NTIA Report 86-208

A Method of Univariate Interpolation That Has the Accuracy of a Third-Degree Polynomial

Hiroshi Akima

U.S. DEPARTMENT OF COMMERCE
Malcolm Baldrige, Secretary

Alfred C. Sikes, Assistant Secretary
for Communications and Information

November 1986

TABLE OF CONTENTS

LIST OF FIGURES

A METHOD OF UNIVARIATE INTERPOLATION THAT HAS THE ACCURACY OF A THIRD-DEGREE POLYNOMIAL

Hiroshi Akima

A method of interpolation that accurately interpolates data values that satisfy a function is said to have the accuracy of that function. The desired or required properties for a univariate interpolation method are reviewed, and the accuracy of a third-degree polynomial is found to be one of the desirable properties. A method of univariate interpolation having the accuracy of a third-degree polynomial while retaining the desirable properties of the method developed earlier by Akima (J. ACM 17, pp. 589-602, 1970) has been developed. The newly developed 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 data points. The method estimates the first derivative of the interpolating function (or the slope of the curve) at each given data point from the coordinates of seven data points. The resultant curve looks natural in many cases when the method is applied to curve fitting. The method is presented with examples. Possible use of a higher-degree polynomial in each interval is also examined.

Key words: curve fitting; interpolation; polynomial; second-degree polynomial; third-degree polynomial; univariate interpolation

1. INTRODUCTION

Interpolation is a mathematical procedure for supplying intermediate terms in a given series of terms. In this report, we consider only interpolation of univariate (one-variable) single-valued functions. We seek a method of interpolation that will produce a natural-looking curve when it is applied to curve fitting. (When there is no risk of confusion, two terms "interpolation" and "curve fitting" will be used synonymously in this report.)

* The author is with the Institute for Telecommunication Sciences, National Telecommunications and Information Administration, U.S. Department of Commerce, Boulder, Colorado 80303-3328.

Some time ago, Akima (1970, 1972) developed a method of interpolation and smooth curve fitting (hereinafter referred to as the original A method) that produced natural-looking curves in many cases. The original A method emphasizes natural appearance of the resultant curve, by suppressing excessive undulations (or wiggles) of the curves. It is described in a textbook (Carnahan and Wilkes, 1973) and is included in the IMSL (International Mathematical and Statistical Libraries, Inc.) Library under the routine name of IQHSCU (IMSL, 1979). Examinations of the method with some "hostile" examples, however, have revealed that the method needs further improvement.

There are several aspects of interpolation. A method emphasizes an aspect. Another method emphasizes another aspect. The method developed by Fritsch and Carlson (1980) or the improved version of the method by Fritsch (1982) or by Fritsch and Butland(1984) (hereinafter referred to collectively as the F-C-B method) outperforms the original A method when monotonicity of the data must be preserved. The method developed by Roulier (1980) or the method by McAllister and Roulier (1981a; 1981b) (hereinafter referred to collectively as the M-R method) preserves the convexity of the data in addition to the monotonicity. There are also many other shape-preserving methods as described by Gregory (1985).

In the primitive stage of development, a method can be superior to all other methods in most cases. In the advanced stage of development with many methods existing, it becomes apparent that a method is superior to other methods only in some applications and another method is superior in other applications. Desired or required properties for interpolation methods differ widely from case to case, and some properties are mutually incompatible.

In this report, we identify desired or required properties for an interpolation method, discuss their mutual compatibility, establish our goals, and derive the guidelines for developing an improved method. It turns out that one of our goals is to develop a method that has the accuracy of third-degree (cubic) polynomial, i.e., a method that accurately interpolates the given data when the given set of data points lies on a cubic curve. We have developed a method (hereinafter referred to as the improved A method) that meets the goals. Like the original A method, the improved A method does not always preserve monotonicity or convexity; we have not intended to preserve it in developing the improved A method. We propose the improved A method as a replacement for the original A method when natural appearance of the resultant curve is important; we do not propose it as a replacement for the F-C-B method or the M-R method when monotonicity or convexity must be preserved.

In this report, desired properties of interpolation are discussed and the goals for development are established in Section 2. An interim method that has the accuracy of a second-degree polynomial and some other desirable properties is developed as the interim step of improvement of the original A method in Section 3. A method that has the accuracy of a third-degree polynomial and meets the goals is developed as the improved A method in Section 4. Possible use of higher-degree polynomials is considered as a variation of the improved A method in Section 5 with the details left to Appendix B. (Both methods developed in Sections 3 and 4 use third-degree polynomials.) Some examples are shown in Section 6. A brief summary and some remarks for the use of the improved A method are given in Section 7. A Fortran subroutine subprogram that implements the method developed in Sections 4 and 5 is listed in Appendix A.

Throughout this report we use some conventions. We assume

- that the x and y variables represent the independent variable and the function value,
- that the x and y variables also represent the abscissa and ordinate of a two-dimensional Cartesian coordinate,
- that the first derivative of the function (or the slope of the curve) is represented by y',
- that the x, y, and y' values at data point P_i are represented by x_i, y_i, and y'_i, and
- that the sequence $\{x_i\}$ is in an increasing order.

2. DESIRED PROPERTIES OF INTERPOLATION

There are several desired properties of interpolation depending on particular purposes of the user. Since some desired properties are mutually incompatible, as discussed later, any one method cannot possess all the desired properties simultaneously. In this section we will identify major properties desired for interpolation, discuss their mutual compatibility, and set our goals by selecting some of them.

In order for the curve to be smooth when interpolation is applied to curve fitting, we require that the interpolating function and its first derivative be continuous, i.e., in mathematical terms, the function be C^1 continuous.

Since it is desired in many cases that the curve is affected only in a small neighborhood of the data point when a data point is added, deleted, or moved, we require that the method be based on local procedures confined to a small neighborhood of the point at which interpolation of the value is required. This requirement was already recognized before the turn of the century; a method based on local procedures was developed by Karup (1899) and later improved by Ackland (1915). We do not consider the so-called global interpolation methods such as the spline-function method (Greville, 1967; Cline, 1974). Although some traditional methods such as Lagrange's or Newton's method (Hildebrand, 1956; Davis, 1975) are based on local procedures, we also do not consider these methods since these methods fail to produce a C^1 continuous curve.

Preserving the shape of the data such as monotonicity and convexity is a desired property in some cases. Monotonicity must be preserved, for example, when the set of input data points represents a probability distribution function. The property of preserving monotonicity dictates that the portion of the curve between a pair of successive data points must lie between the two points in its ordinates and that the portion of the curve between a pair of successive data points having an identical ordinate value must be a horizontal line segment. The property of preserving convexity dictates that the portion of the curve that connects three collinear data points must be a line segment.

Invariance under certain types of coordinate transformation is a desired property in some applications. Desirability of invariance under a linear-scale transformation

$$x = a\,u \qquad y = b\,v, \tag{1}$$

where a and b are nonzero constants, is obvious. In some cases, invariance under another type of linear coordinate transformation

$$x = a\,u \qquad y = b\,u + c\,v, \tag{2}$$

where a, b, and c are nonzero constants, is desirable. In studying the fluctuation of a clock, for example, one may plot either the reading of the clock against the correct time (as an almost 45° slope curve) or the error against the correct time (as an almost horizontal curve), and both plottings should represent physically the same phenomenon.

Symmetry of the method is another desired property in most applications. Symmetry is described here as the property that the method produces a symmetric curve when the data points are symmetric.

Producing a curve that looks natural is another important property desired for the method. Since a skilled draftsman draws a natural-looking curve with French curves, it is desirable that the method simulate a skilled draftsman. Since a natural-looking curve has not been defined mathematically, we will consider here what behavior of a curve makes the curve look natural. It seems that a good way of describing a natural-looking curve is to describe the behavior of the curve in terms of curvature that is the rate of deviation of a curve from a straight line. We believe that a curve does not look natural if its curvature changes excessively and too frequently. We believe that change of curvature must be suppressed as much as possible to make the curve look natural. From this general requirement, several specific requirements are derived. Excessive undulations that are accompanied by excessive curvature changes should be avoided. Since the curve changes from convex to concave or the reverse and therefore changes its curvature at an inflection point, the number of inflection points must be kept to a small number. Since a line segment embedded in a generally curved line exhibits large changes of curvature and additional inflection points near the endpoints of the line segment in many cases, producing such a line segment must be avoided if possible.

In addition to these descriptions in terms of curvature, we also describe the property of producing a natural-looking curve in terms of accuracy of a simple mathematical function. We say that an interpolation method has the accuracy of a function if the method accurately interpolates the data when the data points lie on a curve of the function. We know intuitively that curves of some simple mathematical functions such as a low-degree polynomial or a sine (or cosine) function look good and natural. Therefore, accurate or close interpolation of data points given on such curves is also desired. Particularly, a third-degree polynomial is a polynomial of the lowest degree that can have an inflection point. It can be used at least locally to approximate a curve that has inflection points. Therefore, we require the accuracy of a third-degree polynomial for our method.

Both the method developed by Karup and the original A method have the accuracy of a second-degree polynomial conditionally, i.e., they interpolate given data accurately when the given data points are equally spaced in their abscissas. The osculatory method by Ackland has the accuracy of a second-degree polynomial unconditionally, i.e., it interpolates given data accurately even when the given data points are unequally spaced. We will later review these methods in more detail and develop an interim method that has the (unconditional) accuracy of a second-degree polynomial and other desirable properties.

An interpolation method is desired to be continuous in the sense that the resultant curve changes very little when a small change is made in the input data. The curve resulting from the original A method will change abruptly when three data points, P_{i-2}, P_{i-1}, and P_i, are collinear and three data points, P_i, P_{i+1}, and P_{i+2}, are changed from almost collinear to exactly collinear. If such discontinuous behavior cannot be eliminated entirely, it is desired to reduce the chance of occurrence of such behavior.

Linearity of the method is another desired property in some applications. Linearity is described here as the property that the interpolated values satisfy $y(x) = a\ y^{(1)}(x) + b\ y^{(2)}(x)$ if $y_i = a\ y^{(1)}(x_i) + b\ y^{(2)}(x_i)$ for all i, where a and b are nonzero constants.

Unfortunately, some of these desired properties are mutually incompatible. One of the most serious problems is incompatibility of the property of preserving monotonicity with some other properties. Its incompatibility with the property of invariance under the linear coordinate transformation represented by (2) is obvious. The requirement for preserving monotonicity or convexity sometimes produces embedded line segments and an excessive number of inflection points. The monotonicity or convexity requirement is incompatible with the requirement of accuracy of a third-degree polynomial. It may not be a good idea to require preserving monotonicity or convexity when such a property is not really necessary. Since we already have the F-C-B or M-R method as a good interpolation method, we will drop the requirements for preserving monotonicity and convexity.

Even within the same general desirability for producing a natural-looking curve, requirements for some desirable properties need adjustment and compromise. Suppression of excessive undulations and embedded line segments need a mutual adjustment. When several successive data points are on a straight line (such as the x axis) and other data points are elsewhere, the portion of the curve that connects the collinear data points is generally desired to be a line segment. The original A method and the M-R method produce a line segment when three data points are collinear, and the F-C-B method does the same when three data points are on a horizontal line. This property tends to produce unnatural-looking line segments. We feel that a deviation from the line segment should be allowed when only three data points are collinear. In the method developed in this report, we require a line segment when four data points or more are collinear, but not when only three data points are collinear.

The requirement of line segment for several collinear data points, regardless of the number of collinear data points, is incompatible with requirements for other properties such as continuity and linearity of the method as in the original A method. Although the discontinuity of the method is not entirely eliminated, the increase in the number of collinear data points for a line segment from three to four is expected to reduce the chance of occurrence of discontinuous behavior. This can be accounted for by the fact that the probability of having four collinear data points by chance is much less than the probability of having three collinear data points by chance.

3. THE INTERIM METHOD

As a preliminary step in developing an interpolation method that meets our goals including having the accuracy of a third-degree polynomial, we try to develop, in this section, an interim method that has the accuracy of a second-degree polynomial and other desirable properties. For this purpose, we first review basic procedures and major properties of the osculatory method (Ackland, 1915) that has the accuracy of a second-degree polynomial and the original A method (Akima, 1970, 1972) that has some of the desirable properties.

