A New Method of Interpolation and Smooth Curve Fitting Based on Local Procedures

HIROSHI AKIMA

ESSA Research Laboratories, Boulder, Colorado*

ABSTRACT. A new mathematical method is developed for interpolation from a given set of data points in a plane and for fitting a smooth curve to the points. This method is devised in such a way that the resultant curve will pass through the given points and will appear smooth and natural. It is based on a piecewise function composed of a set of polynomials, each of degree three, at most, and applicable to successive intervals of the given points. In this method, the slope of the curve is determined at each given point locally, and each polynomial representing a portion of the curve between a pair of given points is determined by the coordinates of and the slopes at the points. Comparison indicates that the curve obtained by this new method is closer to a manually drawn curve than those drawn by other mathematical methods.

KEY WORDS AND PHRASES: interpolation, polynomial, slope of curve, smooth curve fitting

CR CATEGORIES: 5.13

1. *Introduction*

To determine a relation between two variables, we either perform computations or make measurements. The result is given as a set of discrete data points in a plane. Knowing that the relation can be represented by a smooth curve, we next try to fit a smooth curve to the set of data points so that it will pass through all the points. Manual drawing is the most primitive method for this purpose and results in a reasonable curve if it is done by a well-trained scientist or engineer. But, since it is very tedious and time consuming, we wish to let a computer draw the curve. The computer must then be provided with necessary instructions for mathematically interpolating additional points between the given data points.

There are several mathematical methods of interpolating a single-valued function from a given set of values [3, 4, 6], but their application to curve fitting sometimes results in a curve that is very different from one drawn manually. The common difficulty is that the resultant curve sometimes shows unnatural wiggles. This seems inevitable if we make any assumption concerning the functional form for the whole set of given data points other than the continuity and the smoothness of the curve.

When we try to fit a smooth curve manually, we do not assume any functional form for the whole curve; we draw a portion of the curve based on a relatively small number of points, without taking into account the whole set of points. This local aspect is a very important feature of manual curve fitting and is the basis for our new method.

It is not easy to develop a mathematical method of smooth curve fitting based on the local procedure. If one of the existing mathematical methods is applied locally or piecewise without special consideration, the continuity of the function or

* Institute for Telecommunication Sciences

its first-order derivative at the junction points cannot be generally guaranteed. However, if we can determine the slope of the curve at each given point locally, we obtain a smooth curve by piecewise application of one of the existing mathematical methods. The osculatory interpolation method [1, 5], which is based on a piecewise function composed of a set of third-degree polynomials with slopes at the junction points locally determined, was developed along this line. But, sometimes this method also gives unnatural curves.

Recently the author proposed a method of interpolation and smooth curve fitting that is based on a piecewise function with slopes at the junction points locally determined under a geometrical condition [2]. In the present study this method is further developed with an improved condition for determining the slope of the curve.

In this paper, except in Appendix B, we describe a new method of interpolation and smooth curve fitting that is applicable to a single-valued function. However, the basic idea of our new method is also applicable to a multiple-valued function, and a method applicable to this function is outlined in Appendix B.

2. *New Method*

Our method is based on a piecewise function composed of a set of polynomials, each of degree three, at most, and applicable to successive intervals of the given points.

We assume that the slope of the curve at each given point is determined locally by the coordinates of five points, with the point in question as a center point, and two points on each side of it. This is discussed in the next section.

A polynomial of degree three representing a portion of the curve between a pair of given points is determined by the coordinates of and the slopes at the two points. This interpolation procedure is described in Section 2.2.

Since the slope of the curve must thus be determined also at the end points of the curve, estimation of two more points is necessary at each end point. This estimation procedure is described in Section 2.3.

2.1. SLOPE OF THE CURVE. With five data points 1, 2, 3, 4, and 5 given in a plane, we seek a reasonable condition for determining the slope of the curve at point 3. It seems appropriate to assume that the slope of the curve at point 3 should approach that of line segment $\overline{23}$ when the slope of $\overline{12}$ approaches that of $\overline{23}$. It is also highly desirable that the condition be invariant under a linear-scale transformation of the coordinate system. With these rather intuitive reasonings as a guideline, the condition of determining the slope is still not unique. In Appendix A, some possible conditions are discussed.

Based on the discussion given in Appendix A, we assume that the slope t of the curve at point 3 is determined by

$$t = (|m_4 - m_3| m_2 + |m_2 - m_1| m_3)/(|m_4 - m_3| + |m_2 - m_1|), \tag{1}$$

where m_1, m_2, m_3, and m_4 are the slopes of line segments $\overline{12}$, $\overline{23}$, $\overline{34}$, and $\overline{45}$, respectively. Under this condition, the slope t of the curve at point 3 depends only on the slopes of the four line segments and is independent of the interval widths. Under condition (1), $t = m_2$ when $m_1 = m_2$ and $m_3 \neq m_4$, and $t = m_3$ when $m_3 = m_4$ and $m_1 \neq m_2$, as desired. It also follows from (1) that, when $m_2 = m_3$, $t = m_2 = m_3$. Invariance of condition (1) under a linear scale transformation of the coordinate system is also obvious.

