants. For example, in (1.2), the positions such as F(u,0) could be cubic Hermite polynomials and the normal derivatives such as $F_{0,1}(u,0)$ could be linear polynomials.

The (1,1) derivatives are called "twists". These are awkward geometric handles which are best avoided. Two sets of solutions to avoiding the specification of twists have recently been developed (Barnhill (1977)):

(1) Construct a preprocessor that calculates the twists from an intermediate C^0 surface.

(2) Construct interpolants whose twists are calculated in terms of F, $F_{1,0}$, and $F_{0,1}$.

Triangular Methods

Triangular Coons patches were initiated by Barnhill, Birkhoff, and Gordon (1973). These interpolants can be discretized like the rectangular ones, to obtain finite dimensional schemes. A variety of 9-parameter C^1 triangular interpolants have been devised and implemented in our SURFED system. Some of these interpolants are illustrated in the movie which goes with this paper. Seven sets of triangular interpolants are given in Barnhill (1977). Before interpolating over triangles, the triangles themselves must be available. F. F. Little has devised an effective method of triangulating given $\{(x_i, y_i)\}_{i=1}^n$ and then of optimizing the triangulation. This is an example of a two-level process: (1) preprocess by means of a fast first triangulation and (2) improve the first triangulation by optimizing according to a certain criterion.

An example of a C^1 interpolant over a triangulation is the Barnhill, Birkhoff, and Gordon scheme given in Klucewicz (1977). The "transfinite" triangular scheme analogous to (1.2) for the standard triangle with vertices (1,0), (0,1), and (0,0) is the following:

(2.1)

$$PF(p,q) = h_0\left(\tfrac{p}{1-q}\right)F(0,q) + \bar{h}_0\left(\tfrac{p}{1-q}\right)(1-q)\cdot$$
$$F_{1,0}(0,q) + P_2F - h_0\left(\tfrac{p}{1-q}\right)P_2F(0,q)$$
$$+ \bar{h}_0\left(\tfrac{p}{1-q}\right)(1-q)\ \left[\tfrac{\partial}{\partial p}P_2F\right](0,q)$$

where

$$P_2F = h_0\left(\tfrac{q}{1-p}\right)F(p,0) + \bar{h}_0\left(\tfrac{q}{1-p}\right)(1-p)F_{0,1}(p,0)$$
$$+ h_1\left(\tfrac{q}{1-p}\right)F(p,1-p) + \bar{h}_1\left(\tfrac{q}{1-p}\right)(1-p)\cdot$$
$$F_{0,1}(p,1-p) - \frac{p^2q(p+q-1)^2}{p+q}\left[\left(\tfrac{\partial}{\partial q}\left(\tfrac{\partial F}{\partial p}\right)\right)(0,0)\right.$$
$$\left.-\left(\tfrac{\partial}{\partial p}\left(\tfrac{\partial F}{\partial q}\right)\right)(0,0)\right].$$

Broadly speaking, rectangular patches should be used where possible, e.g., along the simpler regions of ship hulls. Triangular patches should be used for the more complicated regions which do not have rectangular-like symmetry. Of course, for arbitrarily spaced point data, triangular patches or else the methods of the next Section must be used. Rectangular and triangular patches can be blended smoothly together, with common parameters.

Arbitrarily Spaced Data

For applications such as mapping the bottom of a harbor, the available data is likely to be irregularly spaced. Such data can be interpolated either: (1) by triangulating and then using a triangular interpolant, as in Section 2, or else (2) by using a variant of Shepard's Formula. Shepard's Formula for the data $\{(x_i, y_i, F_i)\}_{i=1}^n$ is the following:

$$(3.1)\quad SF = \frac{\sum_{i=1} w_i F_i}{\sum_{i=1}^{n} w_i}, (x,y) \neq (x_i,y_i), i-1,\ldots n$$

$$SF = F_i,\ (x,y) = (x_i,y_i) \text{ for some } i\ .$$

$$\text{where } w_i = \prod_{\substack{j=1 \\ j \neq i}}^{n} d_j^{\mu} \text{ and } d_j \quad (x-x_j)^2 + (y-y_j)^2 {}^{1/2}\ .$$

We recommend letting $\mu = 2$, in which case Shepard's Formula is an inverse square distance method. Shepard's formula itself has some defects, such as the property that

$$(3.2)\qquad \frac{\partial SF}{\partial x} = \frac{\partial SF}{\partial y} = 0 \text{ at each } (x_i,y_i).$$

Therefore, we have the following Theorem of Barnhill and Gregory to find improved interpolants: If P and Q are two interpolation operators, then their Boolean sum defined by

$$(3.3)\qquad P \oplus Q \equiv P \oplus Q - PQ$$

has the following properties

(1) $P \oplus Q$ has at least the interpolation properties of P and

(2) $P \oplus Q$ has at least the function precision of Q.