In both the osculatory and original A methods, the interpolating function is a piecewise function composed of a set of third-degree (cubic) polynomials, applicable to successive intervals of the given data points. Function value y corresponding to an x value in the interval between x_i and x_{i+1} is calculated by

$$y = a_0 + a_1(x - x_i) + a_2(x - x_i)^2 + a_3(x - x_i)^3, \tag{3}$$

where a_0, a_1, a_2, and a_3 are the coefficients of the polynomial for that interval. These coefficients are determined by the given y values and the estimated y' values (i.e., the first derivatives) at the endpoints of the interval as

$$\begin{aligned} a_0 &= y_i, \\ a_1 &= y_i', \\ a_2 &= -[2(y_i' - m_i) + (y_{i+1}' - m_i)] / (x_{i+1} - x_i), \\ \text{and} \quad a_3 &= [(y_i' - m_i) + (y_{i+1}' - m_i)] / [(x_{i+1} - x_i)^2], \end{aligned} \tag{4}$$

where m_i is the slope of the line segment connecting P_i and P_{i+1} and is represented by

$$m_i = (y_{i+1} - y_i) / (x_{i+1} - x_i). \tag{5}$$

The only difference between the two methods is in the procedure of estimating the first derivative of the interpolating function at each given data point.

In the osculatory method, the first derivative of the interpolating function at data point P_i is estimated with a set of three data points, P_{i-1}, P_i, and P_{i+1}. It is estimated as the first derivative of the second-degree polynomial fitted to the three data points. It is clear from this procedure that the first derivative is accurately estimated when the three data points are on a curve of a second-degree polynomial.

In the original A method, the first derivative of the function at data point P_i is estimated with a set of five points, P_{i-2}, P_{i-1}, P_i, P_{i+1}, and P_{i+2}. Two line-segment slopes, m_{i-1} (the slope of the line segment connecting P_{i-1} and P_i) and m_i (the slope of the line segment connecting P_i and P_{i+1}), are used as the primary estimates of the first derivative, and the final estimate is calculated as the weighted mean of the primary estimates, i.e.,

$$y_i' = (m_{i-1}w_{im} + m_iw_{ip}) / (w_{im} + w_{ip}). \quad (6)$$

The weight for m_{i-1} is the reciprocal of the absolute value of the difference between m_{i-1} and m_{i-2}, and the weight for m_i is the reciprocal of the absolute value of the difference between m_{i+1} and m_i, i.e.,

$$\begin{aligned} w_{im} &= 1 / \text{abs}\{m_{i-1} - m_{i-2}\}, \\ w_{ip} &= 1 / \text{abs}\{m_{i+1} - m_i\}, \end{aligned} \quad (7)$$

where abs{ } stands for "the absolute value of." The basic concept behind the selection of the weight is that the primary estimate based on the data points on the left (or right) side of the point in question should be given a small weight if the data points on the left (or right) side are "volatile" (or far from being collinear). The first derivative is accurately estimated when the five data points are on a curve of a second-degree polynomial and are equally spaced in their abscissas.

Before developing the interim method, we present the expression of the first derivative, at data point P_i, of a second-degree polynomial fitted to a set of three data points, P_i, P_j, and P_k. If we denote the first derivative by $F(i,j,k)$, it is given by

$$\begin{aligned} F(i,j,k) = {} & [(y_j-y_i)(x_k-x_i)^2-(y_k-y_i)(x_j-x_i)^2] \\ & / [(x_j-x_i)(x_k-x_i)(x_k-x_j)]. \end{aligned} \quad (8)$$

Note that the first index in the expression of $F(i,j,k)$ must be i, which is the point number of the point in question, and that the remaining indices can be given in any order.

With this notation, the estimate of the first derivative in the osculatory method is represented by

$$y_i' = F(i,i-1,i+1). \quad (9)$$

The set of three successive data points that can be used for estimating the first derivative of the interpolating function at data point P_i by fitting a second-degree polynomial is not limited to the set of three points used in the osculatory method, i.e., the set of three points, P_{i-1}, P_i, and P_{i+1}. Any set of consecutive three data points can be used if the set includes P_i. There are three qualified sets, i.e., (P_{i-2}, P_{i-1}, P_i), (P_{i-1}, P_i, P_{i+1}), and (P_i, P_{i+1}, P_{i+2}). We can, therefore, fit three second-degree polynomials to the three sets of three data points and calculate three primary estimates. The estimate (9) used in the osculatory method can be considered one of the three primary estimates. The three primary estimates of the first derivative of the interpolating function at P_i are

$$y'_{im} = F(i,i-2,i-1),$$

$$y'_{i0} = F(i,i-1,i+1), \qquad (10)$$

and

$$y'_{ip} = F(i,i+1,i+2).$$

Since all these three primary estimates for the first derivative are accurate when all the five data points, P_{i-2} through P_{i+2}, are on a curve of a second-degree polynomial, use of a weighted mean of the three primary estimates with any set of weights, i.e.,

$$y'_i = (y'_{im}w_{im} + y'_{i0}w_{i0} + y'_{ip}w_{ip}) / (w_{im} + w_{i0} + w_{ip}), \qquad (11)$$

as the final estimate for the first derivative of the interpolating function will yield a method that has the accuracy of a second-degree polynomial. The osculatory method can be considered a special case where the two weights w_{im} and w_{ip} equal zero.

The basic concept behind the original A method dictates that a small weight be assigned to a primary estimate if the primary estimate is calculated from a set of volatile data points. In the interim method, we could take the absolute value of the second derivative of the second-degree polynomial fitted to a set of three data points as a measure of volatility, use the reciprocal of the measure of volatility as the weight, and assign the weight to the primary estimate calculated from the set of data points. Use of the second-degree polynomial as a measure of volatility, however, is applicable only to a set of three data points. For future development, we need a measure of volatility that is independent of the number of data points. As such a measure of volatility, we take the sum of squares of deviations from a straight line of the least-square fit. Then, the measure of volatility is represented by

$$\begin{aligned} V(i,j,k) &= \sum[y - (b_0 + b_1 x)]^2 \\ &= \sum y^2 - (b_0 \sum y + b_1 \sum xy), \end{aligned} \tag{12}$$

where b_0 and b_1 are the coefficients of the first-degree (linear) polynomial of the least-square fit to the data points and are represented by

$$\begin{aligned} b_0 &= [\sum x^2 \sum y - \sum x \sum xy] \,/\, [3 \sum x^2 - (\sum x)^2], \\ b_1 &= [3 \sum xy - \sum x \sum y] \,/\, [3 \sum x^2 - (\sum x)^2]. \end{aligned} \tag{13}$$

In (12) and (13), symbol $\sum$ represents a summation over three data points, P_i, P_j, and P_k. Note that the three indices in the expression of $V(i,j,k)$ in (12) can be given in any order.

In addition to the volatility of the data point set, we include another factor in the weight. We consider that the primary estimate should be given a small weight if the data point set includes a data point or data points far distant from the data point in question. We define the distance factor by

$$D(i,j,k) = (x_j - x_i)^2 + (x_k - x_i)^2. \tag{14}$$

Note that the first index in the expression of $D(i,j,k)$ must be i, which is the point number of the point in question, and that the remaining indices can be given in any order.

We use the reciprocal of the product of $V(i,j,k)$ and $D(i,j,k)$ as the weight and assign the weight to the primary estimate calculated from the set of data points, P_i, P_j, and P_k. Then, the three weights corresponding to the three primary estimates (10) are represented by

$$w_{im} = 1 / [V(i,i-2,i-1)\ D(i,i-2,i-1)],$$

$$w_{i0} = 1 / [V(i,i-1,i+1)\ D(i,i-1,i+1)], \qquad (15)$$

$$w_{ip} = 1 / [V(i,i+1,i+2)\ D(i,i+1,i+2)].$$

When a set of three data points is collinear, the V value equals zero, and the corresponding weight becomes infinite. When any weight becomes infinite, we reset infinite weights to unity and finite weights to zero before using (11) to calculate the final estimate of the first derivative.

Note that the interim method uses a total of five data points P_{i-2} through P_{i+2} to estimate the first derivative at data point P_i.

Because of the use of these weights (15), the interim method has the property that, when a set of three data points, P_{i-1} through P_{i+1}, is collinear, the estimate of the first derivative at P_i equals the slope of the straight line passing through the set of data points. The method also has the property that, when a set of three data points, P_i through P_{i+2}, is collinear, the estimate of the first derivative at P_i equals the slope of the straight line passing through the set of data points unless another set of three data points, P_{i-2} through P_i, is also collinear. Note that the interim method has inherited these properties from the original A method.

When the data point in question is the first or the last data point, only one set of three data points is available and therefore only one primary estimate can be calculated. When the data point in question is the second or the second last data point, only two primary estimates can be calculated. In these cases, we use only the available primary estimate or estimates for calculating the final estimate of the first derivative.

Like the osculatory and original A methods, the interim method also uses a third-degree polynomial for the interval between each pair of successive data points. The interim method interpolates the y value with (3), (4), and (5).

Since the interim method retains some of the desirable properties of the original A method with an additional desirable property of the accuracy of a second-degree polynomial, we also call the interim method the improved A method of the second-degree polynomial version.

We will examine the performance of this interim method with some examples in Section 6.

4. THE METHOD

In the preceding section, we have developed an interim method that has the accuracy of a second-degree polynomial and retains some of the desirable properties of the original A method. In this section we will develop a method that has the accuracy of a third-degree polynomial by modifying the interim method to a third-degree polynomial version. We also modify the osculatory method (which has the accuracy of a second-degree polynomial) in such a way that the modified method has the accuracy of a third-degree polynomial.

Before we proceed, we present the expression of the first derivative, at data point P_i, of a third-degree polynomial fitted to a set of four data points, P_i, P_j, P_k, and P_l. If we denote the first derivative by F(i,j,k,l), it is represented by

$$\begin{aligned} F(i,j,k,l) = [&(y_j-y_i)(x_k-x_i)^2(x_l-x_i)^2(x_l-x_k) \\ &+(y_k-y_i)(x_l-x_i)^2(x_j-x_i)^2(x_j-x_l) \\ &+(y_l-y_i)(x_j-x_i)^2(x_k-x_i)^2(x_k-x_j)] \\ / \ [&(x_j-x_i)(x_k-x_i)(x_l-x_i)(x_k-x_j)(x_l-x_k)(x_l-x_j)]. \end{aligned} \quad (16)$$

Note that the first index in the expression of F(i,j,k,l) must be i, which is the point number of the point in question, and that the remaining indices can be given in any order.

We also present the sum of squares of deviations from a straight line of the least-square fit as the measure of volatility. The measure of volatility is represented by

$$\begin{aligned} V(i,j,k,l) &= \sum[y - (b_0 + b_1x)]^2 \\ &= \sum y^2 - (b_0\sum y + b_1\sum xy), \end{aligned} \quad (17)$$

where b_0 and b_1 are the coefficients of the first-degree (linear) polynomial of the least-square fit to the data points and are represented as

$$b_0 = [\sum x^2 \sum y - \sum x \sum xy] \,/\, [4 \sum x^2 - (\sum x)^2],$$

and (18)

$$b_1 = [4 \sum xy - \sum x \sum y] \,/\, [4 \sum x^2 - (\sum x)^2].$$

In (17) and (18), symbol $\sum$ represents a summation over four data points, P_i, P_j, P_k, and P_l. Note that the four indices in the expression of $V(i,j,k,l)$ in (17) can be given in any order.

The distance factor can be represented as

$$D(i,j,k,l) = (x_j - x_i)^2 + (x_k - x_i)^2 + (x_l - x_i)^2. \tag{19}$$

Note that the first index in the expression of $D(i,j,k,l)$ must be i, which is the point number of the point in question, and that the remaining indices can be given in any order.

There are four sets of four consecutive data points that include P_i, i.e., $(P_{i-3}, P_{i-2}, P_{i-1}, P_i)$, $(P_{i-2}, P_{i-1}, P_i, P_{i+1})$, $(P_{i-1}, P_i, P_{i+1}, P_{i+2})$, and $(P_i, P_{i+1}, P_{i+2}, P_{i+3})$. Seven data points, P_{i-3} through P_{i+3}, are involved. We calculate four primary estimates of the first derivative of the interpolating function, each as the first derivative of a third-degree polynomial fitted to a set of four consecutive data points. These primary estimates are

$$y'_{imm} = F(i,i-3,i-2,i-1),$$

$$y'_{im} = F(i,i-2,i-1,i+1),$$

(20)

$$y'_{ip} = F(i,i-1,i+1,i+2),$$

and

$$y'_{ipp} = F(i,i+1,i+2,i+3).$$

Since these primary estimates are all accurate when all the seven data points are on a curve of a third-degree polynomial, use of any combination of these primary estimates, i.e.,

$$y'_i = (y'_{imm}w_{imm}+y'_{im}w_{im}+y'_{ip}w_{ip}+y'_{ipp}w_{ipp})/(w_{imm}+w_{im}+w_{ip}+w_{ipp}), \tag{21}$$

as the final estimate of the first derivative of the interpolating function yields a method that has the accuracy of a third-degree polynomial.