When $m_1 = m_2 \neq m_3 = m_4$, the slope t is undefined under condition (1): the slope t can take any value between m_2 and m_3 when m_1 approaches m_2 and m_4 approaches m_3 simultaneously. It is a cornerstone of our new method that $t = m_2$ when $m_1 = m_2$ and, similarly, $t = m_3$ when $m_4 = m_3$, and these two rules conflict when $m_1 = m_2 \neq m_3 = m_4$; therefore, no desired curve exists under condition (1) in this special case. (In order to give a definite unique result in all cases, the slope t is equated to $\frac{1}{2}(m_2 + m_3)$ as a convention for this case in the computer programs, which are described in Section 4. This convention is also invariant under a linear-scale transformation of the coordinate system.)

2.2. Interpolation Between a Pair of Points. We try to express a portion of the curve between a pair of consecutive data points in such a way that the curve will pass through the two points and will have at the two points the slopes determined by the procedure described in Section 2.1. To do so, we shall use a polynomial because, as stated by Milne [6], "polynomials are simple in form, can be calculated by elementary operations, are free from singular points, are unrestricted as to range of values, may be differentiated or integrated without difficulty, and the coefficients to be determined enter linearly." Since we have four conditions for determining the polynomial for an interval between two points (x_1, y_1) and (x_2, y_2), i.e.

$$y = y_1 \quad \text{and} \quad \frac{dy}{dx} = t_1 \qquad \text{at} \quad x = x_1,$$

$$y = y_2 \quad \text{and} \quad \frac{dy}{dx} = t_2 \qquad \text{at} \quad x = x_2,$$

where t_1 and t_2 are the slopes at the two points, a third-degree polynomial can be uniquely determined. Therefore, we assume that the curve between a pair of points can be expressed by a polynomial of, at most, degree three.

The polynomial, though uniquely determined, can be written in several ways. As an example we shall give the following form:

$$y = p_0 + p_1(x - x_1) + p_2(x - x_1)^2 + p_3(x - x_1)^3, \tag{2}$$

where

$$p_0 = y_1, \tag{3}$$

$$p_1 = t_1, \tag{4}$$

$$p_2 = [3(y_2 - y_1)/(x_2 - x_1) - 2t_1 - t_2]/(x_2 - x_1), \tag{5}$$

$$p_3 = [t_1 + t_2 - 2(y_2 - y_1)/(x_2 - x_1)]/(x_2 - x_1)^2. \tag{6}$$

2.3. Estimation of Two More Points at an End Point. At each end of the curve, two more points have to be estimated from the given points. We assume for this purpose that the end point (x_3, y_3) and two adjacent given points (x_2, y_2) and (x_1, y_1), together with two more points (x_4, y_4) and (x_5, y_5) to be estimated, lie on a curve expressed by

$$y = g_0 + g_1(x - x_3) + g_2(x - x_3)^2, \tag{7}$$

where the g's are constants. Assuming that

$$x_5 - x_3 = x_4 - x_2 = x_3 - x_1, \tag{8}$$

we can determine the ordinates y_4 and y_5, corresponding to x_4 and x_5, respectively, from (7). The results are

$$\begin{aligned}(y_5 - y_4)/(x_5 - x_4) - (y_4 - y_3)/(x_4 - x_3) \\ &= (y_4 - y_3)/(x_4 - x_3) - (y_3 - y_2)/(x_3 - x_2) \qquad (9) \\ &= (y_3 - y_2)/(x_3 - x_2) - (y_2 - y_1)/(x_2 - x_1).\end{aligned}$$

3. *Comparison With Some Other Methods*

Using a simple example taken from a study of waveform distortion in electronic circuits being conducted by the author, we compare our new method with four others. Assume that the values of x and y at 11 points are as follows:

x	0	1	2	3	4	5	6	7	8	9	10
$y = y(x)$	10	10	10	10	10	10	10.5	15	50	60	85

Knowing from the physical nature of the phenomena that $y(x)$ is a single-valued smooth function of x, we try to interpolate the values of $y(x)$ and to fit a smooth curve to the given set of data points.

First we apply the method of interpolation based on polynomials [4, 6]. This method is, perhaps, the one most often used. The result obtained by applying the tenth-degree polynomial is shown in Figure 1(A).

Second we use another well-known method based on the Fourier series (see [4, Sec. 9.3]). In applying this method to our data, we assume that the whole range of x from 0 to 10 corresponds to one-half of the fundamental period from 0 to π and apply a series of cosine functions up to the tenth-order harmonic term. The result is shown in Figure 1(B).

The third method is based on a spline function [3]. The spline function of degree n is a piecewise function composed of a set of polynomials, each of degree n, at most, and applicable to successive intervals of the given data points. All the polynomials are determined as a set, so that the function and its derivatives of order $1, 2, \cdots, n - 1$ are continuous in the whole range of x. Note that, although the spline function of degree three is somewhat similar to our method, no individual polynomial can be determined locally in the spline function. The result of applying the third-degree spline function is shown in Figure 1(C).

