

Discussed is a method of drawing curves of arbitrary shape on a graphic display screen.

An algorithm for the design of free-form curves is developed by using rational polynomials and the notation of homogeneous coordinates. With this algorithm, spline-like curves can be generated through arbitrarily placed points in a plane or in space.

INTERACTIVE GRAPHICS IN DATA PROCESSING

An algorithm for generating spline-like curves

by D. V. Ahuja

For a computer-aided design system to be truly useful, it must be possible to do freehand sketching and subsequent refining of arbitrarily shaped curves on a display screen. Such curves are vital graphic elements in the design of free-form surfaces. With a light pen and a graphic display console, the designer should be able to delineate and modify at will the curves describing his surface. The mathematical model used to generate such curves should be simple and efficient. "Response time" on the display console should be minimum so that when the curve is bent with the light pen the motion appears continuous. To enhance the importance of such curves for numerically controlled machines, it must be possible to vary the smoothness of the curve to suit various "tolerances." It should also be possible to define a curve in many ways through a set of points.

Traditionally, conics and cubics have been used for the generation of these curves.^{1,2} Coons and Herzog³ point out the drawbacks of these two families of curves for generating free-form curves and suggest the use of rational polynomials instead. It is the intent of this paper to develop a general algorithm for generating spline-like curves through arbitrarily placed points in a plane or in space using rational polynomials.⁴

notation The notation for homogeneous coordinates previously established in the paper by Ahuja and Coons that appears in this issue is used in the following derivations, i.e., a point in space (X, Y, Z) is written in homogeneous form as $wV = [wx \quad wy \quad wz \quad w]$, and a

point in a plane is $wV = [wx \quad wy \quad wz \quad w]$. The actual coordinates are obtained from wV by

$$V = wV/w = [X \quad Y \quad Z \quad 1].$$

Curve development

The ordinary coordinates of a point (X, Y, Z) can be parametrized as follows

$$X = \frac{(a_1u^3 + b_1u^2 + c_1u + d_1)}{(au^3 + bu^2 + cu + d)} \quad (1)$$

$$Y = \frac{(a_2u^3 + b_2u^2 + c_2u + d_2)}{(au^3 + bu^2 + cu + d)} \quad (2)$$

$$Z = \frac{(a_3u^3 + b_3u^2 + c_3u + d_3)}{(au^3 + bu^2 + cu + d)} \quad (3)$$

where u is the parameter and $a, b, c, d, a_1, b_1, c_1, d_1, \dots, a_3, b_3, c_3, d_3$ are constant coefficients.

Then combining Equations 1, 2, and 3 and keeping in mind that

$$X = \frac{wx}{w}, \quad Y = \frac{wy}{w}, \quad \text{and} \quad Z = \frac{wz}{w},$$

we could write

$$\begin{aligned} wV &= [wx \quad wy \quad wz \quad w] \\ &= [u^3 \quad u^2 \quad u \quad 1] \begin{bmatrix} a_1 & a_2 & a_3 & a \\ b_1 & b_2 & b_3 & b \\ c_1 & c_2 & c_3 & c \\ d_1 & d_2 & d_3 & d \end{bmatrix} \end{aligned} \quad (4)$$

Also, since $(wx) = a_1u^3 + b_1u^2 + c_1u + d_1$ then

$$(wx)' = 3a_1u^2 + 2b_1u + c_1 \quad (5)$$

similarly

$$(wy)' = 3a_2u^2 + 2b_2u + c_2 \quad (6)$$

$$(wz)' = 3a_3u^2 + 2b_3u + c_3 \quad (7)$$

$$(w)' = 3au^2 + 2bu + c \quad (8)$$

Combining Equations 5 through 8

$$\begin{aligned} (wV)' &= [(wx)' \quad (wy)' \quad (wz)' \quad w'] \\ &= [3u^2 \quad 2u \quad 1 \quad 0] \begin{bmatrix} a_1 & a_2 & a_3 & a \\ b_1 & b_2 & b_3 & b \\ c_1 & c_2 & c_3 & c \\ d_1 & d_2 & d_3 & d \end{bmatrix} \end{aligned} \quad (9)$$

(Note that the 4×4 matrix on the right sides of Equations 4 and 9 is the same.)

