# Families of Local Matrix Splines

*Tom Duff*
*Computer Division*
*Lucasfilm Ltd*

*Technical Memo No. 104*
*21 Dec 1983*

Presented as tutorial notes (by Alvy Ray Smith) at the 1983 SIGGRAPH, July 1983. This document was reentered in Microsoft Word on 1 Dec 1999. Spelling and punctuation are generally preserved, but trivially minor spelling errors are corrected. Otherwise additions or changes made to the original are noted inside square brackets or in footnotes.

## Abstract

Jim Clark's Cardinal splines are a family of local interpolating splines with an adjustable tension parameter. The family may be described by a matrix which yields spline coefficients as linear functions of knot values. We characterize the Cardinal spline matrix in a way which suggests a method of adding tension to B-splines, and show that this tension corresponds to the $b_2$ parameter of Brian Barsky's $b$-splines. The similarity between the tension parameters of these two splines suggests looking for an interpolating spline family which incorporates a bias parameter analogous to Barsky's $b_1$. We demonstrate two such families with different continuity properties (one is $G^1$, the other $C^1$). Finally, we develop a five-parameter characterization of all $C^0$, $G^1$ translation invariant cubic matrix splines and indicate that all the families we have developed are sub-families of it.

**CR Categories and Subject Descriptors**: G.1.1 [Numerical Analysis] Spline and piecewise polynomial interpolation, G.1.2 [Numerical Analysis] Spline and piecewise polynomial approximation, I.3.5 [Computer Graphics] Curve, surface, solid and object representations.

**General Terms:** Algorithms

**Additional Keywords and Phrases:** $b$-spline, Cardinal spline, matrix spline, local spline, computer aided geometric design

## Introduction

A *local* spline is one for which changing the value of a single knot affects only a bounded number of spline segments in the knot's vicinity. This is a particularly useful property for geometric design systems and computer animation systems, since it means that a designer can adjust the appearance of a particular part of his design without fear of global complications.

Many popular local splines may be characterized by a system of linear equations relating the spline's knots to its polynomial coefficients. These linear equations may be written as a rectangular matrix.

## The Matrix Spline Notation

Given a sequence of *control points* or *knots*, $K_i$ $0 \le i < m$ and an $n+1$ by $s$ matrix $M$ we can define the matrix spline $S_M$ of degree $n$ and support $s$ by

$$S_M(t+i) = \sum_{j=0}^{n} \sum_{k=0}^{s-1} t^{n-j} M_{jk} K_{i+k} \quad 0 \le t < 1, 0 \le i < m-s+1 .$$

We will usually omit the subscript $M$ when the matrix is clear from context, and we will use the notation $S_i(t) = S(i+t)$.

In computer graphics we are particularly concerned with the case $n = 3$, $s = 4$, which makes $M$ one of the 4 by 4 matrices so familiar to computer graphics hardware and software. In this case the above equation may be rewritten as

$$S_M(t+i) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} M \begin{bmatrix} K_i \\ K_{i+1} \\ K_{i+2} \\ K_{i+3} \end{bmatrix} \quad 0 \le t < 1, 0 \le i < m-3 .$$

Most (but not all, see for example [Knuth]) of the well-known local splines may be expressed in this form. For example, the uniform cubic B-spline [Riesenfeld] is represented by

$$M_{Bspline} = \frac{1}{6} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} .$$

Without loss of confusion [sic], we will identify matrices with their splines. Thus, when we speak of “the derivative of $M$”, we will mean “the derivative of the spline whose matrix is $M$”.

The first use of the matrix spline notation of which I am aware is [Catmull].