Next we try to apply the osculatory interpolation method [1, 5]. It is, like our method, based on a piecewise function composed of a set of third-degree polynomials, each applicable to successive intervals of the given points, with the slopes at the given points locally determined. The only difference between this method and ours is the manner of determining the slopes at the given points. In the osculatory interpolation, the determination of the slope involves only three points, i.e. the data point in question as a center point and two neighboring data points. It is assumed that the slope of the desired curve at the data point is equal to the slope at the same point of the curve of the second-degree polynomial passing through the three points involved. From the three data points given, (x_1, y_1), (x_2, y_2), and (x_3, y_3), the slope t at the center point (x_2, y_2) is determined by

$$\begin{aligned}t &= [(x_2 - x_1)^2(y_3 - y_2) + (x_3 - x_2)^2(y_2 - y_1)]/ \\ &\qquad [(x_2 - x_1)^2(x_3 - x_2) + (x_3 - x_2)^2(x_2 - x_1)] \qquad (10) \\ &= [(x_3 - x_2)m_1 + (x_2 - x_1)m_2]/[(x_3 - x_2) + (x_2 - x_1)],\end{aligned}$$

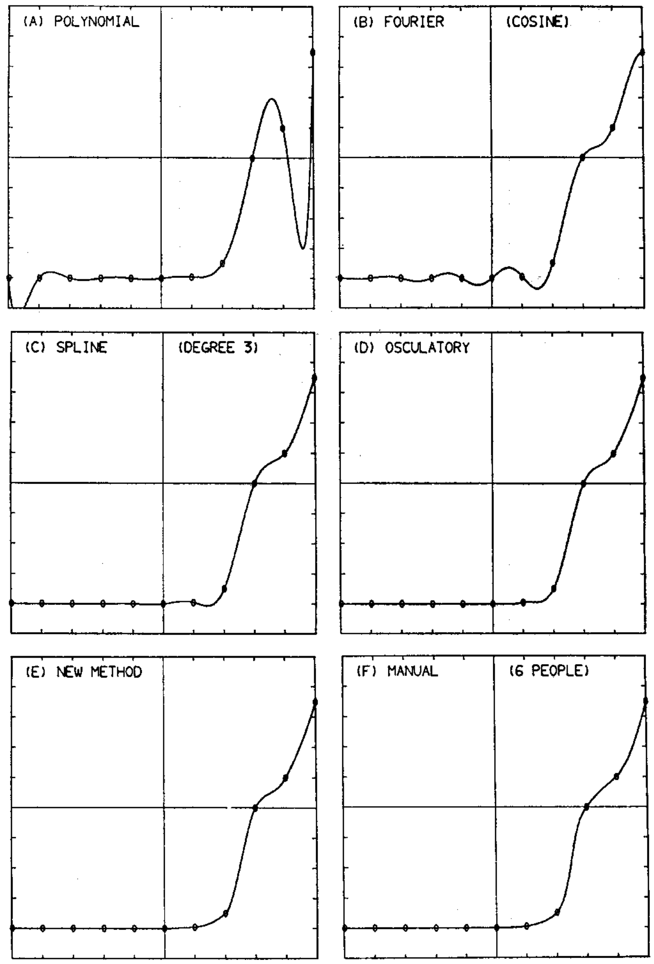

FIG. 1. Comparison of several methods of smooth curve fitting. (Encircled points are given data points.)

where m_1 and m_2 are the slopes of line segments $\overline{12}$ and $\overline{23}$, respectively. The result of applying this method is shown in Figure 1(D).

Finally, we apply our new method, with the result shown in Figure 1(E).

In addition to these results of mathematical methods, the curve obtained manually is shown in Figure 1(F); it is the average of curves drawn manually by six scientists and engineers.

Comparison of the curves in Figure 1 (A)–(F) indicates that the first two methods are definitely unsuitable for the example given. Although the curves obtained by the spline function method and the osculatory interpolation method, shown in Figure 1 (C) and (D), respectively, resemble the one obtained manually, shown in (F), they have maxima and minima that are absent in the manually drawn curve. The curve obtained by our new method, shown in Figure 1(E), is closer than the other curves to the manually drawn curve in (F).

In Figure 2 (A)–(F), we further compare our new method with the spline function method and the osculatory interpolation method with the same y values as