In general, then, if X , Y , and Z are all of the form:

$$X = \frac{f_1(u)}{f(u)}, \quad Y = \frac{f_2(u)}{f(u)}, \quad \text{and} \quad Z = \frac{f_3(u)}{f(u)}$$

where $f(u)$, $f_1(u)$, $f_2(u)$, and $f_3(u)$ are cubic polynomials of the parameter u , we could write in matrix form, as in Equations 4 and 9

$$wV = [u^3 \quad u^2 \quad u \quad 1]A \quad (10a)$$

and

$$(wV)' = [3u^2 \quad 2u \quad 1 \quad 0]A \quad (10b)$$

where A is a matrix of constant coefficients of $f(u)$, $f_1(u)$, $f_2(u)$, and $f_3(u)$. A , a 4×3 matrix for planar points or a 4×4 matrix for points in three-dimensional space, is determined later. By similar techniques the reader may also prove that

$$(wV)'' = [6u \quad 2 \quad 0 \quad 0]A \quad (10c)$$

Figure 1 The spline: a smooth curve passing through points



The problem is to fit a smooth curve that *must* pass through n points $V_1, V_2, V_3, \dots, V_n$ in the order in which they are given in Figure 1. The author proposes to do so in a piecemeal fashion, i.e., we assume that the curve consists of $n - 1$ segments of the form given by Equation 10a which when put together result in the desired curve. For computational simplicity, we set up the restriction that the parameter u assumes values between 0 and 1 in each segment. We now proceed to show how to determine the matrix A for each segment.

We assume that for the curve to be "smooth," the first and second derivative vectors at the "junctions" ($V_1, V_2, \dots, V_n$) between simple segments should be single-valued. The single-valued property of the first derivative vectors ensures slope continuity at the junctions, and the single-valued property of the second derivative vectors ensures curvature continuity at the junctions. Thus at the junction V_3 ,

$$V_{3u}^P = V_{3u}^Q \quad \text{and} \quad V_{3uu}^P = V_{3uu}^Q \quad (11)$$

where superscripts refer to the segments, and subscripts u and uu represent $\partial V / \partial u$ and $\partial^2 V / \partial u^2$.

In general, we can determine the A matrix in the following manner. Consider the first segment V_1V_2 at $u = 0$. Then from Equations 10a and 10b we have

$$w_1V_1 = [0 \quad 0 \quad 0 \quad 1]A$$

$$(w_1V_1)' = [0 \quad 0 \quad 1 \quad 0]A$$

and similarly at $u = 1$

$$w_2V_2 = [1 \quad 1 \quad 1 \quad 1]A$$

$$(w_2V_2)' = [3 \quad 2 \quad 1 \quad 0]A$$

determination of
the A matrix

or we can write in matrix form

$$\begin{bmatrix} w_1 V_1 \\ w_2 V_2 \\ (w_1 V_1)' \\ (w_2 V_2)' \end{bmatrix}_E = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{bmatrix}_B A$$

or

$$A = B^{-1}E = \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} w_1 V_1 \\ w_2 V_2 \\ w_1' V_1 + w_1 V_1' \\ w_2' V_1 + w_2 V_2' \end{bmatrix}$$

The square matrix B^{-1} is constant for all segments because it is dependent on values of u at endpoints of each simple segment which are 0 and 1, and we denote it by M . Then

$$A = M \begin{bmatrix} w_1 & 0 & 0 & 0 \\ 0 & w_2 & 0 & 0 \\ w_1' & 0 & w_1 & 0 \\ 0 & w_2' & 0 & w_2 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_1' \\ V_2' \end{bmatrix} \quad (12)$$