## Families of Splines

Brian Barsky introduced a two-parameter family of matrix splines called the $b$-splines (see [Barsky]). The $b$-splines are continuous and have continuous unit tangent and curvature vectors. In particular, the $b$-splines satisfy

$$\begin{aligned} S_i(1) &= S_{i+1}(0) \\ b_1 S_i'(1) &= S_{i+1}'(0) \\ b_1^2 S_i''(1) + b_2 S_i'(1) &= S_{i+1}''(0) \end{aligned} .$$[1]

Barsky calls the last two of these conditions $G^1$ and $G^2$. They are *geometric* generalizations of the *parametric* continuity conditions $C^1$ and $C^2$. The $b$-spline matrix is

[1] Here $S_i'(1)$ is the derivative from the left and $S_{i+1}'(0)$ is the derivative at the same point from the right.

$$M_{Beta} = \frac{1}{b_2 + 2b_1^3 + 4b_1^2 + 4b_1 + 2} \begin{bmatrix} -2b_1^3 & 2(b_2 + b_1^3 + b_1^2 + b_1) & -2(b_2 + b_1^2 + b_1 + 1) & 2 \\ 6b_1^3 & -3(b_2 + 2b_1^3 + 2b_1^2) & 3b_2 + 6b_1^2 & 0 \\ -6b_1^3 & 6(b_1^3 - b_1) & 6b_1 & 0 \\ 2b_1^3 & b_2 + 4b_1^2 + 4b_1 & 2 & 0 \end{bmatrix}$$

$b_1$ biases or slews the curve to the left or right (parametrically) of the unbiased B-spline. As $b_2$ increases, the curve becomes more tense and more closely approximates its knots.

Another useful family of splines is the Cardinal splines [Clark]. The Cardinal splines are interpolating splines with first derivative continuity. They are defined by the constraints

$$\begin{aligned} S_i(0) &= K_{i+1} \\ S_i(1) &= K_{i+2} \\ S_i'(0) &= c(K_{i+2} - K_i) \\ S_i'(1) &= c(K_{i+3} - K_{i+1}) \end{aligned}.$$

Solving these equations for the spline coefficients yields the matrix

$$\begin{bmatrix} -c & 2-c & c-2 & c \\ 2c & c-3 & 3-2c & -c \\ -c & 0 & c & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}.$$

Using the notation $\mathrm{lerp}(M, N, a) = (1-a)M + aN$ (lerp is an abbreviation for *linear interpolant*)[2], we can rewrite this as $\mathrm{lerp}(M_{Ease}, M_{c=1}, c)$ where

$$M_{Ease} = \begin{bmatrix} 0 & 2 & -2 & 0 \\ 0 & -3 & 3 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix},\ M_{c=1} = \begin{bmatrix} -1 & 1 & -1 & 1 \\ 2 & -2 & 1 & -1 \\ -1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}. \qquad (1)$$

Thus, each of the Cardinal splines is a weighted average of the $c = 1$ spline, which has $S_i'(0) = K_{i+2} - K_i$ and $S_i'(1) = K_{i+3} - K_{i+1}$, and the Ease spline, which has $S_i'(0) = S_i'(1) = 0$[3]. The Cardinal splines may be thought of as variable-tension local interpolating splines which take their corners more sharply as $c \to 0$. An interesting member of the Cardinal spline family has $c = .5$. This is the Catmull-Rom cubic spline, described in [Catmull-Rom] and later in [Brewer-Anderson].

Equation (1) immediately suggests a method of adjusting the tension of B-splines using

---

[2] Note that lerping two spline matrices is equivalent to lerping the coefficients of the polynomials they generate, which in turn is equivalent to lerping the points on the spline curves and their derivatives.

[3] The Ease spline curve is the polygon that interpolates its knots. Its velocity decelerates to zero as it passes through each knot. Thus, its motion as a function of $t$ `eases' in and out of each knot. The term *ease* is taken from the lexicon of conventional (not computer) animators.

$$M_{TenseBspline} = \mathrm{lerp}(M_{Ease}, M_{Bspline}, g_2) = \frac{1}{6}\begin{bmatrix} -g_2 & 12-9g_2 & 9g_2-12 & g_2 \\ 3g_2 & 12g_2-18 & 18-15g_2 & 0 \\ -3g_2 & 0 & 3g_2 & 0 \\ g_2 & 6-2g_2 & g_2 & 0 \end{bmatrix}. \qquad (2)$$

My excitement at discovering this family of splines was tempered by Alvy Ray Smith's revelation (about an hour later) [Smith1] that it is the subfamily of the $b$-splines with $b_1 = 1$ and $b_2 = 12(1-g_2)/g_2$. The main advantage to my formulation is that $g_2$ varies over a nicer range than $b_2$. In particular $0 \le b_2 < \infty$ corresponds to $1 \ge g_2 > 0$. The case $g_2 = 0$, which yields the Ease spline (the only interpolating spline of the family) is unattainable in the $b$-splines.