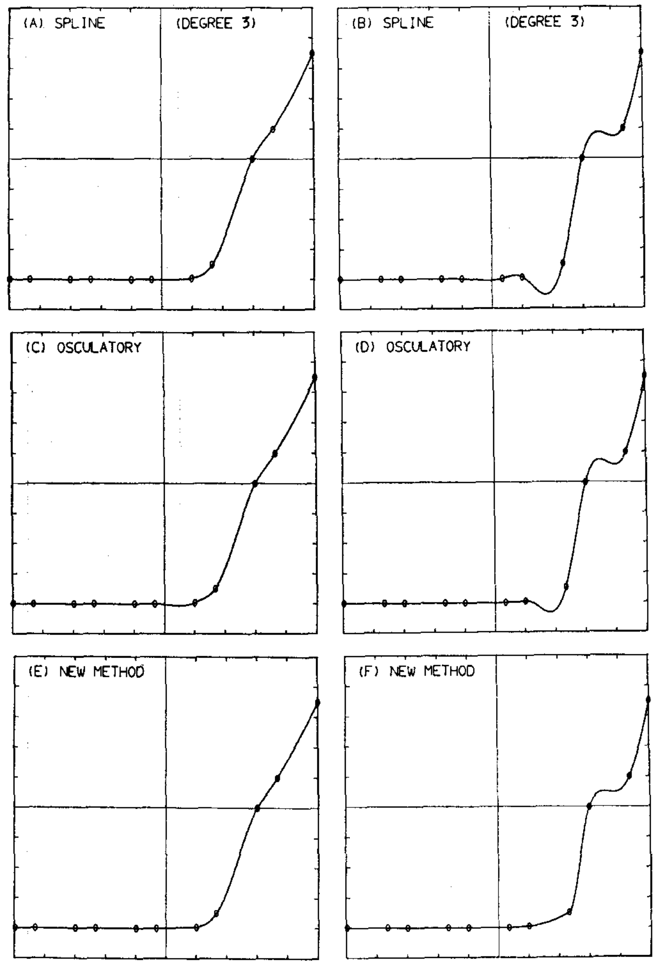

FIG. 2. Further comparison of three methods when the data points are given at unequal intervals. (Encircled points are given data points.)

in Figure 1 (A)–(F), but with different x values. In Figure 2 (A), (C), and (E), the x values are 0, 1, 3, 4, 6, $\cdots$, 13, and 15. They are 0, 2, 3, 5, 6, $\cdots$, 14, and 15 in Figure 2 (B), (D), and (F). The comparison indicates that, even when the data points are given at unequal intervals, our method performs as well as or better than the other two methods.

4. *Computer Applications*

Our new method is further compared with the spline function method and the osculatory interpolation method from the standpoint of its computer applications.

Two types of computer subroutines were programmed for each method, both in the CDC-3800 FORTRAN: one for interpolation and the other for smooth curve fitting. The interpolation subroutine interpolates, from a given set of coordinates of data points and for a given set of abscissa values of desired points, ordinate values of the desired points. The curve fitting subroutine equally divides each interval between a pair of given points, interpolates an ordinate value for each dividing point, and generates a new set of points consisting of the given data points and the interpolated points. For the spline function method, the FORTRAN program given in [3] was used after slight modifications—modifications made to save both program length and computation time for fair comparison.

4.1. PROGRAM LENGTHS. The interpolation and smooth curve fitting subroutines based on our new method occupy 339 and 324 locations, respectively, compared with 245 and 276 locations occupied by the same subroutines based on the osculatory interpolation method. Program lengths of the same subroutines based on the spline function method depend on the maximum number of given data points $L_{\max}$ that can be processed by the subroutines. The subroutines, based on a third-degree spline function, occupy $270 + 3L_{\max}$ and $293 + 3L_{\max}$ locations.

In each application, program length for our method is longer than that for the osculatory interpolation method. Comparison between our method and the spline function method depends on $L_{\max}$; program lengths for these two methods are nearly equal for interpolation and smooth curve fitting at $L_{\max} = 25$ and 10, respectively.

4.2. COMPUTATION TIMES. Each subroutine was run many times on the CDC-3800 computer, and the running time was measured by the internal clock. The subroutines based on the spline function method were run with the error tolerance in iterative solution of the equations for the second derivative of the spline function of 10^{-6} and the input data scaled in several ways, and the averages of the running times were taken. The results are shown in Tables I and II for interpolation and smooth curve fitting, respectively. For simplicity, we denote the number of given data points for both interpolation and curve fitting by L, the number of desired points for interpolation by N, and the number of divisions in each interval for smooth curve fitting by M.

In each application, the time required by the osculatory interpolation is the shortest for a given combination of L and N or L and M. For interpolation, comparison between our new method and the spline function depends on the combination of L and N and also on the way the abscissa values of the desired points are given. Table I indicates that our new method almost always requires less time than the spline function method when the abscissa values of the desired points are given in an ascending order. For smooth curve fitting, the new method always requires less time than the spline function method.

5. *Concluding Remarks*

We have described a new mathematical method of interpolation and smooth curve fitting. For proper application of our new method, the following remarks seem pertinent.

(1) Since the curve obtained by our method passes through all the given points, the method is applicable only when the precise values of the coordinates of the data points are given. All experimental data have some errors in them, and unless the errors are negligible it is more appropriate to smooth the data, i.e. to fit a curve approximating the data appropriately, than to fit a curve passing through all the points.

TABLE I. COMPARISON OF COMPUTATION TIMES OF INTERPOLATION BY VARIOUS METHODS. (L = NUMBER OF GIVEN DATA POINTS AND N = NUMBER OF DESIRED POINTS.)