This Theorem yields many possible improvements of Shepard's Formula. For example, let PF be SF defined in (3.1) and let QF be the quadratic least squares approximation to the F_i. Then $(P \oplus Q)F$ has the following properties:

(1) interpolates to the F_i.

(2) does not have the flat spots due to (3.2).

(3) is in continuity class C^∞.

(4) has quadratic precision.

This Theorem has been implemented by D. Feng so that symbolic P and Q can be entered in $P \oplus Q$. R. F. Riesenfeld's module of this paper reports on Feng's work.

Acknowledgments

This research was supported in part by the National Science Foundation with Grant MCS 74-13017 A01.

References

1. R. E. Barnhill, Representation and Approximation of Surfaces, Mathematical Software III, (edited by J. R. Ric· Academic Press (to appear), 1977, pp. 68-119.

2. R. E. Barnhill, G. Birkhoff, and W. J. Gordon, Smooth Interpolation in Triangles, J. Approx. Theory 8, 114-128, 1973.

3. S. A. Coons, Surfaces for Computer Aided Design Division, Mech. Engin. Dept., MIT, revised, 1967.

4. I. M. Klucewicz, A Piecewise C^1 Interpolant to Arbitrarily Spaced Data, M.S. thesis Mathematics Department, University of Utah, 1977.

PART II

by R. F. Riesenfeld

SYMBOLIC COMPUTATION

Although scientific computation traditionally has had the connotation of numerical computation, advances in the area of symbolic computation have brought its use increasingly into the domain of scientific and engineering computation. Classical numerical analysis gives rise to algorithms and procedures that, when carried out with suitable input values, produce a single number of small set of numbers which we would call "the answer" to a given problem. It is typical of these numerical methods that any change in the input values results in a new problem whose answer costs as much as the previous one. When a problem is solved symbolically, it is solved in such a way that the "answer," typically an expression involving variables, has many valid interpretations that it can take on, being restricted only by the domain of definition of the variables in the expression. This kind of computation is what we humanly perform when we manipulate equations algebraically or when we solve for an indefinite integral. A specific answer is obtained when we bind the variables by substituting specific values for them in the final expression. A similar problem can be solved simply by making a new evaluation of the previously obtained symbolic answer, a process which is often insignificant compared to the amount of work required to compute the original expression.

In 1969 Gordon published a milestone paper which described a general framework for classifying and studying multivariate interpolation schemes. That paper introduced the use of idempotent linear operators, called projectors, to develop an elegant theory in which the Coons patch with its extraordinary interpolation properties is merely the boolean sum $P_1 + P_2$ of two appropriate projectors. At the time of its publication, however, the principal value of the Gordon projectors was primarily theoretical. It gave analysts an organization for considering multivariate interpolants, but projectors probably did not become an actual research tool until people began to study interpolants defined over nonrectilinear regions like triangles. These interpolants are as manifoldly more complex in their structure as they are in their algebraic embodiment. The formula of a triangular patch, written out as algebraic combinations of parametric variables and point evaluations of functions and function derivatives, can easily occupy a full page of a technical report devoted to its description. As an example, see equation (2.1) in Part I. On the other hand, in projector notation, the representation of the same surface interpolant may be very simply expressed as the boolean sum of two or three projectors whose univariate definition is straightforward. It was this impetus that moved D. Feng to investigate the applicability of symbolic computation to computer aided geometric design.

Making some reasonable restrictions, Feng implemented a symbolic processor that allows the user to specify surfaces as boolean combinations of the commonly used projectors. This capability is interfaced with SURFED, the surface editor developed and used by the CAGD Group to analyze and inspect visually various new surface forms. This implementation was realized in FORTRAN by writing the necessary stack support routines in FORTRAN, so that the code is portable, compilable, and compatible with the rest of our system. The addition of this symbolic capability to the surface system is essentially the addition of an <u>interactive</u> symbolic surface specification feature, for it is possible to request the display of a surface that was specified only in projector form. The necessary symbolic computations, and subsequent numerical evaluations are carried out in order to produce a realtime picture of the surface being investigated. Modifications to the parameters of the surface are effected by numerically re-evaluating the same symbolic expression, just as one would in the case in which the formula has been explicitly typed in from a terminal.

One major advantage of this symbolic processor is that the algebraic computations necessary to yield lengthy interpolation formulas can be mechanized so that one has much greater confidence in the correctness of the answer. Even if the computation is performed correctly by hand, entering the formula is also a very error-prone operation. The fact that surface specification can be an effortless interactive activity of forming expressions in a natural high-level language, instead of an onerous low-level chore, means that the researcher is far more likely to experiment with new surface forms and new ideas, a general benefit that interactive computing systems are supposed to engender. Indeed, Feng employed his processor to devise some new surface forms.