The substitution of $g_2$ for $b_2$ can be generalized to other values of $b_1$. If we substitute $b_2 = 2(b_1^3 + 2b_1^2 + 2b_1 + 1)(1-g_2)/g_2$ and $b_1 = -g_1/(g_1+1)$ into the $b$-spline matrix, we get:

$$\mathrm{lerp}(M_{Ease}, \frac{1}{6}\begin{bmatrix} g_1^3 & 1-(g_1+1)^3 & 1-g_1^3 & (g_1+1)^3 \\ -3g_1^3 & 3g_1 & 3(g_1^3-g_1-1) & 1 \\ 3g_1^3 & 3(2g_1^2+g_1+1) & -3g_1(g_1+1)^2 & 0 \\ -g_1^3 & -3g_1^2-3g_1-1) & (g_1+1)^3 & 0 \end{bmatrix}, g_2).$$

Thus, we see that for any fixed $b_1$ (equivalently $g_1$), $b_2$ (eqv. $g_2$) has the effect of picking a matrix (and therefore a curve) that is some weighted average of the Ease spline and the $b_2 = 0$ (eqv. $g_2 = 1$) spline, in the same manner that the Cardinal splines are derived from the $c = 1$ spline.

## Interpolating Splines with Tension Control

The analogy between the Cardinal splines and the $b_2$ parameter of the $b$-splines suggests looking for a generalization of the Cardinal splines that incorporates a parameter analogous to $b_1$. There at least two ways of doing this. $b_1$ expresses the ratio of the lengths of a $b$-spline's derivative vectors as we approach a knot from either side. Therefore let us consider the two interpolating splines whose first derivatives match the $c = 1$ spline's at one end, and are zero at the other end. These have the matrices

$$M_{SkewLeft} = \begin{bmatrix} 0 & 1 & -2 & 1 \\ 0 & -2 & 3 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

and

$$M_{SkewRight} = \begin{bmatrix} -1 & 2 & -1 & 0 \\ 2 & -3 & 1 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}.$$

$M_{SkewLeft}$ has zero derivative at the left end, while $M_{SkewRightt}$ has zero derivative at the right end. Thus,

$$\mathrm{lerp}(M_{SkewLeft}, M_{SkewRight}, t_1) \qquad (3)$$

is a $G^1$ family of interpolating splines. When $t_1 = .5$, formula (3) yields $M_{CatmullRom}$. Lerping this formula with the Ease matrix we get

$$\mathrm{lerp}(M_{Ease}, \mathrm{lerp}(M_{SkewLeft}, M_{SkewRight}, t_1), t_2) = \begin{bmatrix} -t_1t_2 & (t_1-1)t_2+2 & t_1t_2-2 & (1-t_1)t_2 \\ 2t_1t_2 & (1-t_1)t_2-3 & 3-2t_1t_2 & (t_1-1)t_2 \\ -t_1t_2 & 0 & t_1t_2 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}. \qquad (4)$$

This is a two parameter family of $G^1$ interpolating splines with variable tension and bias, which we will call the $t$-splines.

Instead of applying the $b$-spline bias concept directly to the Cardinal splines, we could note that the Catmull-Rom cubic spline is defined by

$$\begin{aligned} S_i(0) &= K_{i+1} \\ S_i(1) &= K_{i+2} \\ S_i'(0) &= \mathrm{lerp}(K_{i+1}-K_i, K_{i+2}-K_{i+1}, d_2) \\ S_i'(1) &= \mathrm{lerp}(K_{i+2}-K_{i+1}, K_{i+3}-K_{i+2}, d_2) \end{aligned} \qquad (5)$$

where $d_2 = .5$.