L	N	Computation times (msec)		
		Spline	*Osculatory*	*New method*
	1	3.0	0.8	1.0
	10*	4.6	2.0	2.6
5	10**	4.7	2.3	3.3
	100*	17	12	16
	100**	21	15	26
	1	7.2	0.8	1.1
	10*	9.2	3.0	3.9
10	10**	9.6	3.6	5.2
	100*	24	17	21
	100**	30	27	46
	1	20	0.9	1.2
	10*	23	3.5	4.4
20	10**	23	4.3	5.8
	100*	42	22	27
	100**	48	33	55
	1	46	1.0	1.2
	10*	49	4.6	6.3
40	10**	50	5.4	7.6
	100*	75	32	36
	100**	81	47	67
	1	120	1.0	1.2
	10*	125	5.3	7.0
100	10**	125	5.8	8.0
	100*	160	41	49
	100**	160	53	74

* When abscissa values of the desired points are given in ascending order.
** When those values are given in random order.

(2) As is true for any method of interpolation, the accuracy of the interpolation cannot be guaranteed, unless the method in question has been checked in advance against precise values or a functional form.

(3) Our method yields a smooth, natural-looking curve and is therefore useful in cases where manual, but tedious, curve fitting will do in principle.

(4) The resultant curve of our method is invariant under a linear-scale transformation of the coordinate system. In other words, different scalings of the coordinates result in equivalent curves.

(5) Our method is nonlinear. In other words, if $y_i = y_i' + y_i''$ for all i, the interpolated values do not, in general, satisfy $y(x) = y'(x) + y''(x)$.

(6) Our method gives exact results when y is a second-degree polynomial of x, provided the abscissas of the data points are equally spaced.

(7) Our method requires only straightforward procedures, not iterative solutions of equations with preassigned error tolerances, which are required by some methods. No problem concerning computational stability or convergence exists in application of our method.

(8) Our method can be implemented as computer subroutines with reasonable

TABLE II. COMPARISON OF COMPUTATION TIMES OF SMOOTH CURVE FITTING BY VARIOUS METHODS. (L = NUMBER OF GIVEN DATA POINTS AND M = NUMBER OF DIVISIONS IN EACH INTERVAL BETWEEN A PAIR OF SUCCESSIVE DATA POINTS.)

L	M	*Computation Times (msec)*		
		Spline	*Osculatory*	*New method*
	2	3.3	1.5	2.0
5	10	4.9	3.0	3.6
	100	21	20	21
	2	8.0	2.8	3.7
10	10	12	6.1	6.9
	100	50	44	46
	2	22	5.2	7.0
20	10	29	12	14
	100	110	93	95
	2	50	10	13
40	10	64	25	28
	100	230	190	200
	2	130	25	33
100	10	170	63	71
	100	590	480	500

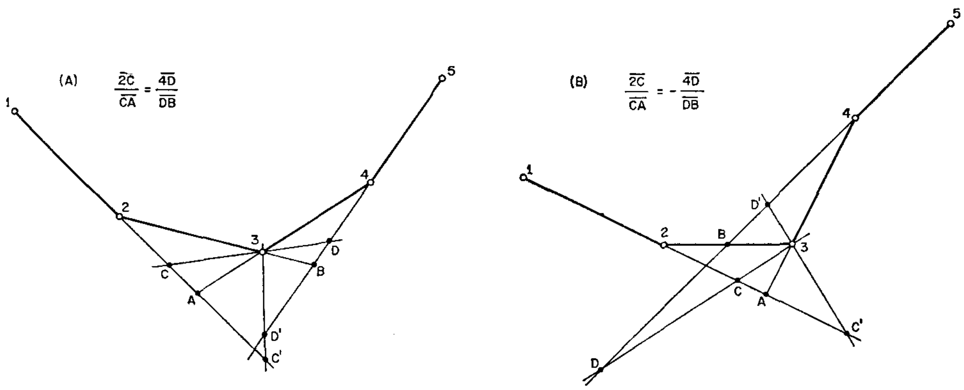

FIG. 3. A condition for determining the slope of the tangent to the curve, previously proposed [2]

program length. There is no drawback in computation time compared with typical existing mathematical methods.

(9) Our method, as reported on in the text of this paper, is applicable only to a single-valued function, but the basic idea of our new method is also applicable to the case of a multiple-valued function. A modified method for this case is outlined in Appendix B.

Appendix A. A Supplementary Note on the Determination of the Slope of the Curve

With five data points 1, 2, 3, 4, and 5, as shown in Figure 3, the author [2] previously proposed a condition that the slope of line segment $\overline{CD}$, tangent to the curve at point 3, is determined by

$$|\overline{2C}/\overline{CA}| = |\overline{4D}/\overline{DB}|, \tag{11}$$

where the point of intersection of the two straight lines extended from line segments $\overline{12}$ and $\overline{34}$ is denoted by A, the same point corresponding to line segments $\overline{23}$ and $\overline{45}$ by B, and the points of intersection of the tangent with two straight lines extended from $\overline{12}$ and $\overline{45}$ are denoted by C and D, respectively. The ratio of two line segments, which is used when the two line segments are on a straight line, is used here to symbolically represent the ratio of the lengths of the two line segments with a plus or minus sign depending on whether these two line segments have the same sense or not. An analytical expression of the slope of $\overline{CD}$ under condition (11) will be derived in the next paragraph.