The matrix A , as such, is too general for all practical purposes. Any arbitrary choice of w 's results in one curve. So we restrict our studies to

$$[w_1 \ w_2 \ w_1' \ w_2'] = [1 \ 1 \ a \ b]$$

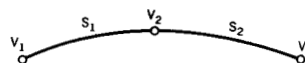
where a and b are independent quantities. For this arbitrary simplification, Equation 10a still contains ordinary parametric cubics (for $a = b = 0$), conics, and a family of rational parametric cubics for values other than $a = b = 0$. A second arbitrary simplification of the problem consists in making $(w_i^P)' = (w_i^Q)'$, i.e., w' is single-valued at each junction.

Typically for two adjacent simple segments, the first might be defined by $[w_1 \ w_2 \ w_1' \ w_2'] = [1 \ 1 \ a_1 \ a_2]$, and the second would then be defined by $[w_2 \ w_3 \ w_2' \ w_3'] = [1 \ 1 \ a_2 \ a_3]$.

With these arbitrary simplifications for each simple segment, matrix A can be evaluated by Equation 12 if the "junction tangent vectors" $V_2', V_3', \dots, V_{n-1}'$ are known. For the present, we assume that the "end" tangent vectors V_1' and V_n' are known. (The case where they are unknown is disposed of later.) However, in evaluating $V_1', V_2', \dots, V_n'$, we must satisfy the conditions imposed by Equation 11. We now show how this can be done.

First consider only two simple segments as shown in Figure 2. Since $w_1 = w_2 = w_3 = 1$, we can evaluate V_2'' for the two segments S_1 and S_2 and equate these expressions. V_2'' for segment S_1 at $w = 1$ is determined as follows:^{3,5} From Equation 10c, we have

Figure 2 Two simple segments



$$\begin{aligned}
(w_2 V_2)'' &= [6u \quad 2 \quad 0 \quad 0]A \\
&= [6u \quad 2 \quad 0 \quad 0]M \begin{bmatrix} w_1 V_1 \\ w_2 V_2 \\ w_1' V_1 + w_1 V_1' \\ w_2' V_2 + w_2 V_2' \end{bmatrix} \quad (13a)
\end{aligned}$$

However

$$(w_2 V_2)'' = w_2'' V_2 + 2w_2' V_2' + w_2 V_2''$$

Therefore

$$w_2 V_2'' = (w_2 V_2)'' - w_2'' V_2 - 2w_2' V_2' \quad (13b)$$

From 13a, on simplifying

$$(w_2 V_2)'' = 6V_1 + 2a_1 V_1 - 6V_2 + 4a_2 V_2 + 2V_1' + 4V_2'$$

Also w_2'' is the last component of the vector $(w_2 V_2)''$ and therefore

$$w_2'' = [+6 \quad -6 \quad 2 \quad 4] \begin{bmatrix} w_1 \\ w_2 \\ w_1' \\ w_2' \end{bmatrix}$$

(because the last components of V_1 , V_2 , and V_1' and V_2' are 1, 1, 0, and 0 respectively).

$$w_2'' = 6 - 6 + 2w_1' + 4w_2' = 2a_1 + 4a_2$$

Now from Equation 13b

$$\begin{aligned}
w_2 V_2'' &= 6V_1 + 2a_1 V_1 - 6V_2 + 4a_2 V_2 + 2V_1' + 4V_2' - 2a_1 V_2 \\
&\quad - 4a_2 V_2 - 2a_2 V_2' \quad (13c)
\end{aligned}$$

for $w_2 = 1$, writing Equation 13c in matrix form, we have

$$V_2'' = [1 \quad 1 \quad a_1 \quad a_2] \begin{bmatrix} 6(V_1 - V_2) + 2V_1' \\ 4V_2' \\ 2(V_1 - V_2) \\ -2V_2' \end{bmatrix}$$

which may be written as

$$V_2'' = [1 \quad 1 \quad a_1 \quad a_2] \begin{bmatrix} 6 & -6 & 2 & 0 \\ 0 & 0 & 0 & 4 \\ 2 & -2 & 0 & 0 \\ 0 & 0 & 0 & -2 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_1' \\ V_2' \end{bmatrix} \quad (14)$$