With these notations, developing the final method by modifying the interim method to the third-degree polynomial version is rather straightforward. Like the interim method, the final method uses all primary estimates, i.e., four primary estimates calculated by (20) in this case. It uses the reciprocal of the product of V(i,j,k,l) and D(i,j,k,l) calculated by (17) and (19) as the weight for the primary estimate calculated from the set of data points, P_i, P_j, P_k, and P_l. Then, the four weights corresponding to the four primary estimates (20) are represented by

$$w_{imm} = 1 \,/\, [V(i,i-3,i-2,i-1)\; D(i,i-3,i-2,i-1)],$$

$$w_{im} = 1 \,/\, [V(i,i-2,i-1,i+1)\; D(i,i-2,i-1,i+1)],$$

$$w_{ip} = 1 \,/\, [V(i,i-1,i+1,i+2)\; D(i,i-1,i+1,i+2)], \qquad (22)$$

and

$$w_{ipp} = 1 \,/\, [V(i,i+1,i+2,i+3)\; D(i,i+1,i+2,i+3)].$$

When a set of four data points is collinear, the V value equals zero, and the corresponding weight becomes infinite. When any weight becomes infinite, we reset infinite weights to unity and finite weights to zero before using (21) to calculate the final estimate of the first derivative.

Note that the final method uses a total of seven data points, P_{i-3} through P_{i+3}, to estimate the first derivative at data point P_i.

Because of the use of these weights shown in (22), the final method has the property that, when a set of four data points, P_{i-1} through P_{i+2}, is collinear, the estimate of the first derivative at P_i equals the slope of the straight line passing through the set of data points. The method also has the property that, when a set of four data points, P_i through P_{i+3}, is collinear, the estimate of the first derivative at P_i equals the slope of the straight line passing through the set of data points unless another set of four data points, P_{i-3} through P_i, is also collinear. It is clear from these properties of the method that, when a set of four data points or more is collinear, the method will produce a line segment across the set of data points.

When the data point in question is the first or the last data point, only one set of four data points is available and therefore only one primary estimate can be calculated. When the data point in question is the second or the second last data point, only two primary estimates can be calculated. When the data point in question is the third or the third last data point, only three primary estimates can be calculated. In these cases, the method uses only the available primary estimate or estimates for calculating the final estimate of the first derivative.

Like the interim method as well as the osculatory and original A methods, the final method also uses a third-degree polynomial for the interval between each pair of successive data points. This method interpolates the y value with (3), (4), and (5).

Since the final method is expected to retain the desirable properties of the original A method with additional desirable property of the accuracy of a third-degree polynomial, we also call the final method the improved A method of the third-degree polynomial version or simply the improved A method.

For completeness in later comparisons, we develop another method by modifying the osculatory method to its third-degree polynomial version. The osculatory method of the original version (or the second-degree polynomial version) uses only one primary estimate based on the set of three data points that includes the data point in question as the center point. It therefore uses no weights based on the volatility of the data points. To satisfy the symmetry requirements, the third-degree polynomial version uses two primary estimates, each based of the set of four data points that includes the data point in question near the center of the set. It uses no weights based on the volatility of the data points; it uses a simple mean of the second and third primary estimates in (20). If we use (21) to calculate the final estimate of the first derivative, we can represent the weights as

$$w_{imm} = 0,$$

$$w_{im} = w_{ip}, \tag{23}$$

and

$$w_{ipp} = 0.$$

Note that this method uses only five data points, P_{i-2} through P_{i+2}, to estimate the final estimate of the first derivative.

In this osculatory method of the third-degree polynomial version, we also use a third-degree polynomial for the interval between each pair of successive data points. This method interpolates the y value with (3), (4), and (5).

We will examine the performances of these two methods with some examples in Section 6.

5. USE OF A HIGHER-DEGREE POLYNOMIAL (A VARIATION)

So far in the preceding two sections, we have concentrated on the procedure for estimating the first derivative of the interpolating function at each data point. We have assumed a third-degree polynomial to be applied to the interval of each successive pair of data points. A third-degree polynomial is not, however, the only function that is determined by the values of the function and first derivative at two points. A hyperbolic function (or a combination of exponential functions), rational function (i.e., a quotient of two polynomials), a combination of two second-degree polynomials, and higher-degree polynomial are some of the examples of such functions. In this section, we present an interpolating function based on an nth-degree polynomial, with n being equal to three or greater. (See Appendix B for the behavior, in an interval, of some interpolating functions including the polynomials of various degrees.)

For simplicity, we consider an interpolating function $y = y(x)$ in an interval between P_i and P_{i+1} in a new coordinate system in which the abscissa values equal 0 and 1 at P_i and P_{i+1}, respectively, and the ordinate values equal to 0 at these two points. We call the new coordinate system the u-v coordinate system. The linear coordinate transformation between the u-v coordinate system and the x-y coordinate system is represented by

$$
\begin{aligned}
x - x_i &= (x_{i+1} - x_i)u, \\
y - y_i &= (y_{i+1} - y_i)u + v.
\end{aligned}
\qquad (24)
$$

The first derivatives in the two coordinate systems, $y' = dy/dx$ and $v' = dv/du$, are related by

$$y' - m_i = v' / (x_{i+1} - x_i), \tag{25}$$

where

$$m_i = (y_{i+1} - y_i) / (x_{i+1} - x_i). \tag{26}$$

It is clear from (24) through (26) that

$$\begin{aligned} &u = 0, \quad v = 0, \quad v_0' = (y_i' - m_i)\,(x_{i+1} - x_i) \quad \text{at } P_i, \\ &u = 1, \quad v = 0, \quad v_1' = (y_{i+1}' - m_i)\,(x_{i+1} - x_i) \quad \text{at } P_{i+1}, \end{aligned} \tag{27}$$

where v_0' and v_1' are the v' values at $u = 0$ and $u = 1$. The set of equations (27) indicates that the u-v coordinate system has the property described in the beginning of this paragraph.

As the $v(u)$ function, we present here an nth-degree polynomial in u represented by

$$v(u) = A_0\,[u + u^n] + A_1\,[(1 - u) + (1 - u)^n]. \tag{28}$$

The coefficients A_0 and A_1 are given by

$$\begin{aligned} A_0 &= \quad [v_0' + (n - 1)v_1'] / [n(n - 2)], \\ A_1 &= - [(n - 1)v_0' + v_1'] / [n(n - 2)]. \end{aligned} \tag{29}$$

When the y and y' values are given at P_i and P_{i+1}, we can calculate the v' values at these points by (27) and (26), and the A_0 and A_1 coefficients by (29). For a given x value, we can calculate the corresponding u value by

$$u = (x - x_i) / (x_{i+1} - x_i), \tag{30}$$

which is equivalent to the first equation in (24), the v(u) value by (28), and finally the y value by

$$y = y_i + (y_{i+1} - y_i)u + v(u), \tag{31}$$

which is equivalent to the second equation in (24). It is easy to show that (31) with supplementary relations (26) through (30) reduces to (3) with (4) and (5) when n equals three.

Use of a higher-degree polynomial has an advantage. Undulations in the resultant curve will be reduced when the interpolation method is applied to curve fitting. Use of a higher-degree polynomial, however, has a disadvantage, also. The resultant curve sometimes is too "tight," i.e., the portion of the curve between a pair of successive data points is so close to the line segment connecting the pair of data points that the whole curve looks as if it were deflected. Use of a higher-degree polynomial has another disadvantage. The interpolation method will not achieve the accuracy of a third-degree polynomial.

We have implemented the variation (i.e., the use of a higher-degree polynomial described in this section) in the improved A method as a user option. In the Fortran subprogram subroutine listed in Appendix A, selection of the degree of the polynomial is left to the user. Depending on the user's situation, the user has an option: the user will either use a value greater than three as the degree of the polynomial to reduce undulations while giving up the accuracy of a third-degree polynomial, or the user will use a value of three retaining the accuracy of a third degree polynomial. Dependence of the performance of the curve of v(u) in (28) on the degree of the polynomial n is shown graphically in Appendix B. Examples presented in Section 6 include curve fitted with n = 6. It is expected that the user of the improved A method will develop a general idea on the selection of the degree of the polynomial from the information in Appendix B and the examples in Section 6.

6. EXAMPLES

This section illustrates performances of the methods developed in this report with examples in Figures 1 through 11. In each figure presented in this section, curves resulting from two existing methods are also plotted for comparison. Six curves in each figure are, from the top to the bottom,

(1) the original osculatory method (or the osculatory method of the second-degree polynomial version developed by Ackland, 1915)

(2) the modified osculatory method (or the osculatory method of the third-degree polynomial version developed in Section 4)

(3) the original A method (developed by Akima, 1970)

(4) the interim method (or the improved A method of the second-degree polynomial version developed in Section 3)

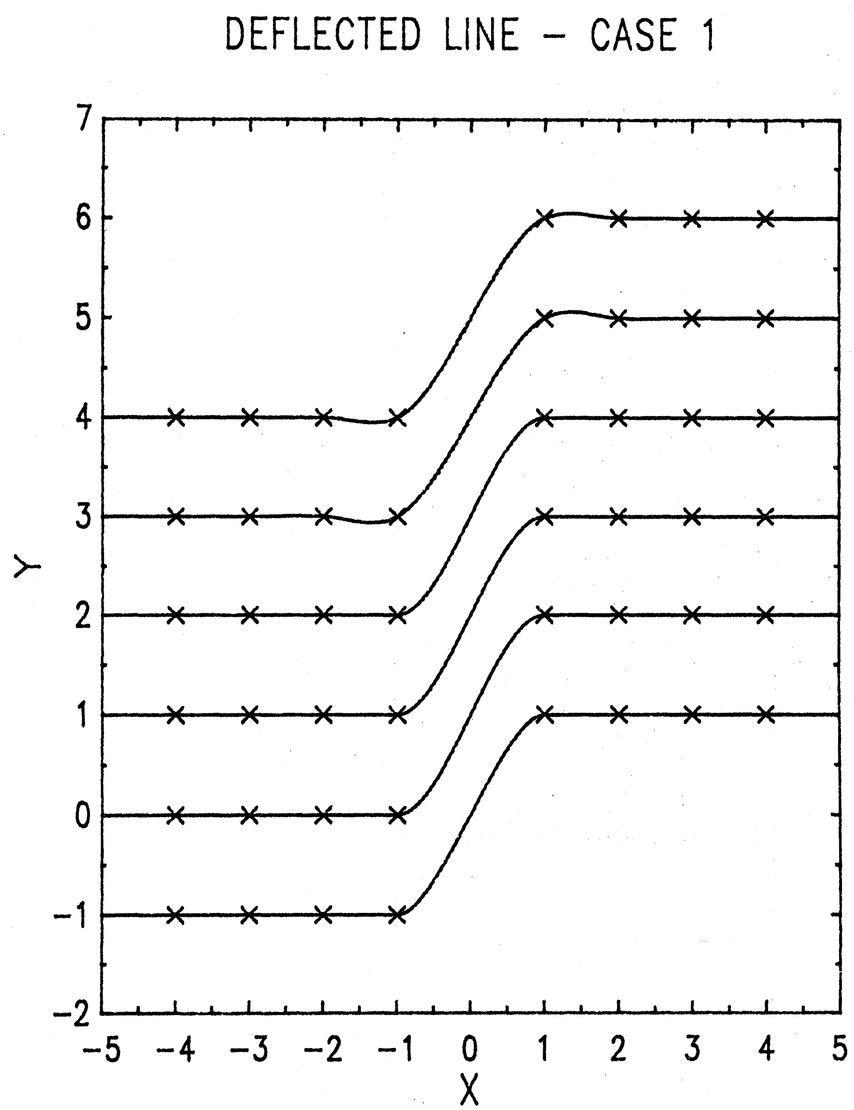
(5) the improved A method with n = 3 (or the improved A method of the third-degree polynomial version developed in Section 4, without the variation described in Section 5)

(6) the improved A method with n = 6 (or the improved A method of the third-degree polynomial version developed in Section 4, with the variation described in Section 5, with the degree of polynomial set to six).

Each data point is plotted with an "x" symbol. The x and y coordinate values of the data points are tabulated above the caption of each figure.

In Figure 1, data points are taken from a deflected line. The top two curves (resulting from the two osculatory methods) exhibit overshoots in the horizontal portions of the curve, while the overshoots are nonexistent in other curves (resulting from the original A method, interim method, and improved A method). The bottom two curves (resulting from the improved A method) illustrate the effect of the degree of polynomials, n = 3 versus n = 6.