Let the coordinates of points 1, 2, 3, 4, 5, A, B, C, and D in Figure 3 be denoted by (x, y) with subscripts 1, 2, 3, 4, 5, a, b, c, and d, respectively. For simplicity, we define

$$a_i = x_{i+1} - x_i \qquad (i = 1, 2, 3, 4), \tag{12}$$

$$b_i = y_{i+1} - y_i \qquad (i = 1, 2, 3, 4). \tag{13}$$

We denote the slopes of $\overline{12}$, $\overline{23}$, $\overline{34}$, $\overline{45}$, and $\overline{CD}$ by m_1, m_2, m_3, m_4, and t, respectively, i.e.

$$m_i = (y_{i+1} - y_i)/(x_{i+1} - x_i) = b_i/a_i \qquad (i = 1, 2, 3, 4), \tag{14}$$

$$t = (y_d - y_c)/(x_d - x_c). \tag{15}$$

Then, it is clear from Figure 3 that the following equations should hold:

$$(y_a - y_2)/(x_a - x_2) = (y_c - y_2)/(x_c - x_2) = b_1/a_1, \tag{16}$$

$$(y_4 - y_b)/(x_4 - x_b) = (y_4 - y_d)/(x_4 - x_d) = b_4/a_4, \tag{17}$$

$$(y_3 - y_a)/(x_3 - x_a) = b_3/a_3, \tag{18}$$

$$(y_b - y_3)/(x_b - x_3) = b_2/a_2, \tag{19}$$

$$(y_3 - y_c)/(x_3 - x_c) = (y_d - y_3)/(x_d - x_3) = t. \tag{20}$$

Using (12) and (13), we have, from (16),

$$[(y_3 - y_a) - b_2]/[(x_3 - x_a) - a_2] = b_1/a_1. \tag{21}$$

Eliminating $(y_3 - y_a)$ from (18) and (21), we obtain

$$(x_3 - x_a)/a_3 = (a_1b_2 - a_2b_1)/(a_1b_3 - a_3b_1). \tag{22}$$

Following the same procedure as in obtaining (22) from (16) and (18), we obtain

$$(x_b - x_3)/a_2 = (a_3b_4 - a_4b_3)/(a_2b_4 - a_4b_2), \tag{23}$$

$$x_3 - x_c = (a_1b_2 - a_2b_1)/(a_1t - b_1), \tag{24}$$

$$x_d - x_3 = (a_3b_4 - a_4b_3)/(b_4 - a_4t). \tag{25}$$

Since condition (11) can be written as

$$|[(x_3 - x_c) - a_2]/[(x_3 - x_a) - (x_3 - x_c)]| = |[a_3 - (x_d - x_3)]/[(x_d - x_3) - (x_b - x_3)]|, \quad (26)$$

we obtain, by eliminating $(x_3 - x_a)$, $(x_b - x_3)$, $(x_3 - x_c)$, and $(x_d - x_3)$ from (22)–(26), a quadratic equation of the form

$$|S_{12}S_{24}|(a_3t - b_3)^2 = |S_{13}S_{34}|(a_2t - b_2)^2, \quad (27)$$

where

$$S_{ij} = a_ib_j - a_jb_i \qquad (i \neq j). \quad (28)$$

From the requirement that points 2 and 4 must lie on the same side of the tangent to the curve at point 3, it follows that

$$(a_2t - b_2)(a_3t - b_3) \leq 0. \quad (29)$$

Therefore, we have

$$|S_{12}S_{24}|^{\frac{1}{2}}(b_3 - a_3t) = |S_{13}S_{34}|^{\frac{1}{2}}(a_2t - b_2). \quad (30)$$

Solving this equation, we obtain

$$t = (w_2b_2 + w_3b_3)/(w_2a_2 + w_3a_3), \quad (31)$$

where

$$w_2 = |S_{13}S_{34}|^{\frac{1}{2}}, \quad (32)$$

$$w_3 = |S_{12}S_{24}|^{\frac{1}{2}}. \quad (33)$$

This is an analytical expression of the slope of $\overline{\mathrm{CD}}$ determined under condition (11).

The relations (31)–(33) can also be expressed as

$$t = (w_2'm_2 + w_3'm_3)/(w_2' + w_3'), \quad (34)$$

where

$$w_2' = (\text{sign of } a_2)\,|(m_3 - m_1)(m_4 - m_3)|^{\frac{1}{2}}, \quad (35)$$

$$w_3' = (\text{sign of } a_3)\,|(m_2 - m_1)(m_4 - m_2)|^{\frac{1}{2}}. \quad (36)$$

Note that, in the case of a single-valued function, we can always make a_2 and a_3 positive and simplify (35) and (36).