When $d_2 = 0$ and $d_2 = 1$ the matrices satisfying (5) are

$$M_{Left} = \begin{bmatrix} -1 & 2 & -1 & 0 \\ 2 & -4 & 2 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

and

$$M_{Right} = \begin{bmatrix} 0 & 1 & -2 & 1 \\ 0 & -1 & 2 & -1 \\ 0 & -1 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}.$$

respectively.

Varying $d_2$ from 0 to 1 causes the spline's derivative at $K_i$ to vary from $K_{i+1}-K_i$ to $K_{i+2}-K_{i+1}$, causing the curve to slew to the left or the right. The bias that $d_2$ causes is $C^1$ rather than $G^1$. Lerping the $d_2$ family of splines with the Ease spline produces a two parameter family of $C^1$ interpolating splines with variable tension and bias. The matrix for this family, which we will call the $d$-splines, is

$$\mathrm{lerp}(M_{Ease}, \mathrm{lerp}(M_{Left}, M_{Right}, d_1), d_2) = \begin{bmatrix} (d_1-1)d_2 & 2-d_1d_2 & (1-d_1)d_2-2 & d_1d_2 \\ 2(1-d_1)d_2 & (3d_1-1)d_2-3 & 3-d_2 & -d_1d_2 \\ (d_1-1)d_2 & (1-2d_1)d_2 & d_1d_2 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}. \qquad (6)$$

## A Five-Parameter Family of $G^1$ Local Splines

For a given cubic polynomial, $S_i(t)$ is uniquely specified if we know $S_i(0)$, $S_i(1)$, $S_i'(0)$, and $S_i'(1)$. Each of these is a linear combination of the coefficients of

$S_i(t)$, and therefore is a linear combination of the knots. If we restrict our attention to those matrix splines which are $C^0$, $G^1$ and translation invariant, how are our choices for $S_i(0)$, $S_i(1)$, $S_i'(0)$, and $S_i'(1)$ restricted? By *translation invariant*, we mean that adding some constant $D$ to each of $K_i$ has no effect on $S_i(t)$ other than to add $D$ to it everywhere[4]. That is

$$\begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} M \begin{bmatrix} K_i + D \\ K_{i+1} + D \\ K_{i+2} + D \\ K_{i+3} + D \end{bmatrix} = S_i(t) + D \ .$$

The $C^0$ condition means that $S_{i+1}(0) = S_i(1)$. This implies that $S_i(1)$ cannot depend on $K_i$, since $K_i$ is not one of the knots of $S_{i+1}(t)$. Therefore, $S_i(1)$ and $S_{i+1}(0)$ may be expressed as $aK_{i+1} + bK_{i+2} + cK_{i+3}$. This implies that $S_i(0) = aK_i + bK_{i+1} + cK_{i+2}$. Translation independence implies that

$$a(K_i + D) + b(K_{i+1} + D) + c(K_{i+2} + D) = aK_i + bK_{i+1} + cK_{i+2} + D \ .$$

Since this must be true independent of $K_i$, $K_{i+1}$, and $K_{i+2}$, it is true when $K_i = K_{i+1} = K_{i+2} = 0$, and therefore $a + b + c = 1$.

Similarly, $G^1$ implies that $S_i'(0)$ cannot depend on $K_{i+3}$ and $S_i'(1)$ cannot depend on $K_i$, and therefore, for some *d*, *e*, *f*, and *g*

$$\begin{aligned} S_i'(0) &= (1-g)(dK_i + eK_{i+1} + fK_{i+2}) \\ S_i'(1) &= g\,(dK_{i+1} + eK_{i+2} + fK_{i+3}) \end{aligned} \ .$$