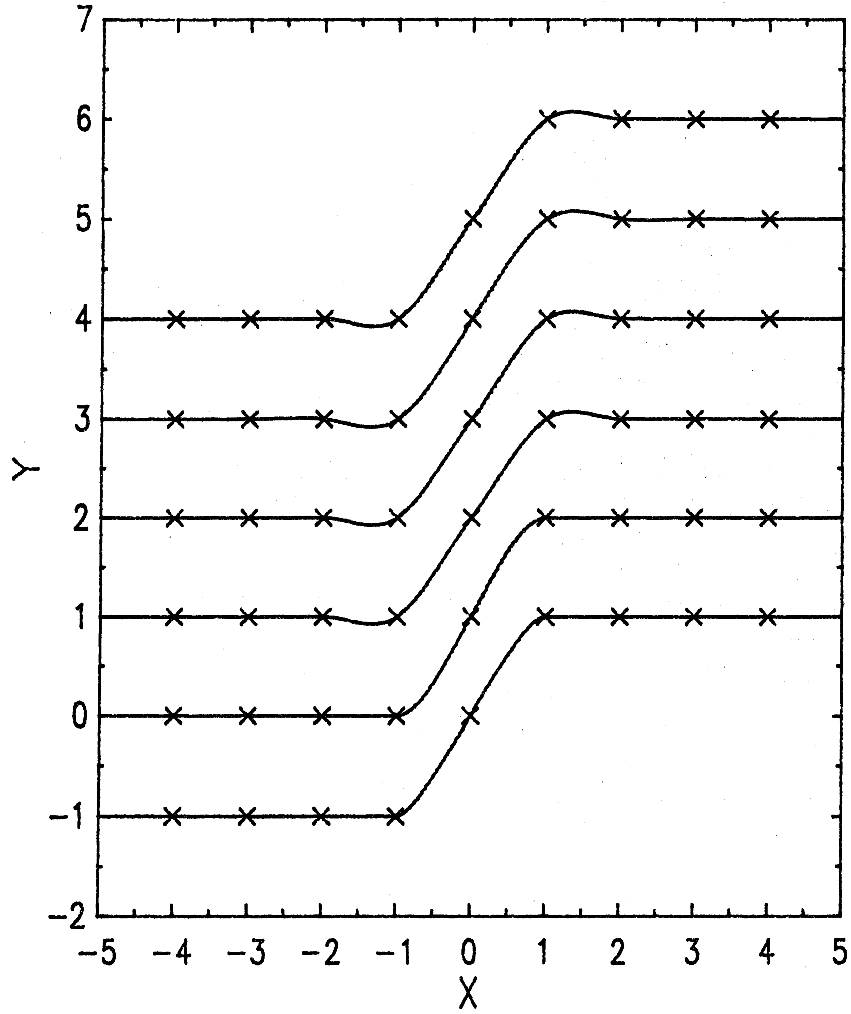
In Figure 2, data points are taken also from a deflected line; they consist of all data points in Figure 1 plus a data point at the center of the sloping region. The top four curves (resulting from the two osculatory methods, original A method, and interim method) exhibit overshoots in the horizontal portions of the curve, while the overshoots are nonexistent in the bottom two curves (resulting from the improved A method). The bottom two curves again illustrate the effect of the degree of polynomials, n = 3 versus n = 6.



x =	-4	-3	-2	-1	1	2	3	4
y =	-1	-1	-1	-1	1	1	1	1

Figure 1. Deflected-line data, Case 1.

DEFLECTED LINE - CASE 2



x =	-4	-3	-2	-1	0	1	2	3	4
y =	-1	-1	-1	-1	0	1	1	1	1

Figure 2. Deflected-line data, Case 2.

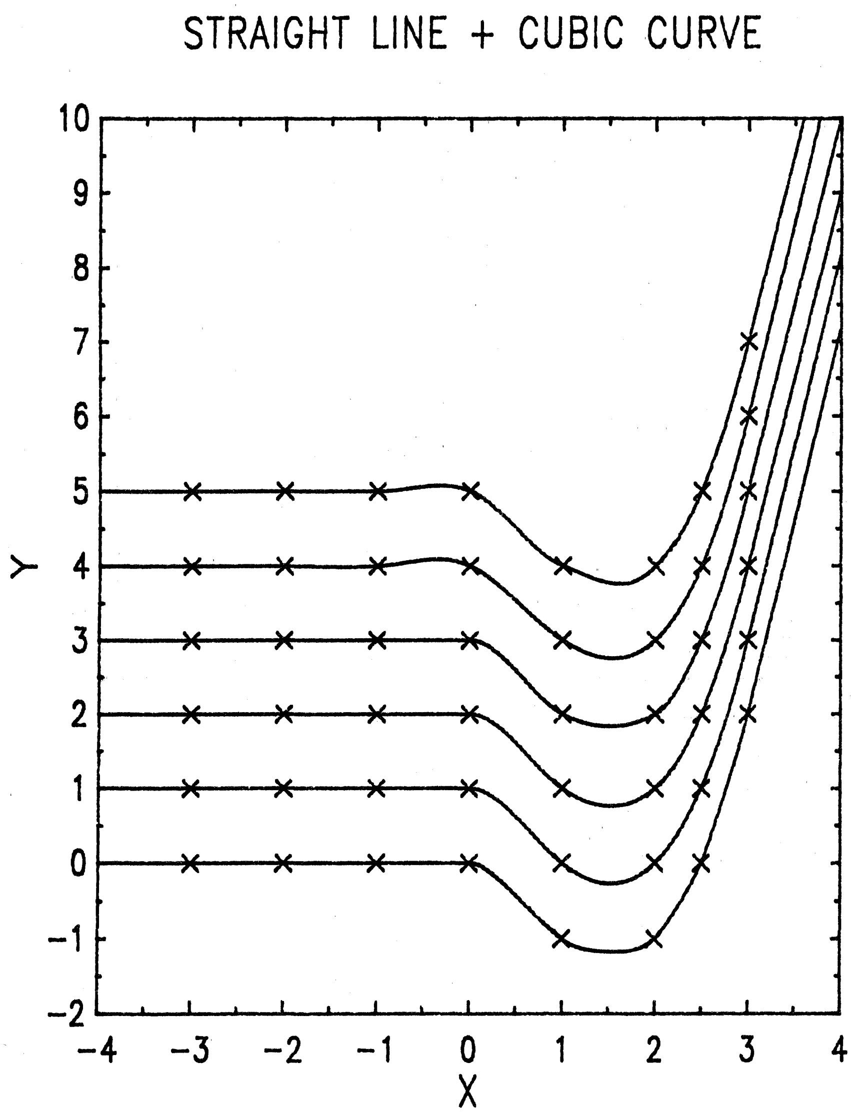
In Figure 3, the first four data points are on a horizontal straight line and the last six data points are on a curve of a third-degree polynomial, with the third and fourth points overlapping on both lines. The top two curves (resulting from the two osculatory methods) exhibit overshoots around $x = 0$, while the other four curves (resulting from the original A method, interim method, and improved A method) look good.

Figure 4 demonstrates how the improved A method interpolates the data points given on the curves of a simple known function. The data points are on a cubic curve at unequal intervals. As is expected, the second curve (resulting from the modified osculatory method) and the second curve from the bottom (resulting from the improved A method with $n = 3$) look good, while the first, third, and fourth curves (resulting from the original osculatory method, original A method, and interim method, respectively) exhibit irregularities. In each of the two intervals around the center point, the portion of the first curve (from the original osculatory method) has an inflection point. In the two intervals around the center point, the portions of the third and fourth curves (from the original A method and interim method, respectively) are line segments. The disadvantage of the use of a higher-degree polynomial is demonstrated in the bottom curve (resulting from the improved A method with $n = 6$).

Figure 5 also demonstrates how the improved A method interpolates the data points given on the curves of a simple known function. The data points are on a sine curve at unequal intervals. In this rather contrived example, the general trends of the curves in Figure 4 are even more pronounced. Figures 4 and 5 indicate that higher-degree polynomials should be used sparingly.

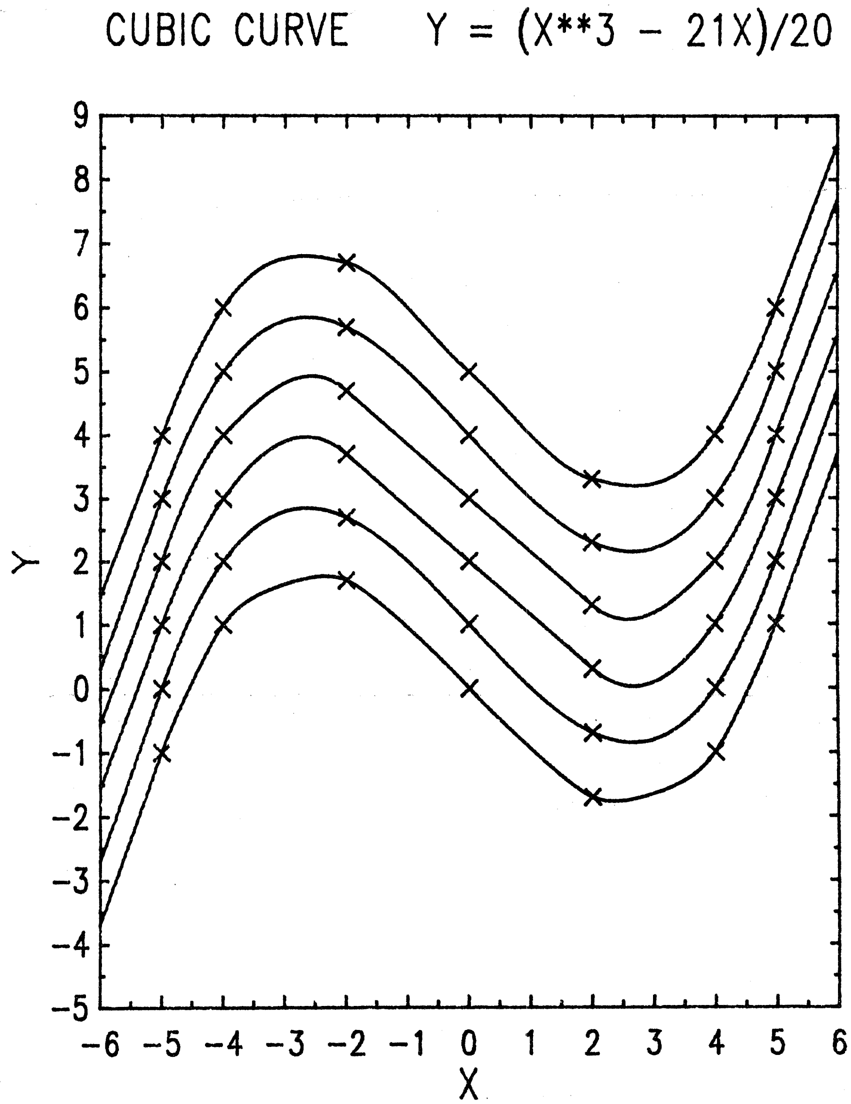
The data points for Figure 6 are taken from Akima (1970). The top two curves (resulting from the two osculatory methods) exhibit undulations in the interval between $x = 7$ and 8, while all other curves (resulting from the original A method, interim method, and improved A method) look good. We will modify this data point set in several ways and see how the curves resulting from various methods behave for each of the modified data point sets in the figures that follow.