Since condition (11) is a purely geometrical one, it is invariant under a linear transformation of the coordinate system, which includes a linear scale transformation and a rotation. It follows from (34)–(36) and also from the geometrical construction dictated by (11), that the slope t depends only on the slopes of four secants, i.e. on the quantities m_1, m_2, m_3, and m_4, and is independent of the interval widths. It is also clear from (34)–(36) that $m_1 = m_2$, $m_3 \neq m_1$, and $m_4 \neq m_3$ implies $t = m_1 = m_2$, and similarly that $m_3 = m_4$, $m_1 \neq m_2$, and $m_4 \neq m_2$ implies $t = m_3 = m_4$: these are highly desirable properties. When $m_1 = m_2 = m_3$ or when $m_2 = m_3 = m_4$, t is undefined by (34); however, this difficulty can easily be settled by taking $t = m_2 = m_3$ for such cases as a natural and logical extension.

However, despite its desirable properties, condition (11) has a serious drawback.

It follows from (34)–(36) that $t = m_2$ when $m_2 = m_4$, $m_3 \neq m_1$, and $m_4 \neq m_3$, and similarly that $t = m_3$ when $m_3 = m_1$, $m_2 \neq m_1$, and $m_4 \neq m_2$. These properties are by no means desirable or to be expected.

The simplest way to eliminate these undesirable properties for the case of a single-valued function is, perhaps, to modify weighting coefficients w_2' and w_3' in (34) to read

$$w_2' = | m_4 - m_3 |, \tag{37}$$

$$w_3' = | m_2 - m_1 |, \tag{38}$$

instead of (35) and (36), respectively. By this modification, most of the desirable properties of condition (11) are retained. The only exception is that the property of invariance under a rotation of the coordinate system is lost. But the requirement for invariance under a rotation is immaterial in the case of interpolation of a single-valued function. Equation (34) with w_2' and w_3' defined by (37) and (38), respectively, is used in the text of this paper as the equation for determining the slope. This is equivalently given in the text as condition (1).

In the case of a multiple-valued function, on the other hand, the property of invariance under a rotation of the coordinate system is essential. While retaining this property, the undesirable properties of (11) can be eliminated by modifying w_2 and w_3 in (31) to read

$$w_2 = | S_{34} |, \tag{39}$$

$$w_3 = | S_{12} |, \tag{40}$$

instead of (32) and (33), respectively. Invariance under a rotation by using (31) with these modified w_2 and w_3 can be realized from the fact that $| S_{ij} |$ is an invariant quantity representing twice the area of the triangle bounded by the vectors (a_i, b_i) and (a_j, b_j), and that (31) can be written in the form

$$S_{2t}/S_{t3} = | S_{12} | / | S_{34} | , \tag{41}$$

if a_t and b_t are chosen so that $t = b_t/a_t$ and S_{2t} and S_{t3} are defined by

$$S_{2t} = a_2 b_t - a_t b_2 , \tag{42}$$

$$S_{t3} = a_t b_3 - a_3 b_t , \tag{43}$$

respectively, as S_{ij} is defined by (28). These modified w coefficients in (39) and (40) correspond to

$$w_2' = a_2 | a_3 a_4 (m_4 - m_3) |, \tag{44}$$

$$w_3' = a_3 | a_1 a_2 (m_2 - m_1) |. \tag{45}$$

It follows from (44) and (45) that another property of dependence of t on m_1, m_2, m_3, and m_4 only is lost, but this property is not important in the case of a multiple-valued function. Equation (31) with the weighting coefficients defined by (39) and (40) is used in Appendix B.

Appendix B. An Outline of the Method of Interpolation for a Multiple-Valued Function

As described in Appendix A, (31) with (39) and (40) can be used in principle to determine the slope of the curve in the case of a multiple-valued function. To present the method in a form that is ready to be implemented in a computer program, however, it is necessary to rewrite all the equations without a ratio of the increment in y to that in x, such as m_i or t in Appendix A. Since (39) and (40) are given in terms of the increments in the x and y directions, we have only to rewrite and eliminate t from (31). This can be done by the use of $\cos\theta$ and $\sin\theta$ instead of $t = \tan\theta$, where θ is the angle of the tangent to the curve measured from the x axis. The results are

$$\cos\theta = a_0(a_0^2 + b_0^2)^{-\frac{1}{2}}, \tag{46}$$

$$\sin\theta = b_0(a_0^2 + b_0^2)^{-\frac{1}{2}}. \tag{47}$$

where

$$a_0 = w_2a_2 + w_3a_3\,, \tag{48}$$

$$b_0 = w_2b_2 + w_3b_3\,, \tag{49}$$

$$w_2 = |S_{34}| = |a_3b_4 - a_4b_3|, \tag{50}$$

$$w_3 = |S_{12}| = |a_1b_2 - a_2b_1|, \tag{51}$$

$$a_i = x_{i+1} - x_i \qquad (i = 1, 2, 3, 4), \tag{52}$$

$$b_i = y_{i+1} - y_i \qquad (i = 1, 2, 3, 4). \tag{53}$$

We assume that the curve between a pair of data points (x_1, y_1) and (x_2, y_2) can be expressed by