Similarly, at the beginning of segment S_2 , we can write^{3,5}

$$V_2'' = [1 \quad 1 \quad a_2 \quad a_3] \begin{bmatrix} 0 & 0 & -4 & 0 \\ -6 & 6 & 0 & -2 \\ 0 & 0 & -2 & 0 \\ 2 & -2 & 0 & 0 \end{bmatrix} \begin{bmatrix} V_2 \\ V_3 \\ \underline{V_2'} \\ \underline{V_3'} \end{bmatrix} \quad (15)$$

Note the V_2' entries (those underscored) in Equations 14 and 15. These entries ensure tangent vector continuity at the "junction" V_2 . We equate these two expressions to make V_2'' single-valued, thus ensuring curvature continuity at V_2 .

$$\begin{aligned} & [6 + 2a_1 \quad -6 - 2a_1 \quad 2 \quad 4 - 2a_2] \begin{bmatrix} V_1 \\ V_2 \\ V_1' \\ V_2' \end{bmatrix} \\ &= [-6 + 2a_3 \quad 6 - 2a_3 \quad -4 - 2a_2 \quad -2] \begin{bmatrix} V_2 \\ V_3 \\ V_2' \\ V_3' \end{bmatrix} \end{aligned}$$

The equation can be rearranged to read:

$$\begin{aligned} & [6 + 2a_1 \quad -2a_1 - 2a_3 \quad -6 + 2a_3] \begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix} \\ &= [-2 \quad -8 \quad -2] \begin{bmatrix} V_1' \\ V_2' \\ V_3' \end{bmatrix} \end{aligned}$$

simplifying

$$[1 \quad 4 \quad 1] \begin{bmatrix} V_1' \\ V_2' \\ V_3' \end{bmatrix} = [-3 - a_1 \quad a_1 + a_3 \quad 3 - a_3] \begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix} \quad (16)$$

Now let us extend Equation 16 for four points. We may write

$$\begin{bmatrix} 1 & 4 & 1 & 0 \\ 0 & 1 & 4 & 1 \end{bmatrix} \begin{bmatrix} V_1' \\ V_2' \\ V_3' \\ V_4' \end{bmatrix} = \begin{bmatrix} -3 - a_1 & a_1 + a_3 & 3 - a_3 & 0 \\ 0 & -3 - a_2 & a_2 + a_4 & 3 - a_4 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \end{bmatrix}$$

We could extend this pattern to n points as follows:

$$\begin{bmatrix} 1 & 4 & 1 & 0 & 0 & \cdot & \cdot & \cdot & \cdot & 0 \\ 0 & 1 & 4 & 1 & 0 & 0 & \cdot & \cdot & \cdot & 0 \\ 0 & 0 & 1 & 4 & 1 & 0 & \cdot & \cdot & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & 0 & \cdot & \cdot & \cdot & 1 & 4 & 1 \end{bmatrix} \begin{bmatrix} V_1' \\ V_2' \\ V_3' \\ \cdot \\ \cdot \\ \cdot \\ V_n' \end{bmatrix} \\
 = \underbrace{\begin{bmatrix} -3-a_1 & a_1+a_3 & 3-a_3 & 0 & 0 & 0 \\ 0 & -3-a_2 & a_2+a_4 & 3-a_4 & 0 & 0 \\ 0 & 0 & -3-a_3 & a_3+a_5 & 3-a_5 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & -3-a_{n-2} & a_{n-2}+a_n & 3-a_n \end{bmatrix}}_F \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ \cdot \\ \cdot \\ \cdot \\ V_n \end{bmatrix}$$