The data point set for Figure 7 is Modification A of the original data point set for Figure 6. Two leftmost data points are removed from the original set and the remaining data points are moved horizontally. As is expected, removal of the two points has no effect on the curves resulting from all methods. The undulations in the top two curves (resulting from the two



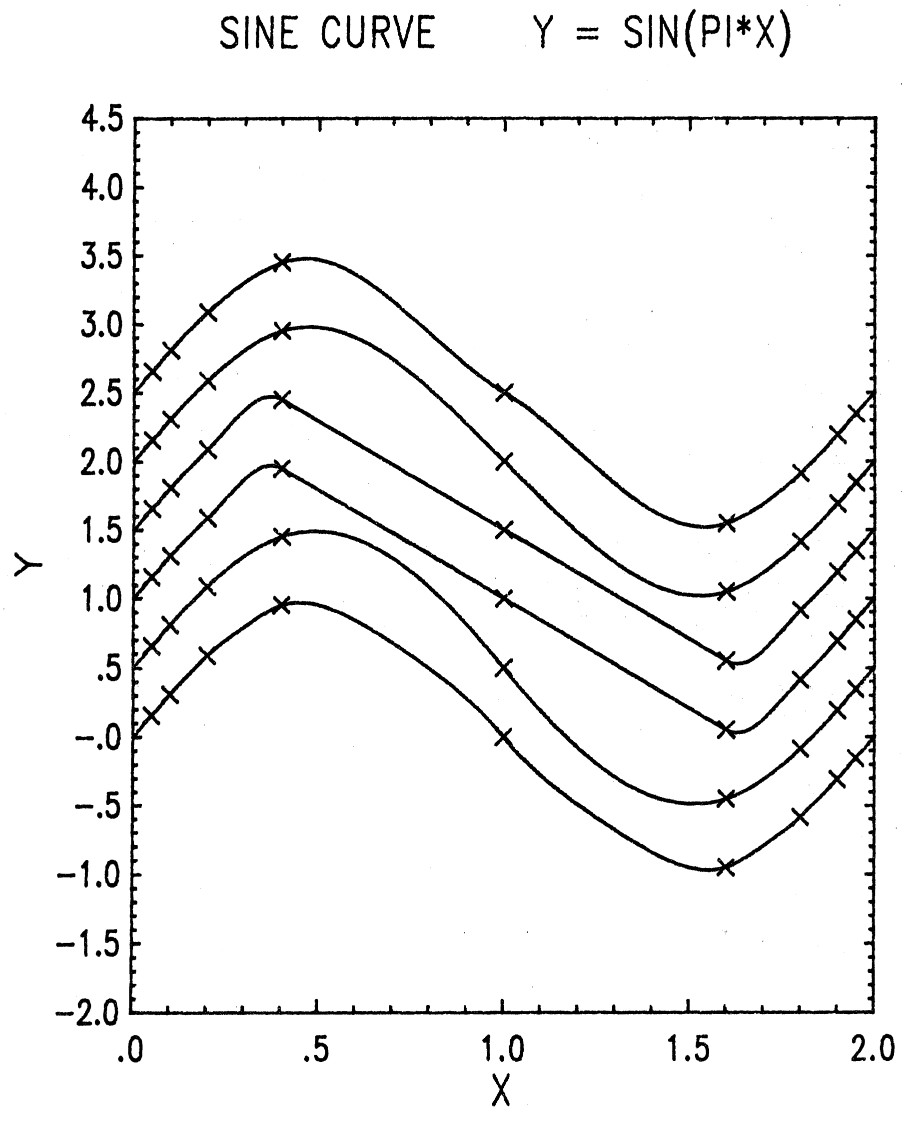
x =	-3	-2	-1	0	1	2	2.5	3
y =	0	0	0	0	-1	-1	0	2

Figure 3. Straight line plus cubic curve, $y = 0$ and $y = x^3/3 - x^2/2 - 5x/6$.



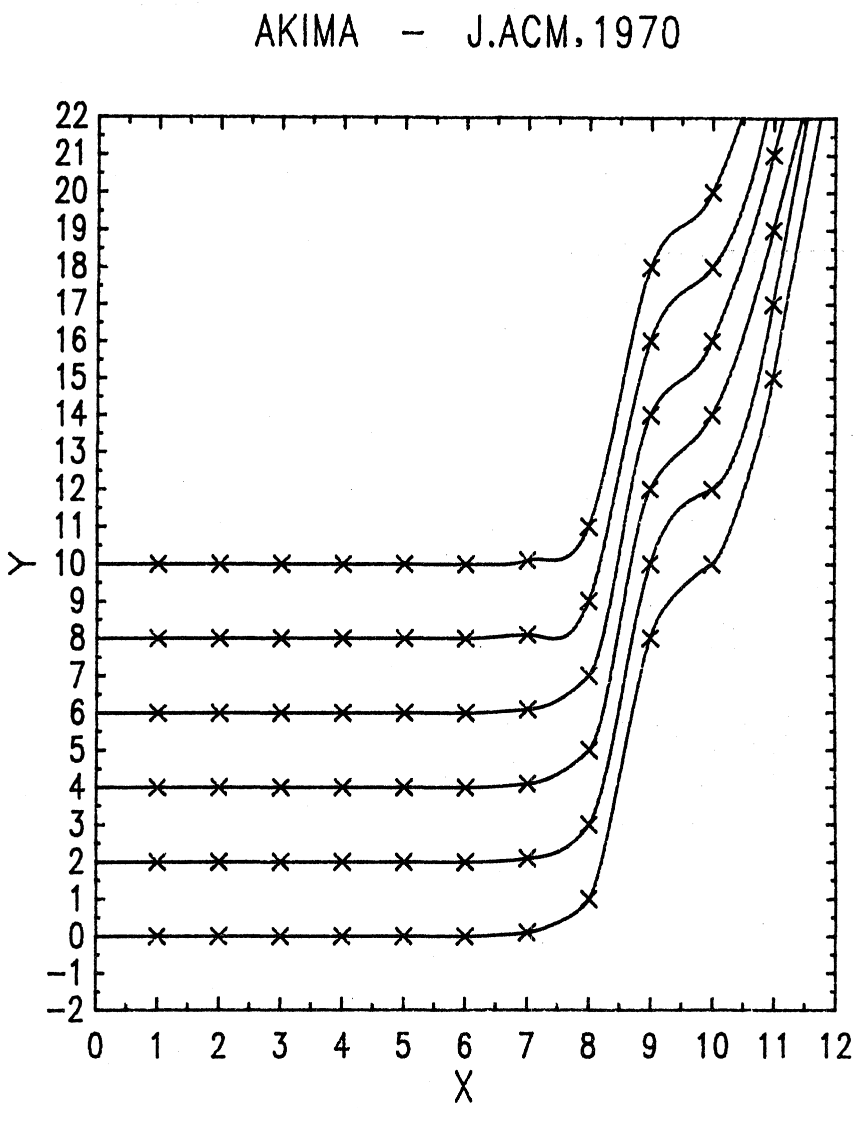
x =	-5	-4	-2	0	2	4	5
y =	-1	1	1.7	0	-1.7	-1	1

Figure 4. Cubic curve, $y = (x^3 - 21x)/20$.



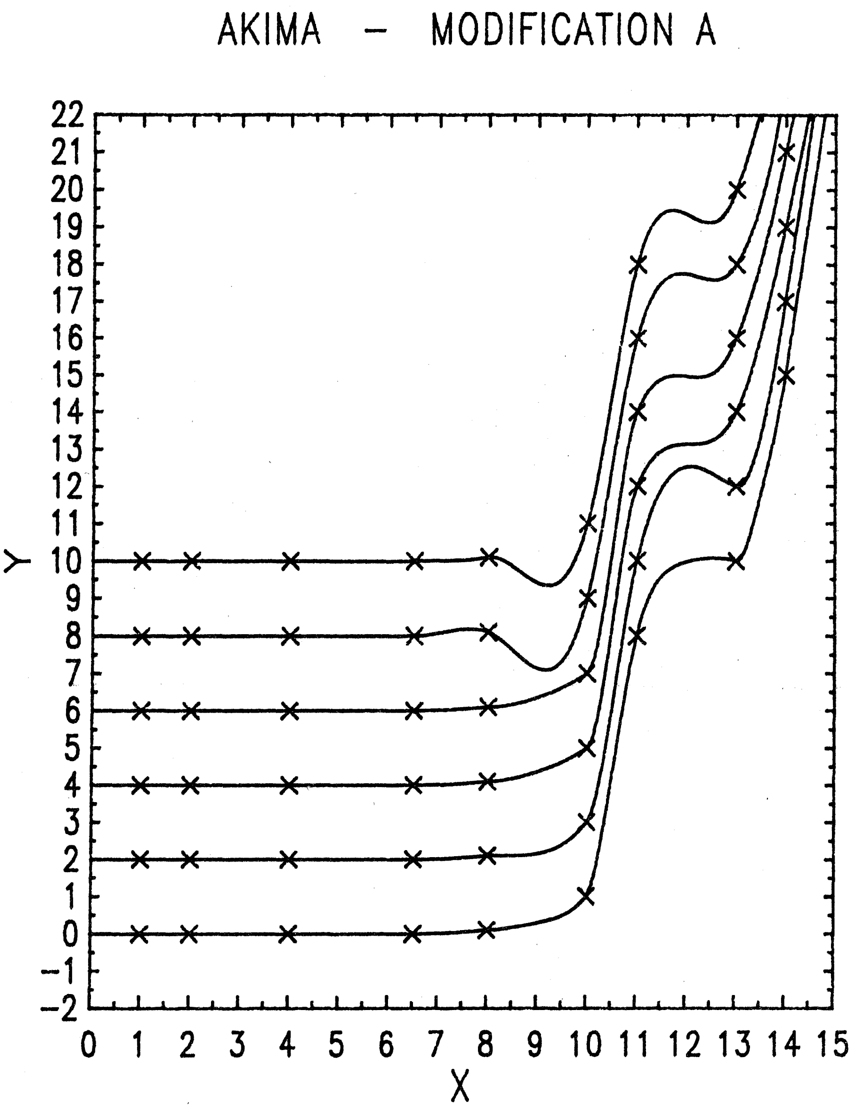
x	=	0.05	0.10	0.20	0.40	1.00	1.60	1.80	1.90	1.95
y	=	.1564	.3090	.5878	.9511	.0000	-.9511	-.5878	-.3090	-.1564

Figure 5. Sine curve, $y = \sin(\pi x)$.



x	=	1	2	3	4	5	6	7	8	9	10	11
y	=	0	0	0	0	0	0	0.1	1	8	10	15

Figure 6. Akima data (J.ACM, 1970).



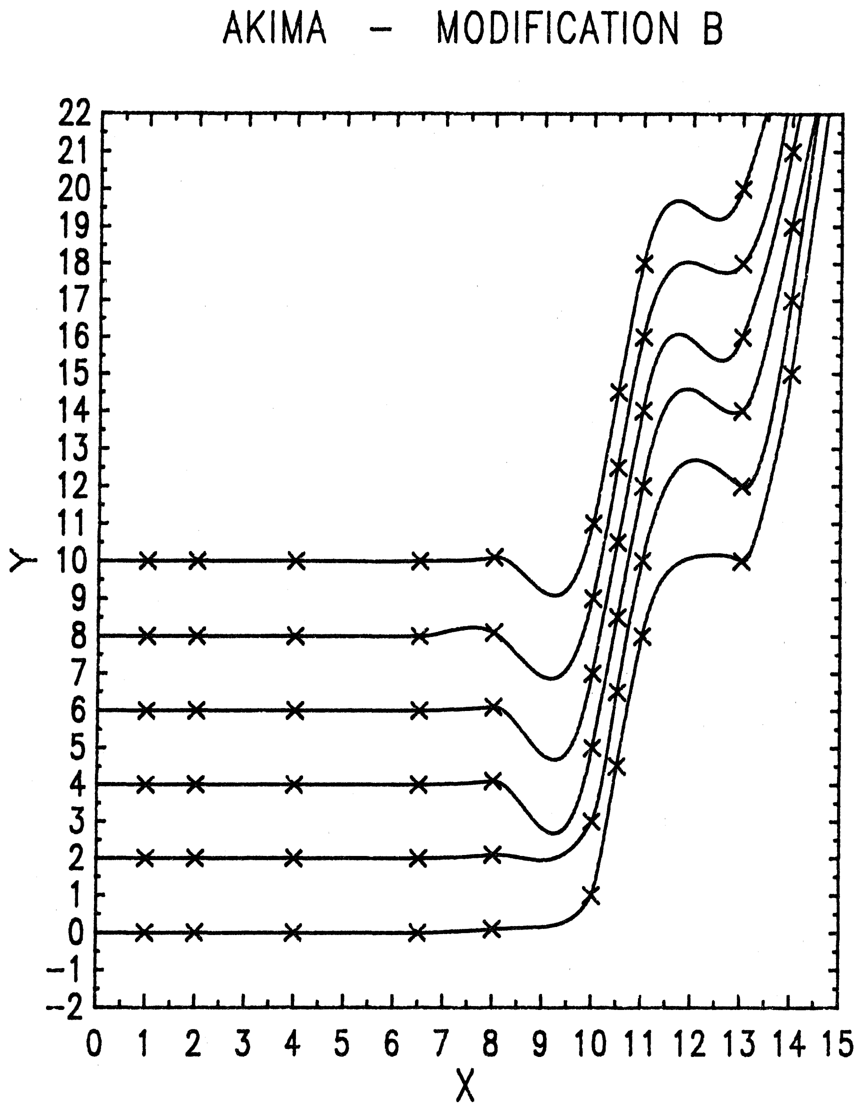
x =	1	2	4	6.5	8	10	11	13	14
y =	0	0	0	0	0.1	1	8	10	15

Figure 7. Akima data, Modification A.

osculatory methods), now in the interval between x = 8 and 10, are more pronounced in Figure 7 than in Figure 6. The third and fourth curves (resulting from the original A method and interim method) look good. The second curve from the bottom (resulting from the improved A method with n = 3) exhibits a small undulation in the interval between x = 8 and 10, but the bottom curve (resulting from the improved A method with n = 6) does not. In the bottom two curves (resulting from the improved A methods), the negative slope of the curve at x = 13 may look a little strange, but the second curve from the bottom looks good as a whole if monotonicity of the curve is not required. The bottom curve changes its direction so fast around x = 13 that it looks as if it were deflected at this point. Although the bottom curve behaves better than the second curve from the bottom in the interval between x = 8 and 10, the latter behaves better than the former around x = 13.