Translation independence requires that $S_i'(1)$ not change when $D$ is added to the knots. Therefore

$$g\,(d(K_{i+1} + D) + e(K_{i+2} + D) + f(K_{i+3} + D)) = g\,(dK_{i+1} + eK_{i+2} + fK_{i+3})$$

and thus $d + e + f = 0$. Therefore, we can substitute

$$\begin{aligned} a &= i(1-s) \\ b &= is \\ c &= 1 - a - b = 1 - i \\ d &= t(d-1) \\ e &= t(1-2d) \\ f &= -d - e = td \end{aligned}$$

to yield

$$\begin{aligned} S_i(0) &= \mathrm{lerp}(K_{i+1}, \mathrm{lerp}(K_i, K_{i+2}, s), i) \\ S_i(1) &= \mathrm{lerp}(K_{i+2}, \mathrm{lerp}(K_{i+1}, K_{i+3}, s), i) \\ S_i'(0) &= (1-g)\,\mathrm{lerp}(0, \mathrm{lerp}(K_{i+1} - K_i, K_{i+2} - K_{i+1}, d), t) \\ S_i'(1) &= g\ \mathrm{lerp}(0, \mathrm{lerp}(K_{i+2} - K_{i+1}, K_{i+3} - K_{i+2}, d), t) \end{aligned} \ . \qquad (7)$$

It should be clear from the derivation that this family includes all the $C^0$, $G^1$, translation invariant cubic matrix splines. The intuitive functions of the five parameters are as follows:

[4] Note that *all* matrix splines are invariant under scales and rotations about the origin.

- $i$ controls how close $S_i(t)$ comes to interpolating $K_{i+1}$ and $K_{i+2}$. When $i=0$, $S_i(0)=K_{i+1}$, and $S_i(1)=K_{i+2}$. When $i=1$, $S_i(0)$ and $S_i(1)$ are independent of $K_{i+1}$ and $K_{i+2}$, respectively.
- $s$ controls how much $S_i(0)$ and $S_i(1)$ slew parallel to the lines $(K_i, K_{i+2})$ and $(K_{i+1}, K_{i+3})$. When $s=.5$, $S_i(0)$ will lie symmetrically between $K_i$ and $K_{i+2}$, as will $S_i(1)$ between $K_{i+1}$ and $K_{i+3}$.
- $g$ controls the "geometricity" of the spline. When $g=.5$, the spline will be $C^1$. For other values of $g$, the spline will be given a "kick" in the direction of its tangent as it passes through $S_i(0)$.
- $t$ controls the tension on the spline. When $t=0$, $S_i'(t)$ goes to zero as the curve passes each knot. As $t$ get larger, the curve gets less tense, passing its knots at greater speed.
- $d$ controls the direction the curve heads as it passes each knot. When $d=0$, $S_i'(0)$ heads "left", parallel to the line $(K_i, K_{i+1})$. When $d=1$, $S_i'(0)$ heads "right", parallel to $(K_{i+1}, K_{i+2})$.

The matrix for this family of splines, which we will call the M-splines[5], is

$$\begin{bmatrix} -dgt+dt+gt-2is+2i-t & 3dgt-2dt-2gt+2is-4i+t+2 & -3dgt+dt+gt+2is+2i-2 & dgt-2is \\ 2dgt-2dt-2gt+3is-3i+2t & -5dgt+4dt+3gt-3is+6i-2t-3 & 4dgt-2dt-gt-3is-3i+3 & 3is-dgt \\ t(-dg+d+g-1) & t(2dg-2d-g+1) & dt(1-g) & 0 \\ i(1-s) & 1-i & is & 0 \end{bmatrix}.$$

All the spline families we have discussed above are subfamilies of the M-splines. Our tense B-spline family has $s=.5$, $i=g_2/3$, $g=.5$, $d=.5$, and $t=2g_2$. The Cardinal splines are the subfamily with $i=0$, $g=.5$, $d=.5$, and $t=4c$. (The value of $s$ is irrelevant when $i=0$.) The $t$-splines are the class with $i=0$, $g=1-t_1$, $g=.5$, and $t=2t_2$. The $d$-splines have $i=0$, $g=.5$, $d=d_1$, and $t=2d_2$.