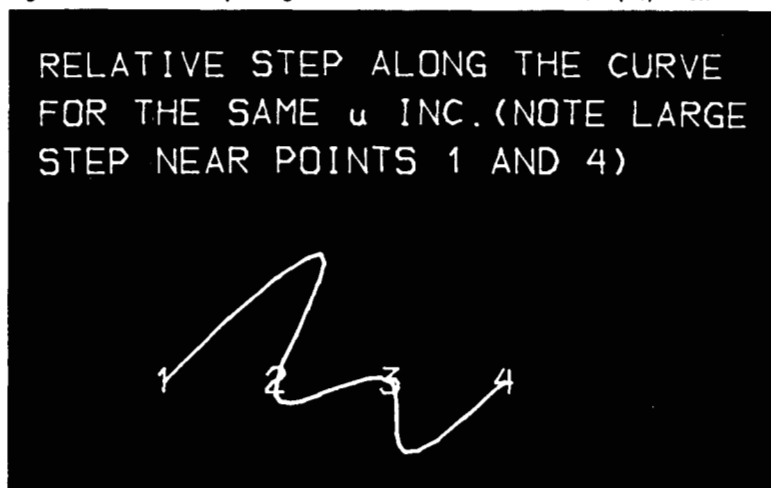
which may be simplified to

$$\begin{bmatrix} V_2' \\ V_3' \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ V_{n-1}' \end{bmatrix} = \begin{bmatrix} 4 & 1 & & & & & \\ & 1 & 4 & 1 & & & \\ & & 1 & 4 & 1 & & \\ & & \cdot & \cdot & \cdot & \cdot & \\ & & \cdot & \cdot & \cdot & \cdot & \\ & & & & & 1 & 4 \end{bmatrix}^{-1} F \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ \cdot \\ \cdot \\ \cdot \\ V_n \end{bmatrix} - \begin{bmatrix} V_1' \\ 0 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ V_n' \end{bmatrix} \quad (17)$$

We can evaluate V_2' , V_3' , $\dots$, V_{n-1}' from this equation and, hence, the A matrix for each segment from Equation 12.

Having parametrized each simple segment as in Equation 10a, we must break it into a suitable number of vectors to be able to draw it smoothly by means of the image-generation routines of the IBM 2250 or similar routines provided for the CalComp plotter and other digital machines. Evidently this is done by varying u between 0 and 1 for each segment. However, we have observed (e.g., Figure 3) that for a specified change (Δu) in the value of the parameter, a larger vector is obtained in the region of small curvature (where the rate of bending is low and a small step is not desired) and a small vector in the region of sharp curvature (where bending is sharp and a small step is desired). Thus a value of Δu , which is suitable for a region of maximum curvature in each segment, is suitable for the rest of the segment. This eliminates the iterative techniques to step along the curve. It may also be noted that if the elements of the

Figure 3 A relative step along the curve for the same u increment (Δu) = 0.1



first two rows of the matrix A for the segment are all zero, then no vectors are to be computed as the actual curve in that region is a straight line (discussed by Ahuja and Coons in another paper in this issue).

Thus we have developed a general algorithm for generating spline-like curves. But thus far, we have not considered the effect of varying $a_1, a_2, \dots, a_n$ on the behavior of the curve. This effect can be predicted mathematically. We proceed to do so and then suggest a scheme for implementing this algorithm, thereby pointing out its notable features. Let us consider the equation of the curve through one simple segment

varying the
a factors

$$wV = [u^3 \quad u^2 \quad u \quad 1]M \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ a_1 & 0 & 1 & 0 \\ 0 & a_2 & 0 & 1 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_1' \\ V_2' \end{bmatrix}$$

Differentiating with respect to a_1

$$\frac{\partial(wV)}{\partial a_1} = \frac{\partial w}{\partial a_1} V + w \frac{\partial V}{\partial a_1}$$

which gives

$$w \frac{\partial V}{\partial a_1} = \frac{\partial(wV)}{\partial a_1} - \frac{\partial w}{\partial a_1} V.$$