The data point set for Figure 8 is Modification B. It consists of the data points for Figure 7 (Modification A) and an additional point at x = 10.5, i.e., at the center of the line segment that has the steepest slope. With this additional data point, the top two curves (resulting from the two osculatory methods) are almost unaffected and remain unacceptable. The third and fourth curves (resulting from the original A method and interim method) are totally unacceptable; both the original A method and interim method join the two osculatory methods and produce large undulations in the interval between x = 8 and x = 10. The undulation in the interval between x = 8 and 10 in the second curve from the bottom (resulting from the improved A method with n = 3) is a little more pronounced in Figure 8 than in Figure 7. Even in the bottom curve (resulting from the improved A method with n = 6), a small undulation emerges in the same interval. The slope of the curve at x = 13 in the fourth curve is smaller than the same curve in Figure 7. The behaviors of the bottom two curves around x = 13 remain almost unchanged from Figure 7.

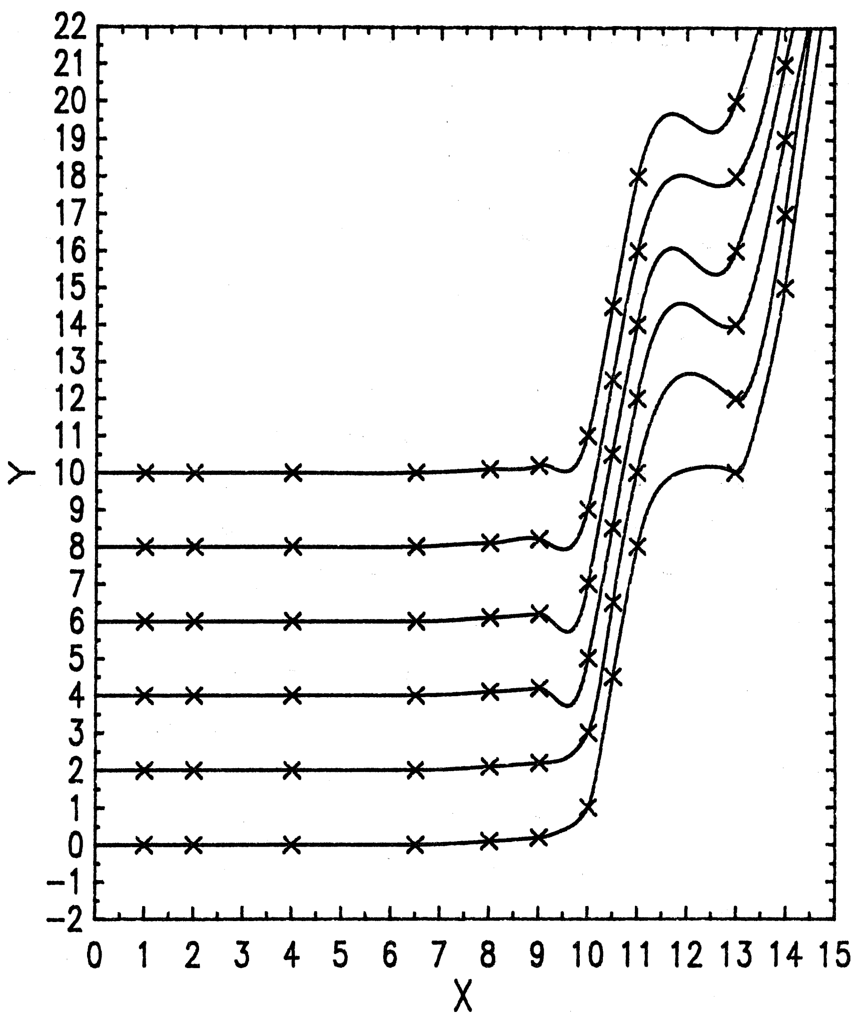
The data point set for Figure 9 is Modification C. It consists of the data points for Figure 8 (Modification B) and an additional point at x = 9. With this additional data point, the top four curves (resulting from the two osculatory methods, original A method, and interim method) are not improved to an acceptable level, while the bottom two curves (resulting from the improved A method) are improved considerably. Undulations that existed in the interval between x = 8 and 10 in the bottom two curves in Figure 8 are nonexistent any more in Figure 9. Improvement of the behavior of the curve by insertion of an



x =	1	2	4	6.5	8	10	10.5	11	13	14
y =	0	0	0	0	0.1	1	4.5	8	10	15

Figure 8. Akima data, Modification B.

AKIMA - MODIFICATION C



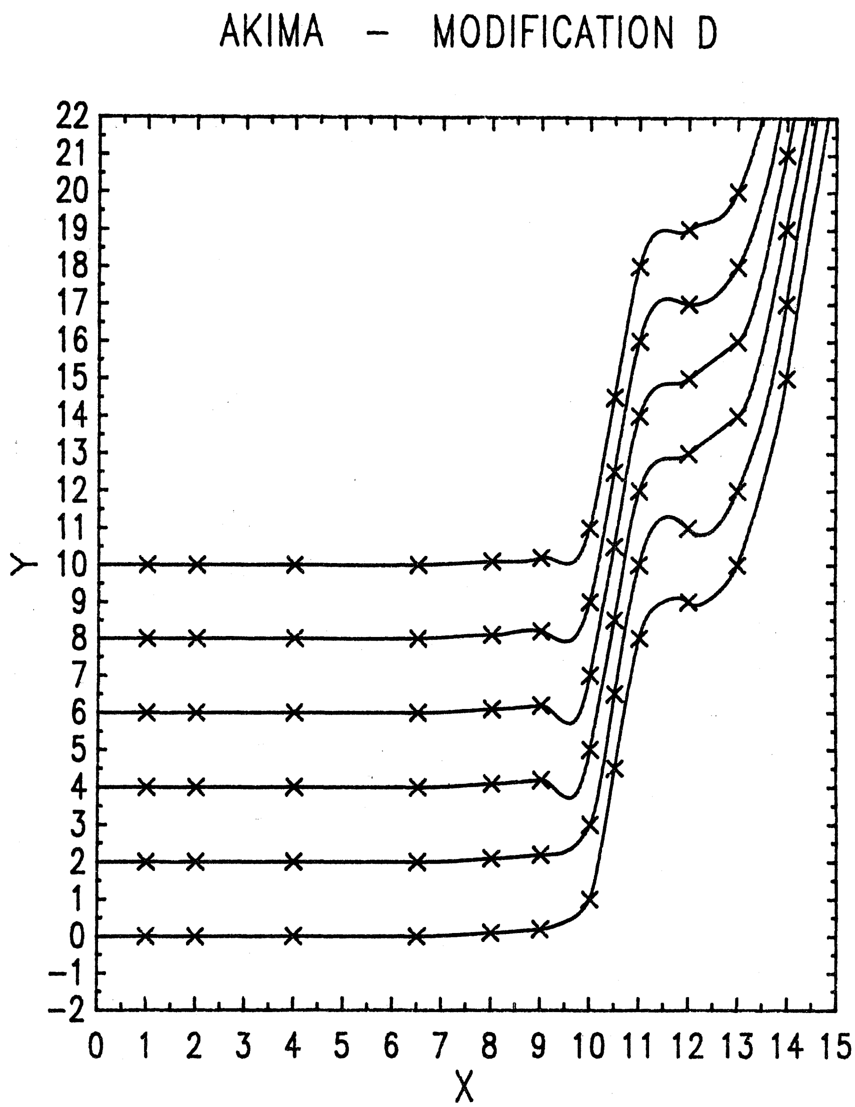
x =	1	2	4	6.5	8	9	10	10.5	11	13	14
y =	0	0	0	0	0.1	0.2	1	4.5	8	10	15

Figure 9. Akima data, Modification C.

additional data point in the troubled area is a desirable characteristic of an interpolation method, and it seems that the improved A method has that characteristic. As is expected, the slope of all curves at x = 13 are unaffected by the additional point at x = 9.

The data point set for Figure 10 is Modification D. It consists of the data points for Figure 9 (Modification C) and an additional point at x = 12. With this additional data point, interesting results are observed. The first curve (resulting from the original osculatory method) is degraded; an inflection point emerges in each side of the newly added data point. The second curve (resulting from the modified osculatory method) is improved; it does not have inflection points near the newly added data points. The third and fourth curves (resulting from the original A method and interim method) are degraded; a straight line segment is embedded in a generally curved line. The bottom two curves (resulting from the improved A method) are improved; the slopes of the curves at x = 13 look more natural than the same slopes in Figure 9. Again, improvement of the behavior of the curve by insertion of an additional data point is a desirable characteristic of an interpolation method, while degrading of the behavior by insertion of an additional data point is an undesirable characteristic of an interpolation method. In Figure 10, the bottom curve (resulting from the improved A method with n = 6) is better than the second curve from the bottom (with n = 3); a higher-degree polynomial works well without adverse side effect in this example.

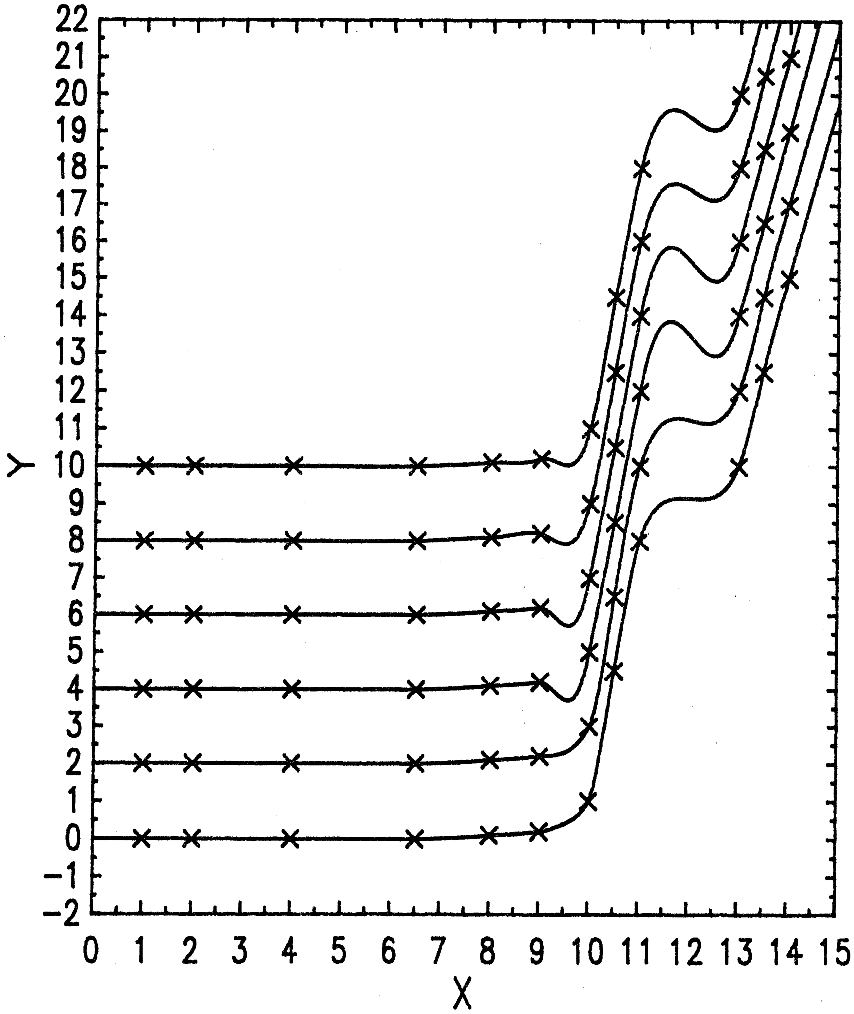
The data point set for Figure 11 is Modification E, which is another modification of Modification C but not a direct modification of D. It consists of the data points for Figure 9 (Modification C) and an additional point at x = 13.5. With this additional data point, the first and third curves (resulting from the original osculatory method and original A method) are almost unchanged from Figure 9, while all other curves are changed to some extent. Although it is hard to say that the second and fourth curves (resulting from the modified osculatory method and interim method) in Figure 11 are better than the same curves in Figure 9, we can say unequivocally that the bottom two curves (resulting from the improved A method) in Figure 11 are better than the same curves in Figure 9; the slopes of these curves at x = 13 look more natural than the same slopes in Figure 9. Figure 11 is another example in which a higher-degree polynomial works effectively.



x =	1	2	4	6.5	8	9	10	10.5	11	12	13	14
y =	0	0	0	0	0.1	0.2	1	4.5	8	9	10	15

Figure 10. Akima data, Modification D.