## Examples

Figure 1 shows several members of each of the spline families developed in this paper. These pictures were drawn by a '50s Formica boomerang design system written in Ideal [VanWyk] and Emacs. Each of the illustrations uses the same set of six knots and varies one of the parameters of the family from 0 to 1 in steps of 1/4. Curves drawn with longer dashed lines correspond to larger parameter values, with the solid curve showing the parameter set to 1.

The upper left illustration shows tensed B-splines with $g_2=0(.25)1$. The upper right shows Cardinal splines with $c=0(.25)1$. The lower left shows $t$-splines with $t_1=0(.25)1$ and $t_2=.5$. The lower right shows $d$-splines with $d_1=0(.25)1$ and $d_2=.5$.

The illustrations clearly show the effects of the tension and bias parameters. As $g_2$ or $c$ approaches zero, the curves more closely approach the polygon connecting the knots. As $t_1$ or $g_1$ varies from zero to one, the curves slew from left to right.

## Conclusions

The original observation on which this work is based is that the Cardinal spline tension parameter is equivalent to lerping the $c=1$ and Ease splines. This

[5] M is for Matrix, since this class includes most of the useful cubic matrix splines.

suggested an analogous method of applying tension to B-splines, which Alvy Ray Smith demonstrated was equivalent to the $b_2$ parameter of Barsky's $b$-splines. This in turn suggested looking for an extension of the Cardinal splines that have a bias parameter analogous to $b_1$.

There are two families of interpolating splines with bias and tension controls, one of them $G^1$ (the $t$-splines) and the other $C^1$ (the $d$-splines). We have constructed a five-parameter $G^1$ family which subsumes the $b$-, $c$-, $t$-, and $d$-splines and which exhausts the translation invariant $G^1$ cubic matrix splines.

All of this work was made easy by the matrix spline notation. Linear conditions on polynomials and their derivatives are equivalent to corresponding conditions on their coefficients (because the derivative is a linear operator). Since the coefficients of a matrix spline are linear combinations of its knots, these conditions are therefore equivalent to corresponding conditions on the matrix. This equivalence makes the proofs of continuity criteria trivial (to the point where I haven't bothered including any in this paper), and frees the imagination in its search for splines with interesting and useful properties.

## Acknowledgements

This work owes much to conversations with Alvy Ray Smith during the spring of 1983. He discovered the mapping $b_2 = 12(1-g_2)/g_2$ which shows that my tense B-splines are just a sub-class of the $b$-splines. He also encouraged the research by including parts of it in his notes for a tutorial on splines given at Siggraph '83 [Smith2]. Tom Porter's critical reading of multiple drafts of this paper aided the presentation greatly.

## Bibliography

[Barsky] Brian A Barsky, *The Beta-Spline: A Local Representation Based on Shape Parameters and Fundamental Geometric Measures*, PhD dissertation, Department of Computer Science, University of Utah, Salt Lake City, Dec 1981.

[Brewer-Anderson] J A Brewer and D C Anderson, *Visual Interaction with Overhauser Curves and Surfaces*, **Computer Graphics**, Vol 11, No 2 (Siggraph '77 Proceedings), Summer 1977, 132-137.

[Catmull] Edwin Catmull, *A Subdivision Algorithm for Computer Display of Curved Surfaces*, PhD dissertation, Department of Computer Science, University of Utah, Salt Lake City, Dec 1974.

[Catmull-Rom] Edwin Catmull and Raphael Rom, *A Class of Local Interpolating Splines*, **Computer Aided Geometric Design**, edited by Robert E Barnhill and Richard F Riesenfeld, Academic Press, San Francisco, 1974, 317-326.

[Clark] James H Clark, *Parametric Curves, Surfaces and Volumes in Computer Graphics and Computer-Aided Geometric Design*, Technical Report 221, Computer Systems Laboratory, Stanford University, Palo Alto, California, Nov 1981.

[Knuth] Donald E. Knuth, *Mathematical Typography*, **Bulletin (New Series) of the American Mathematical Society**, Mar 1979, 337-372.

[Riesenfeld] Richard Riesenfeld, *Applications of B-spline Approximation to Geometric Problems of Computer-Aided Design*, Technical Report UTEC-Csc-73-126, Department of Computer Science, University of Utah, Salt Lake City, Mar 1973.