Now

$$\frac{\partial(wV)}{\partial a_1} = [u^3 \quad u^2 \quad u \quad 1]M \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_3' \\ V_2' \end{bmatrix} \quad (18)$$

which after simplification is equal to

$$[u^3 - 2u^2 + u]V_1$$

Also $\partial w/\partial a_1$ is the third column of Equation 18 which is

$$[u^3 - 2u + u \quad 0 \quad 0 \quad 0] \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} = u^3 - 2u^2 + u$$

$$w \frac{\partial V}{\partial a_1} = [u^3 - 2u^2 + u][V_1 - V]$$

or

$$\frac{\partial V}{\partial a_1} = \frac{1}{w} [u^3 - 2u^2 + u][V_1 - V]$$

Thus, a general point V corresponding to a value u moves along the vector $(V_1 - V)$ as a_1 is changed. The scalar $(1/w)(u^3 - 2u^2 + u)$ describes the magnitude of this motion. By a similar procedure, we can show that the point V moves along $(V_2 - V)$ as a_2 is changed. Quantities $a_1, a_2, \dots, a_n$ have a predictable effect on the shape of the curve. Their adjustment can modify the curve segment shape so as to remove or introduce inflection points at will, without disturbing the end conditions or other constraints (input points) of the curve.

implementing
the algorithm

The following scheme is suggested for implementing this algorithm. We start by making $a_1, a_2, \dots, a_n = 0$, thus generating the curve by piecemeal parametric cubics (if end tangent vectors are also unknown, we can assume them to be equal to $[0 \ 0 \ 0]$). We shall then modify the curve by changing the quantities $a_1, a_2, \dots, a_n$ as desired. The effect of the end tangent vectors on curve shape and the utilization of the a factors to modify the shape of the curve is illustrated in Figures 4 and 5. In Figure 4, for a slope of 90° , the end tangent vector at point 1 is varied from $[0 \ 1 \ 0]$ (curve 2) to $[0 \ 3 \ 0]$ (curve 3) to $[0 \ 5 \ 0]$ (curve 4). The tangent vector $[0 \ 0 \ 0]$ at both ends results in a straight line (curve 1). In Figure 5, an extreme case of cusp is picked deliberately to illustrate the usefulness of rational polynomials where parametric cubics gave a cusp; a_1 is changed from 0 to 5 to eliminate the cusp at point 2 (dotted curve).

Looking back at Equation 17, we notice that the inverse of matrix

$$\begin{bmatrix} 4 & 1 & & & \\ & 1 & 4 & 1 & \\ & \cdot & \cdot & \cdot & \cdot \\ & \cdot & \cdot & \cdot & \cdot \\ & & & 1 & 4 \end{bmatrix}$$