<u>Smooth Curves and Total Positivity</u>

Designing curves that are aesthetic, graceful, and "sweet" has been part of geometric design as long as this has been considered an area of human endeavor. The field of CAGD has inherited this problem, and consequently it has been concerned with making quantified statements in this hithertofore qualitative, nonscientific process. Judgment and experience were the essential ingredients of geometric design before the introduction of the computer. Recently there has been considerable research devoted to mathematical analysis of approximations that are "wiggle free", and more recently there have been efforts to apply these ideas toward developing improved human-machine graphical interfaces for CAGD. In 1977 J. Lane linked the mathematical theory of total positivity to this application, a relationship which will be outlined in this section.

P. Bézier of Renault attracted widespread attention in the early 1970's with a method of curve design that seemed to assure the user that the output curve had satisfactory shape characteristics. Using a polygon to roughly specify the curve, Bézier succeeded in gaining interactive control over shape. An analysis of Bézier curves revealed that they are mathematically related to Bernstein approximation. In particular Bézier curves shared the exceptional property of being <u>variation diminishing</u>.

<u>Definition TP1</u>: <u>An approximation scheme is variation</u> diminishing is the number of intersections of the approximation with any straight line does not exceed the number of crossings of that straight line by the primitive function.

This definition captures the intuitive notion that the approximation to a primitive function should have no more wiggles in it than the original primitive function itself has.

An analytical approach to the theory of variation diminishing approximations has been developed by Schoenberg, Karlin, and others. They have shown that, if a set of basis functions (blending functions) is totally positive, then the associated approximation method is variation diminishing. Although the connection between the theory of total positivity and the variation diminishing property is not intuitive, the definition of total positivity is given below as an indication of the kind of mathematical analysis that it invokes.

Definition TP2: A set of basis functions $\{\phi_i(x)\}$ is totally positive provided that, for all $i_1 < i_2 \ldots < i_n$ and $x_1 < x_2 < \ldots < x_n$, we have the inequality

$$\det \begin{vmatrix} \phi_{i_1}(x_1) & \ldots & \phi_{i_1}(x_n) \\ \vdots & & \\ \phi_{i_n}(x_1) & \ldots & \phi_{i_n}(x_n) \end{vmatrix} \geq 0,$$

for all finite values of n.

The application of total positivity has shown in a straightforward manner that the Bézier approximation method is totally positive, thus variation diminishing. Furthermore it has been used to establish that B-spline methods are also variation diminishing. By using total positivity Lane has developed new methods for curve design which incorporate the tension-like properties studied by Nielson and Dube and others, while maintaining the valuable variation diminishing property. This feature permits the designer to "tighten" a curve within a certain region where its appearance is unsatisfactory.

There are several composition theorems that assure the total positivity of compound functions that are composed of more elementary totally positive functions. These theorems make it possible to devise many new schemes that enjoy the variation diminishing property. It also helps one to analyze various ad hoc schemes that have been proposed.

The proofs and deductions in the theory of total positivity tend to be somewhat specialized to the area, but as researchers in CAGD become more accustomed to them, it is likely to become a more widely used concept and tool.

Hayes Surface Form

The standard tensor product definition of a surface is given by

$$\sum_{i=1}^{M} \sum_{j=1}^{N} P_{ij} \phi_i(u) \phi_j(v)$$

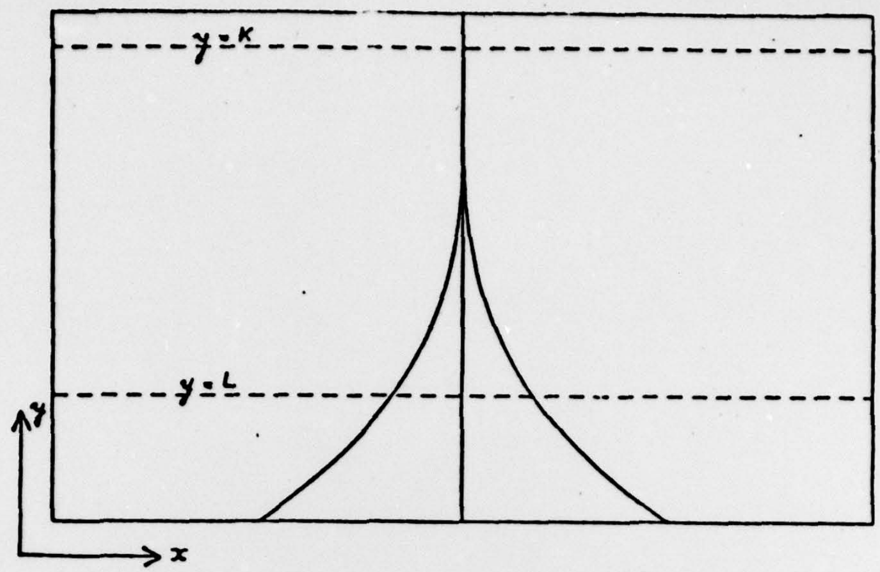
Hayes observed that, while it was traditional for the points P_{ij} to fall on a rectangular grid, it was not essential for the application of the tensor product formula. It is only necessary that there be a P_{ij} defined for all valid index values. For the case of tensor product B-spline surfaces, special effects are possible if the Cox-deBoor Algorithm is used to evaluate them.