AKIMA - MODIFICATION E



x =	1	2	4	6.5	8	9	10	10.5	11	13	13.5	14
y =	0	0	0	0	0.1	0.2	1	4.5	8	10	12.5	15

Figure 11. Akima data, Modification E.

7. CONCLUSIONS

We have identified properties desired or required for an interpolation method, discussed their mutual compatibility, established our goals, and derived the guidelines for developing an improved method that will produce a good-looking curve when the method is used for smooth curve fitting. We have realized that one of our goals is to develop a method that has the accuracy of a third-degree (cubic) polynomial, i.e., a method that accurately interpolates the given set of data points when the data points lie on a cubic curve. We have also realized that the univariate interpolation method based on local procedures originally developed by Akima (1970) and called the original A method has some of the desired properties. We have improved the original A method in such a way that the improved method called the improved A method has the accuracy of a third-degree polynomial while retaining the desired properties of the original A method. As demonstrated in the examples, the improved A method generally yields a curve that looks much more natural than what will result from the original A method.

The improvement has been made in the procedure of estimating the first derivative of the interpolating function (or the slope of the curve) at each given data point. The improved A method first calculates four primary estimates for the first derivative, each as the first derivative of a third-degree polynomial fitted to a set of four consecutive data points. It calculates the final estimate of the first derivative as the weighted mean of the four primary estimates. The weight for each primary estimate is the reciprocal of the product of the measure of volatility in the ordinate and the measure of dispersion in the abscissa of the set of four data points. The sum of squares of the deviations of the ordinate values of the four data points from the straight line of least-square fit is used as the measure of volatility of the set of data points. The sum of squares of the distances from the data point in question of the remaining three data points in the set is used as the measure of dispersion of the set of data points.

Like the original A method, the improved A method uses a third-degree polynomial in an interval between each pair of data points as a default. In addition, we have also implemented possible use of a higher-degree polynomial for an interval between a pair of data points as an option. Although undulations are generally reduced by the use of a higher-degree polynomial as demonstrated in some examples, our other examples indicate that the use of a higher-degree polynomial sometimes distorts curves that would look good otherwise. A higher-degree polynomial option should therefore be exercised prudently and sparingly when the method is used for smooth curve fitting and naturalness of the resultant curve is of primary importance. Note that the use of a higher-degree polynomial will inevitably void the accuracy of a third-degree polynomial even though estimation of the first derivative at a data point is based on a third-degree polynomial.

Like the original A method, the improved A method does not always preserve monotonicity or convexity; we have not intended to preserve it in developing the improved A method. We propose the improved A method as a replacement for the original A method when natural appearance of the resultant curve is important; we do not propose it as a replacement for the F-C-B method (Fritsch and Carlson, 1980; Fritsch, 1982; Fritsch and Butland, 1984) or the M-R method (Roulier, 1980; McAllister and Roulier, 1981a, 1981b) when monotonicity or convexity must be preserved.

The improved A method can easily be implemented in a computer program. A Fortran subroutine subprogram that implements the improved A method is described in Appendix A with its listing.

Since the original A method has been improved without changing its basic concept, most remarks given to the original A method apply to the improved A method as well. Some remarks pertinent to proper application of the improved A method follow.

(1) The method does not smooth the data. In other words, the resultant curve passes through all the given data points if the method is applied to smooth curve fitting. Therefore, the method is applicable only when the precise y values are given or where the errors are negligible.

(2) As is true for any method of interpolation, the accuracy of the improved A method cannot be guaranteed, unless it is known that the given data points lie on a curve of a third-degree polynomial.

(3) Unless the option for a higher-degree polynomial is exercised, the method has the accuracy of a third-degree polynomial, i.e., the method gives exact results when y is a third-degree polynomial in x even when the y values of the data points are given at unequal intervals.

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

(5) The method is invariant under a linear-scale transformation of the coordinate system, i.e., $x = a\ u$ and $y = b\ v$, where a and b are nonzero constants. In other words, different scaling of the coordinates results in equivalent curves.

(6) The method is invariant under a linear transformation of the coordinate system represented by $x = a\ u$ and $y = b\ u + c\ v$, where a, b, and c are nonzero constants.

(7) The method is nonlinear. In other words, if $y_i = y^{(1)}(x_i) + y^{(2)}(x_i)$ for all i, the interpolated values do not, in general, satisfy $y(x) = y^{(1)}(x) + y^{(2)}(x)$.

(8) The method will produce a periodic curve (such as the one for a function of an angle) from a set of periodic data points that covers a complete cycle if three additional data points corresponding to the preceding or following cycle are supplied on each side of the given data point set.

(9) The method requires only straightforward procedures. No problem concerning computational stability or convergence exists in the application of the method.

8. ACKNOWLEDGMENTS

The author is grateful to Dr. Frederick N. Fritsch of the Lawrence Livermore National Laboratory, Livermore, California, for his useful information on various methods of shape-preserving interpolation; to George A. Hufford and A. Donald Spaulding of the Institute for Telecommunication Sciences of the National Telecommunications and Information Administration of the U.S. Department of Commerce, Boulder, Colorado, for their review of the manuscript of this report and useful discussions on the subject of this report; and to Gregory R. Hand of the same Institute for his helpful assistance in plotting the curves in the examples.

9. REFERENCES

Ackland, T. G. (1915), On osculatory interpolation, where the given values of the function are at unequal intervals, J. Inst. Actuar. 49, pp. 369-375.

Akima, H. (1970), A new method of interpolation and smooth curve fitting based on local procedures, J. ACM 17, pp. 589-602, October.

Akima, H. (1972), Algorithm 433: Interpolation and smooth curve fitting based on local procedures, Comm. ACM 15, pp. 914-918, October.

Carnahan, B., and J. O. Wilkes (1973), Digital Computing and Numerical Methods (With FORTRAN-IV, WATFOR, AND WATFIV Programming) (Wiley, New York). (pp. 321-322, in particular).

Cline, A. K. (1974), Scalar- and planar-valued curve fitting using splines under tension, Comm. ACM 17, pp. 218-220, April.

Davis, P. J. (1975), Interpolation and Approximation (Dover, New York). (Chapters 2 and 3, in particular)

Fritsch, F. N. (1982), Piecewise cubic Hermite interpolation package, SIAM 30th Anniversary Meeting, Stanford, California, July.

Fritsch, F. N., and J. Butland (1984), A method for constructing local monotone piecewise cubic interpolants, SIAM J. Sci. Stat. Comp. 5, pp. 300-304.

Fritsch, F. N., and R. E. Carlson (1980), Monotone piecewise cubic interpolation, SIAM J. Numer. Analysis 17, pp. 238-246, April.

Gregory, J. (1985), A review of curve interpolation with shape control, IMA Conference on Algorithms for Approximation of Functions and Data, Shrivenham, England, July.

Greville, T. N. E. (1967), Spline functions, interpolation, numerical quadrature. In Mathematical Methods for Digital Computers, Vol. 2, A. Ralston and H. S. Wilf (Eds.) (Wiley, New York), Chapter 8.

Hildebrand, F. B. (1956), Introduction to Numerical Analysis (McGraw-Hill, New York). (Chapters 2, 3, and 4, in particular)

IMSL (International Mathematical and Statistical Libraries, Inc.) (1979), IMSL Library Reference Manual, Edition 7, Chapter I, January.

Karup, J. (1899), On a new mechanical method of graduation. In Transactions of the Second International Actuarial Congress (C. and E. Layton, London), pp. 78-109.

McAllister, D. F., and J. A. Roulier (1981a), An algorithm for computing a shape-preserving osculatory quadratic spline, ACM Trans. Math. Software 7, pp. 331-347, September.

McAllister, D. F., and J. A. Roulier (1981b), Algorithm 574: Shape-preserving osculatory quadratic spline [E1,E2], ACM Trans. Math. Software 7, pp. 384-386, September.

Roulier, J. A. (1980), Constrained interpolation, SIAM Sci. Stat. Comput. 1, pp. 333-344, September.

APPENDIX A: THE UVIPIA SUBROUTINE SUBPROGRAM

The UVIPIA subroutine subprogram implements the improved A method developed in the text. Possible use of a higher-degree polynomial is also implemented as the user's option. Unless the option of the use of a higher-degree polynomial is exercised by the user, the method has the accuracy of a third-degree (cubic) polynomial even when the input data points are given at unequal intervals.

The UVIPIA subroutine subprogram is written in ANSI (American National Standards Institute) Standard Fortran (Publication X3.9-1978, ANSI, 345 East 47th Street, New York, NY 10017).