Figure 4 Effect of end tangent vector variation

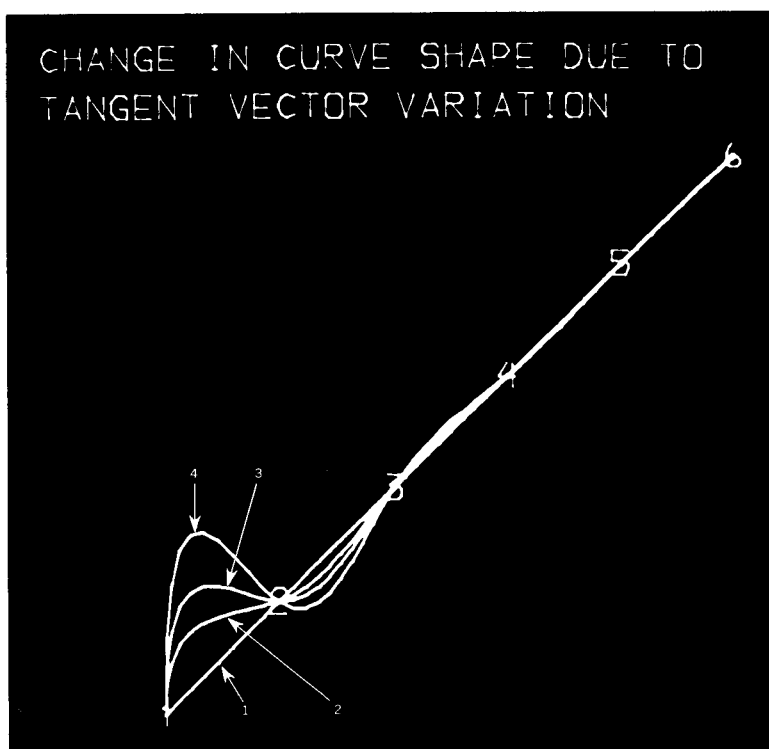


Figure 5 Effect of changing α 's on curve shape

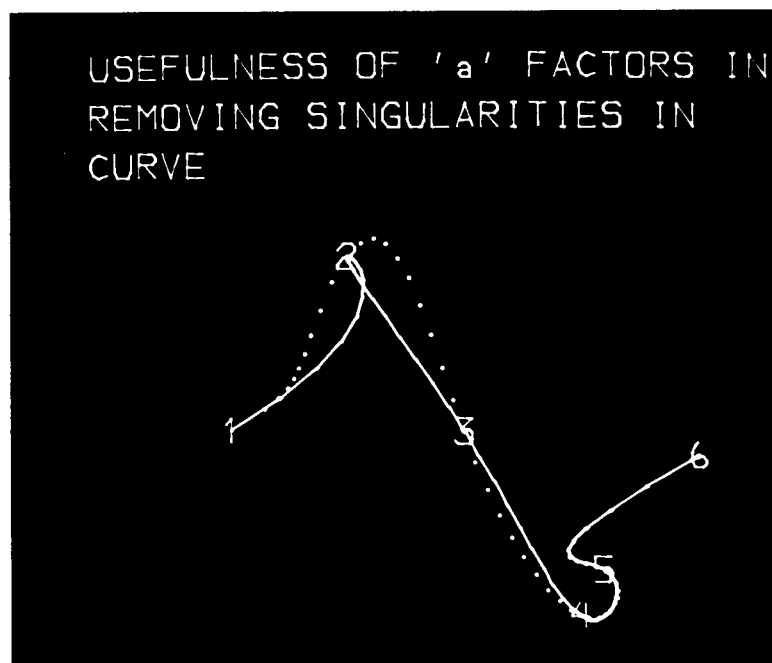


Figure 6 An "ill-conditioned" spline through numbered points

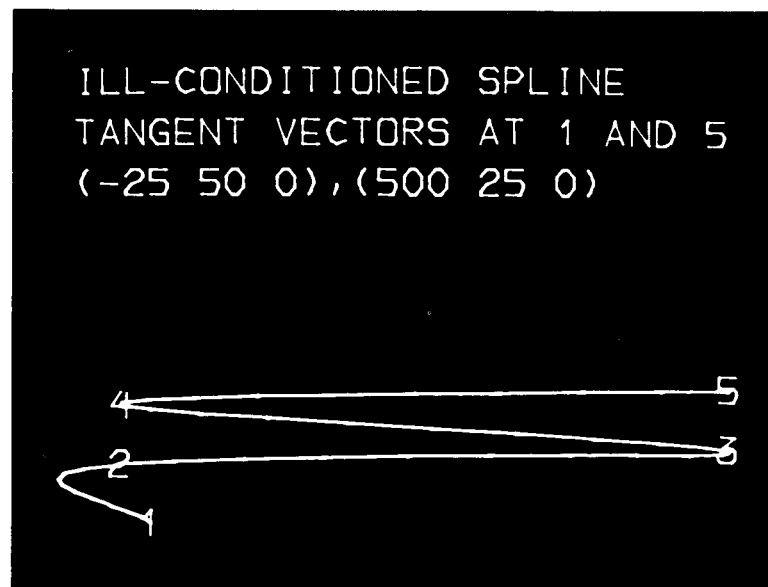
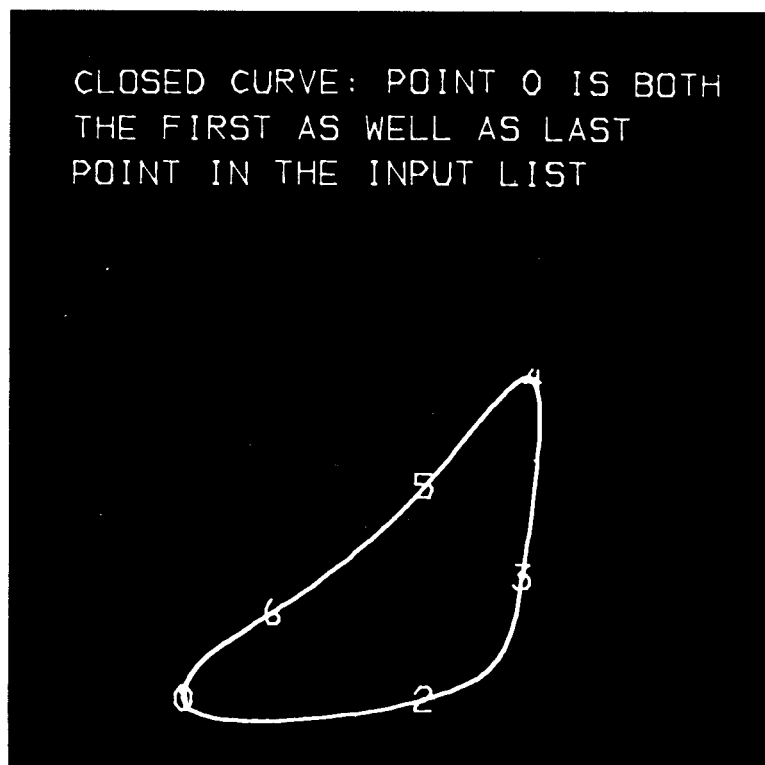


Figure 7 Closed curve through numbered points



will have to be computed only once; the changes in curve shape would be computed with a minimum of computer time. We also notice the absence of trigonometric functions, which should further improve the "response time."

Summary comments

We have developed an efficient method for generating spline-like curves. It is an easy matter to compute shape changes. Furthermore, curve shape can be defined in a number of ways such that we can have conic segments, cubic segments, and rational parametric cubic segments as part of one curve, all generated by the same equation. It is also possible to construct closed curves (and/or ill-conditioned splines) by this method as in Figures 6 and 7. We conclude this discussion by stating that further studies should be directed at the effect of $a_1, a_2, \dots, a_n$ on second derivatives and an attempt should be made to predict these changes so as to minimize the "stored energy" in the spline.

ACKNOWLEDGMENT

The author is highly indebted to Professor S. A. Coons of the Massachusetts Institute of Technology whose help and suggestions made this work possible.

CITED REFERENCES AND FOOTNOTE

1. A. H. Fowler and C. W. Wilson, *Cubic Spline, A Curve-Fitting Routine*, AEC R and D Report, Y-1400, Atomic Energy Commission, Washington, D. C. (September 1962).
2. B. Asker, "The spline curve, a smooth interpolating function used in numerical design of ship lines," *BIT* 2, 76-82 (1962).
3. S. A. Coons and B. Herzog, *Surfaces for Computer-Aided Design*, AIAA paper No. 67-895, American Institute of Aeronautics and Astronautics, New York, New York (October 1967).
4. The algorithm in this paper was developed from ideas expressed by T. E. Johnson as in his "Arbitrarily Shaped Space Curves for C. A. D.," Summer Session Course on Computer-Aided Design, Massachusetts Institute of Technology (August 1-12, 1966).
5. S. A. Coons, *Surfaces for Computer-Aided Design of Space Forms*, MAC-TR-41; Clearinghouse for Federal Scientific and Technical Information, Springfield, Virginia, 76-99 (June 1967).