What are the knots or points of derivative discontinuity in univariate spline curves become whole lines of derivative discontinuities or knot lines in the tensor product surface. Normally these knot lines correspond to an orthogonal set of lines in the parameter domain, but Hayes noticed that so long as the order of the knot lines was not violated, the tensor product B-spline formula was formally defined and made sense. In fact the lines can be allowed to coalesce and the Cox-deBoor algorithm yields a surface with diminished or deficient continuity across the coalescent knot lines, just as coalescent knots yield deficient splines in the univariate case. If enough knot lines are brought together, a knot line of sufficient deficiency to produce a cusp is produced. This situation is depicted in the figure.

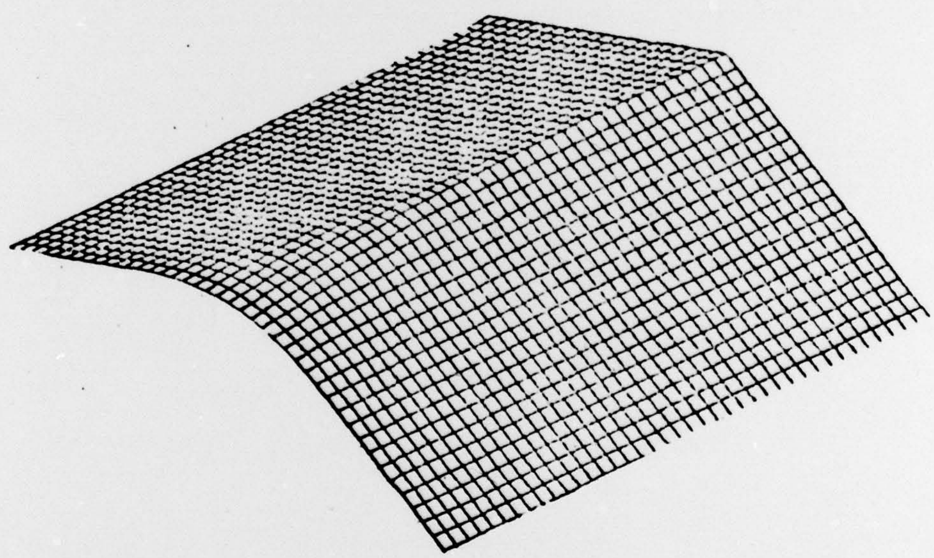
Finally Hayes pointed out that the knot lines themselves can be defined in the parameter space using the B-spline method of designing curves. The resulting power of this more general form is that one can control cusps in B-spline surfaces, feathering them out in a very smooth and aesthetic manner. This is one of the most useful methods of design that involve manipulating the locations of the knots of a spline in the parameter space. Perhaps this approach will encourage people to try other variations that employ parameters related to the domain of definition of the surface. A generalization and extension of the curved knot line technique to transfinite interpolants is thoroughly given by Nielson and Wixom (4).

Acknowledgments

This research was supported in part by the National Science Foundation (MCS-74-13017-A01) and the Office of Naval Research (N00014-77-C-0157).



A knot-set producing a fading discontinuity



A surface containing a fading discontinuity in slope

(Courtesy of J. G. Hayes [2, p. 12-13]. Figures from original Hayes NPL Rep. NAC 58)

References

1. D. Y. Feng, "A Symbolic System for the development of Surface Interpolates," University of Utah, 1977.

2. J. G. Hayes, "New Shapes from Bicubic Splines," NPL Report #58, 1974.

3. J. M. Lane, "Shape Operators for Computer Aided Geometric Design," University of Utah, 1977.

4. G. M. Nielson and J. A. Wixom "Approximation Theory Techniques for Curve and Surface Description," <u>These Proceedings</u>.

5. R. E. Barnhill and R. F. Riesenfeld, joint editors, <u>Computer Aided Geometric Design</u>, Academic Press, 1974.

6. SURFED - "An Interactive Editor for for Free-Form Surfaces," by R. Peter Dube, Gary J. Herron, Frank F. Little, and Richard F. Riesenfeld.