[Smith1] Alvy Ray Smith, personal communication, Apr 1983.

[Smith2] Alvy Ray Smith, *Spline Tutorial Notes*, Technical Memo #77, Computer Graphics Project, Lucasfilm Ltd, **Introduction to Computer Animation Tutorial Notes**, Siggraph '83, Jul 1983, 64-75.

[VanWyk] Christopher J Van Wyk, *Ideal User's Manual*, Computer Science Technical Report #103, Bell Laboratories, Murray Hill, NJ, Dec 17, 1981.

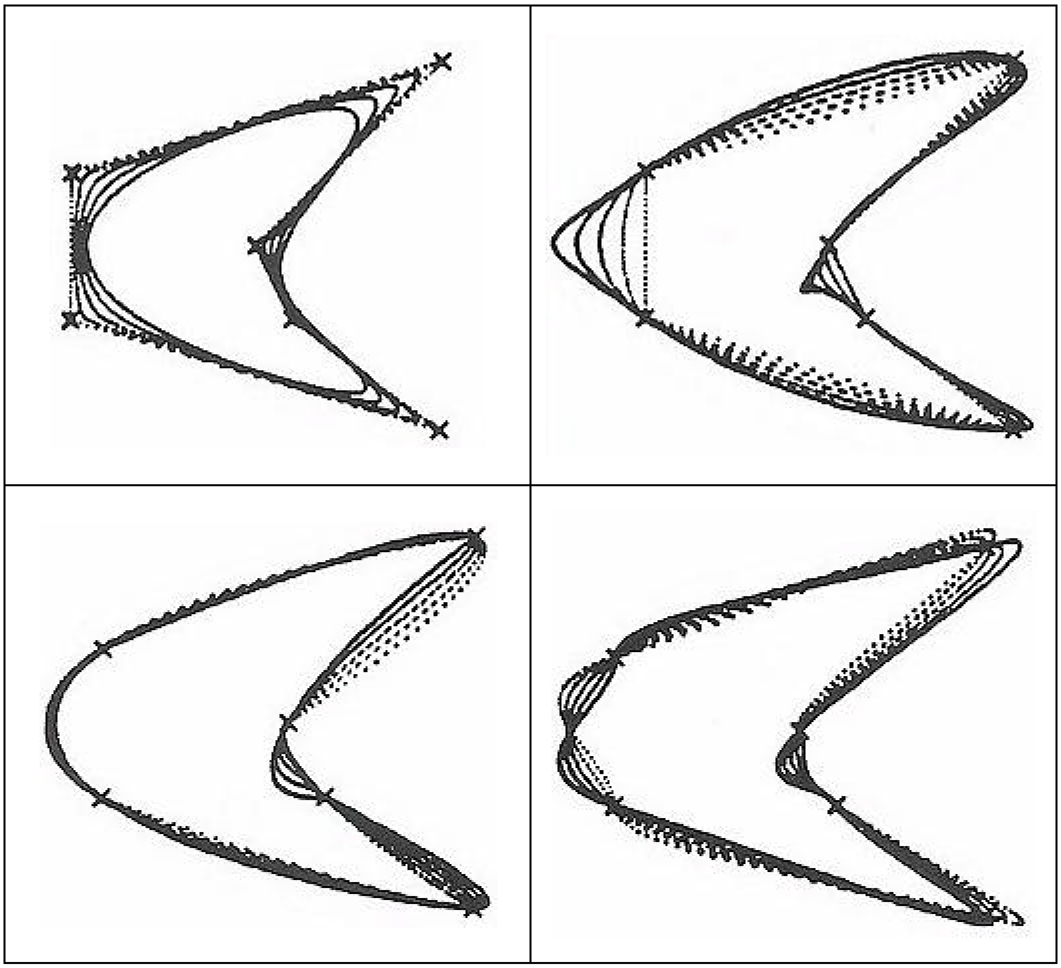

**Fig. 1—Examples of Spline Families**