$$x = p_0 + p_1z + p_2z^2 + p_3z^3, \tag{54}$$

$$y = q_0 + q_1z + q_2z^2 + q_3z^3, \tag{55}$$

where the p's and q's are constants, and z is a parameter that varies from 0 to 1 as the curve is traversed from (x_1, y_1) to (x_2, y_2). Since the coordinates of the two points (x_1, y_1) and (x_2, y_2), as well as the direction of the curve $(\cos\theta_1, \sin\theta_1)$ and $(\cos\theta_2, \sin\theta_2)$ at these points, are given, we further assume that x and y satisfy the condition

$$x = x_1\,,\quad y = y_1\,,\quad dx/dz = r\cos\theta_1\,,\quad \text{and}\quad dy/dz = r\sin\theta_1 \quad \text{at}\quad z = 0,$$

$$x = x_2\,,\quad y = y_2\,,\quad dx/dz = r\cos\theta_2\,,\quad \text{and}\quad dy/dz = r\sin\theta_2 \quad \text{at}\quad z = 1,$$

where

$$r = [(x_2 - x_1)^2 + (y_2 - y_1)^2]^{\frac{1}{2}}. \tag{56}$$

From these conditions we can uniquely determine the p and q constants. The results are

$$p_0 = x_1\,, \tag{57}$$

$$p_1 = r\cos\theta_1\,, \tag{58}$$

$$p_2 = 3\,(x_2 - x_1) - r\,(\cos\theta_2 + 2\cos\theta_1), \tag{59}$$

$$p_3 = -2\,(x_2 - x_1) + r\,(\cos\theta_2 + \cos\theta_1), \tag{60}$$

$$q_0 = y_1\,, \tag{61}$$

$$q_1 = r\sin\theta_1\,, \tag{62}$$

$$q_2 = 3\,(y_2 - y_1) - r\,(\sin\theta_2 + 2\sin\theta_1), \tag{63}$$

$$q_3 = -2\,(y_2 - y_1) + r\,(\sin\theta_2 + \sin\theta_1). \tag{64}$$

Except for the case of a closed curve, estimation of two more points from the data points is required at each end of the curve. We assume for this purpose that the end point $(x_3\,, y_3)$ and two adjacent given points $(x_2\,, y_2)$ and $(x_1\,, y_1)$, together with two more points $(x_4\,, y_4)$ and $(x_5\,, y_5)$ to be estimated, lie on a curve expressed by

$$x = g_0 + g_1 z + g_2 z^2, \tag{65}$$

$$y = h_0 + h_1 z + h_2 z^2, \tag{66}$$

where the g's and h's are constants and z is a parameter. Assuming that

$$x = x_i \quad \text{and} \quad y = y_i \quad \text{at} \quad z = i \qquad (i = 1, 2, 3, 4, 5),$$

we can determine the g and h constants and, consequently, also the coordinates $(x_4\,, y_4)$ and $(x_5\,, y_5)$. The results can be expressed as

$$(x_5 - x_4) - (x_4 - x_3) = (x_4 - x_3) - (x_3 - x_2) = (x_3 - x_2) - (x_2 - x_1), \tag{67}$$

$$(y_5 - y_4) - (y_4 - y_3) = (y_4 - y_3) - (y_3 - y_2) = (y_3 - y_2) - (y_2 - y_1). \tag{68}$$

The interpolated curve is invariant under a rotation of the coordinate system but variant under a linear-scale transformation. Therefore, both the abscissa and the ordinate should be scaled with their respective units having an equal actual length on the graph.

ACKNOWLEDGMENTS. The author expresses his deep appreciation to E. L. Crow, R. K. Rosich, J. S. Washburn, and R. E. Wilkerson of the Institute for Telecommunication Sciences, ESSA Research Laboratories, for their helpful discussions.

REFERENCES

1. ACKLAND, T. G. On osculatory interpolation, where the given values of the function are at unequal intervals. *J. Inst. Actuar. 49* (1915), 369–375.
2. AKIMA, H. A method of smooth curve fitting. ESSA Tech. Rep. ERL 101-ITS 73. U S Government Printing Office, Washington, D. C., Jan. 1969.
3. GREVILLE, T. N. E. Spline functions, interpolation, and numerical quadrature. In *Mathematical Methods for Digital Computers, Vol. 2*, A. Ralston and H. S. Wilf (Eds.), Wiley, New York, 1967, Ch. 8.
4. HILDEBRAND, F. B. *Introduction to Numerical Analysis*. McGraw-Hill, New York, 1956, Ch. 2, 3, 4, and 9.
5. KARUP, J. On a new mechanical method of graduation. In *Transactions of the Second International Actuarial Congress*. C. and E. Layton, London, 1899, pp. 78–109.
6. MILNE, W. E. *Numerical Calculus*. Princeton U. Press, Princeton, N. J., 1949, Ch. III.

RECEIVED DECEMBER, 1969; REVISED APRIL, 1970