Fortran listing of the UVIPIA subroutine subprogram follows. The listing includes a test program in front of the subprogram. The user information of the subprogram including the description of the input and output arguments is given in the beginning of the subprogram.

```
      PROGRAM  TTUVIP
C   ***   TEST PROGRAM FOR THE UVIPIA SUBROUTINE   ***
C THIS PROGRAM TESTS THE UVIPIA SUBROUTINE BY CALLING UVIPIA
C WITH TWO NP VALUES IN A SEQUENCE.  (NP IS ONE OF THE INPUT
C ARGUMENTS OF UVIPIA AND REPRESENTS THE DEGREE OF THE POLY-
C NOMIAL FOR THE INTERVAL BETWEEN EACH PAIR OF DATA POINT.)
C WITH EACH NP VALUE, THIS PROGRAM CALLS UVIPIA IN THREE WAYS:
C (1) IN REPEATED CALLS WITH INTERPOLATION AT A POINT IN
C     EACH CALL,
C (2) IN ONE CALL WITH INTERPOLATION AT ALL POINTS, AND
C (3) IN ONE CALL AS IN (2) BUT WITH INVERTED DATA SET.
C THE PROGRAM THEN COMPARES THE RESULTS WITH THE EXPECTED VALUES
C PRESTORED IN THE PROGRAM AND PRINTS OUT THE DIFFERENCES IN TWO
C PAGES.
C IF ALL ENTRIES IN THE LAST THREE COLUMNS ON EACH PAGE OF THE
C PRINTOUT ARE ALL ZEROS, UVIPIA IS CONSIDERED TO BE WORKING AS
C EXPECTED.  OTHERWISE, SOMETHING IS WRONG IN UVIPIA AND/OR THE
C TEST PROGRAM ITSELF.
C SPECIFICATION STATEMENTS
      PARAMETER   (NNP=2, NDP=10, NIP=31)
      DIMENSION   NP(NNP)
      DIMENSION
     1  XD(NDP),XD3(NDP),YD(NDP),YD3(NDP),
     2  XI(NIP),XI3(NIP),YI(NIP,NNP),YI1(NIP),YI2(NIP),YI3(NIP)
      DATA  NP/ 3, 6/
      DATA  XD/
     1   1.0, 2.0, 4.0, 6.5, 8.0,10.0,10.5,11.0,13.0,14.0/
      DATA  YD/
     1   4*0.0,     0.1, 1.0, 4.5, 8.0,10.0,15.0/
      DATA  XI/
     1   0.0, 0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5,
     2   4.0, 4.5, 5.0, 5.5, 6.0, 6.5, 7.0, 7.5,
     3   8.0, 8.5, 9.0, 9.5,10.0,10.5,11.0,11.5,
     4  12.0,12.5,13.0,13.5,14.0,14.5,15.0/
      DATA  YI/
     1   8*0.0,
     2   6*0.0, 0.015, 0.052,
     3   0.100, 0.036,-0.045, 0.172, 1.000, 4.500, 8.000,10.075,
     4  10.705,10.483,10.000,11.204,15.000,19.767,24.533,
     1   8*0.0,
     2   6*0.0, 0.020, 0.057,
     3   0.100, 0.134, 0.166, 0.314, 1.000, 4.500, 8.000, 9.689,
     4  10.101,10.180,10.000,11.663,15.000,19.767,24.533/
C CALCULATION
C DATA PREPARATION
   10 DO 11  I=1,NDP
        II=NDP+1-I
        XD3(II)=XD(1)*XD(NDP)-XD(I)
        YD3(II)=YD(I)
   11 CONTINUE
      DO 12  K=1,NIP
        XI3(K)=XD(1)*XD(NDP)-XI(K)
   12 CONTINUE
C INTERPOLATION
```

```
   20 DO 69  INP=1,NNP
        NPI=NP(INP)
        DO 21  K=1,NIP
          CALL UVIPIA(NPI,NDP,XD,YD,1,XI(K),YI1(K))
   21   CONTINUE
        CALL UVIPIA(NPI,NDP,XD,YD,NIP,XI,YI2)
        CALL UVIPIA(NPI,NDP,XD3,YD3,NIP,XI3,YI3)
C PRINTING OF THE RESULTS
   50   WRITE (*,99050) NPI
        DO 59  K=1,NIP
          DYI1=YI1(K)-YI(K,INP)
          DYI2=YI2(K)-YI(K,INP)
          DYI3=YI3(K)-YI(K,INP)
          IF(MOD(K,4).EQ.2)    WRITE (*,99051)
          IF(K.LE.NDP)  THEN
            WRITE (*,99052) XD(K),YD(K),
     1                      XI(K),YI(K,INP),YI1(K),DYI1,DYI1,DYI3
          ELSE
            WRITE (*,99053) XI(K),YI(K,INP),YI1(K),DYI1,DYI1,DYI3
          END IF
   59   CONTINUE
   69 CONTINUE
      WRITE (*,99080)
      STOP
C FORMAT STATEMENTS
99050 FORMAT('1',
     1 'TTUVIP',14X,'TEST PROGRAM FOR UVIPIA',29X,'NP =',I2////
     2 6X,'DATA POINT',9X,'INTERPOLATED POINTS ',
     3 8X,'      DIFFERENCE      '//
     4 6X,'XD       YD',9X,'XI        YI       YI1',
     5 8X,'DYI1     DYI2    DYI3'/)
99051 FORMAT(1X)
99052 FORMAT(1X,2F9.3,2X,3F9.3,2X,3F9.3)
99053 FORMAT(19X,2X,3F9.3,2X,3F9.3)
99080 FORMAT('1')
      END
```

```
      SUBROUTINE  UVIPIA(NP,ND,XD,YD,NI,XI, YI)
C  ***  UNIVARIATE INTERPOLATION (IMPROVED AKIMA METHOD)  ***
C THIS SUBROUTINE PERFORMS UNIVARIATE INTERPOLATION.  IT IS
C BASED ON THE IMPROVED AKIMA METHOD.
C IN THIS METHOD, THE INTERPOLATING FUNCTION IS A PIECEWISE
C FUNCTION COMPOSED OF A SET OF POLYNOMIALS OF THE DEGREE SPECI-
C FIED BY THE USER, APPLICABLE TO SUCCESSIVE INTERVALS OF THE
C GIVEN DATA POINTS.  THE METHOD USES SEVEN DATA POINTS FOR
C ESTIMATING THE FIRST DERIVATIVE OF THE FUNCTION (OR THE SLOPE
C OF THE CURVE) AT EACH DATA POINT.  IT HAS THE ACCURACY OF A
C THIRD-DEGREE POLYNOMIAL IF THE DEGREE OF THE POLYNOMIALS FOR
C THE INTERPOLATING FUNCTION IS SET TO THREE.
C THE INPUT ARGUMENTS ARE
C   NP = DEGREE OF THE POLYNOMIALS FOR THE INTERPOLATING
C        FUNCTION,
C   ND = NUMBER OF INPUT DATA POINTS
C        (MUST BE EQUAL TO 2 OR GREATER),
C   XD = ARRAY OF DIMENSION ND, CONTAINING THE ABSCISSAS OF
C        THE INPUT DATA POINTS
C        (MUST BE IN A MONOTONIC INCREASING ORDER),
C   YD = ARRAY OF DIMENSION ND, CONTAINING THE ORDINATES OF
C        THE INPUT DATA POINTS,
C   NI = NUMBER OF POINTS FOR WHICH INTERPOLATION IS DESIRED
C        (MUST BE EQUAL TO 1 OR GREATER),
C   XI = ARRAY OF DIMENSION NI, CONTAINING THE ABSCISSAS OF
C        THE DESIRED POINTS.
C IF AN INTEGER VALUE SMALLER THAN 3 IS GIVEN AS THE NP ARGUMENT
C TO THIS SUBROUTINE, THIS SUBROUTINE ASSUMES NP = 3.
C THE XI ARRAY ELEMENTS NEED NOT BE MONOTONIC, BUT THIS SUBROU-
C TINE INTERPOLATES FASTER IF THE ELEMENTS ARE GIVEN IN A MONO-
C TONIC (EITHER INCREASING OR DECREASING) ORDER.
C THE OUTPUT ARGUMENT IS
C   YI = ARRAY OF DIMENSION NI, WHERE THE ORDINATES OF THE
C        DESIRED POINTS ARE TO BE STORED.
C SPECIFICATION STATEMENTS
      DIMENSION   XD(*),YD(*),XI(*), YI(*)
      DIMENSION   X(3),Y(3)
      EQUIVALENCE (X1,X(1)),(X2,X(2)),(X3,X(3)),
     1            (Y1,Y(1)),(Y2,Y(2)),(Y3,Y(3))
C ERROR CHECK
   10 IF(ND.LE.1)    GO TO 90
      IF(NI.LE.0)    GO TO 91
      DO 11  ID=2,ND
        IF(XD(ID).LE.XD(ID-1))      GO TO 92
   11 CONTINUE
C BRANCHING OFF SPECIAL CASES
      IF(ND.LE.4)    GO TO 50
C GENERAL CASE  --  FIVE DATA POINTS OR MORE
C CALCULATION OF THE EPSLN VALUE
   20 YDMN=YD(1)
      YDMX=YD(1)
      DO 21  ID=2,ND
        YDMN=MIN(YDMN,YD(ID))
        YDMX=MAX(YDMX,YD(ID))
```

```
   21 CONTINUE
      EPSLN=((YDMX-YDMN)**2)*1.0E-12
C CALCULATION OF SOME LOCAL VARIABLES
      NPO=MAX(3,NP)
      NPM1=NPO-1
      RENPM1=NPM1
      RENNM2=NPO*(NPO-2)
C MAIN CALCULATION FOR THE GENERAL CASE
   30 IINTPV=-1
      DO 39  II=1,NI
        XII=XI(II)
C LOCATES THE INTERVAL THAT INCLUDES THE POINT IN QUESTION.
        IF(XII.LE.XD(1))  THEN
          IINT=0
        ELSE IF(XII.LT.XD(ND))  THEN
          IDMN=1
          IDMX=ND
          IDMD=(IDMN+IDMX)/2
   31     IF(XII.GE.XD(IDMD))  THEN
            IDMN=IDMD
          ELSE
            IDMX=IDMD
          END IF
          IDMD=(IDMN+IDMX)/2
          IF(IDMD.GT.IDMN)    GO TO 31
          IINT=IDMD
        ELSE
          IINT=ND
        END IF
C INTERPOLATION OR EXTRAPOLATION IN ONE OF THE THREE SUBCASES
        IF(IINT.LE.0)  THEN
C SUBCASE 1  --  WHEN XI(II) IS EQUAL TO XD(1) OR LESS
          IF(IINT.NE.IINTPV)  THEN
            IINTPV=IINT
            X0=XD(1)
            X1=XD(2)-X0
            X2=XD(3)-X0
            X3=XD(4)-X0
            Y0=YD(1)
            Y1=YD(2)-Y0
            Y2=YD(3)-Y0
            Y3=YD(4)-Y0
            DLT=X1*X2*X3*(X2-X1)*(X3-X2)*(X3-X1)
            A1=(((X2*X3)**2)*(X3-X2)*Y1
     1         +((X3*X1)**2)*(X1-X3)*Y2
     2         +((X1*X2)**2)*(X2-X1)*Y3)/DLT
          END IF
          YI(II)=Y0+A1*(XII-X0)
C END OF SUBCASE 1
        ELSE IF(IINT.GE.ND)  THEN
C SUBCASE 2  --  WHEN XI(II) IS EQUAL TO XD(ND) OR GREATER
          IF(IINT.NE.IINTPV)  THEN
            IINTPV=IINT
            X0=XD(ND)
```

```
                X1=XD(ND-1)-X0
                X2=XD(ND-2)-X0
                X3=XD(ND-3)-X0
                Y0=YD(ND)
                Y1=YD(ND-1)-Y0
                Y2=YD(ND-2)-Y0
                Y3=YD(ND-3)-Y0
                DLT=X1*X2*X3*(X2-X1)*(X3-X2)*(X3-X1)
                A1=(((X2*X3)**2)*(X3-X2)*Y1
     1             +((X3*X1)**2)*(X1-X3)*Y2
     2             +((X1*X2)**2)*(X2-X1)*Y3)/DLT
              END IF
              YI(II)=Y0+A1*(XII-X0)
C END OF SUBCASE 2
            ELSE
C SUBCASE 3  --  WHEN XI(II) IS BETWEEN XD(1) AND XD(ND)
              IF(IINT.NE.IINTPV)  THEN
C CALCULATION OF THE COEFFICIENTS OF THE THIRD-DEGREE POLY-
C NOMIAL OR THE FACTORS FOR THE TWO BASIS POLYNOMIALS
                IINTPV=IINT
C CALCULATION OF THE ESTIMATES OF THE DERIVATIVES
                DO 37  IEPT=1,2
                  ID0=IINT+IEPT-1
                  X0=XD(ID0)
                  Y0=YD(ID0)
                  SMPEF=0.0
                  SMWTF=0.0
                  SMPEI=0.0
                  SMWTI=0.0
                  DO 36  IPE=1,4
C PRIMARY ESTIMATE OF THE DERIVATIVE
                    IF(IPE.EQ.1)  THEN
                      ID1=ID0-3
                      ID2=ID0-2
                      ID3=ID0-1
                    ELSE IF(IPE.EQ.2)  THEN
                      ID1=ID0+1
                    ELSE IF(IPE.EQ.3)  THEN
                      ID2=ID0+2
                    ELSE
                      ID3=ID0+3
                    END IF
                    IF(ID1.LT.1.OR.ID2.LT.1.OR.ID3.LT.1.OR.
     1                 ID1.GT.ND.OR.ID2.GT.ND.OR.ID3.GT.ND)
     2                   GO TO 36
                    X1=XD(ID1)-X0
                    X2=XD(ID2)-X0
                    X3=XD(ID3)-X0
                    Y1=YD(ID1)-Y0
                    Y2=YD(ID2)-Y0
                    Y3=YD(ID3)-Y0
                    DLT=X1*X2*X3*(X2-X1)*(X3-X2)*(X3-X1)
                    PE=(((X2*X3)**2)*(X3-X2)*Y1
     1                 +((X3*X1)**2)*(X1-X3)*Y2
```

```
     2              +((X1*X2)**2)*(X2-X1)*Y3)/DLT
C WEIGHT FOR THE PRIMARY ESTIMATE
               SX=X1+X2+X3
               SY=Y1+Y2+Y3
               SXX=X1*X1+X2*X2+X3*X3
               SXY=X1*Y1+X2*Y2+X3*Y3
               DNM=4.0*SXX-SX*SX
               B0=(SXX*SY-SX*SXY)/DNM
               B1=(4.0*SXY-SX*SY)/DNM
               DY0=-B0
               DY1=Y1-(B0+B1*X1)
               DY2=Y2-(B0+B1*X2)
               DY3=Y3-(B0+B1*X3)
               VAR=DY0*DY0+DY1*DY1+DY2*DY2+DY3*DY3
               IF(VAR.GT.EPSLN)  THEN
                 WT=1.0/(VAR*SXX)
                 SMPEF=SMPEF+PE*WT
                 SMWTF=SMWTF+WT
               ELSE
                 SMPEI=SMPEI+PE
                 SMWTI=SMWTI+1.0
               END IF
   36        CONTINUE
C FINAL ESTIMATE OF THE DERIVATIVE
             IF(SMWTI.LT.0.5)  THEN
               YP=SMPEF/SMWTF
             ELSE
               YP=SMPEI/SMWTI
             END IF
             IF(IEPT.EQ.1)  THEN
               YP0=YP
             ELSE
               YP1=YP
             END IF
   37      CONTINUE
C COEFFICIENTS OF THE THIRD-DEGREE POLYNOMIAL OR THE FACTORS FOR
C THE BASIC POLYNOMIALS
           X0=XD(IINT)
           DX=XD(IINT+1)-X0
           Y0=YD(IINT)
           DY=YD(IINT+1)-Y0
           IF(NP0.LE.3)  THEN
C COEFFICIENTS OF THE THIRD-DEGREE POLYNOMIAL WHEN NP.LE.3
             A1=YP0
             YP1=YP1-YP0
             YP0=YP0-DY/DX
             A2=-(3.0*YP0+YP1)/DX
             A3= (2.0*YP0+YP1)/(DX*DX)
           ELSE
C FACTORS FOR THE POLYNOMIALS WHEN NP.GT.3
             T0=YP0*DX-DY
             T1=YP1*DX-DY
             AA0= (T0+RENPM1*T1)/RENNM2
             AA1=-(RENPM1*T0+T1)/RENNM2
```

```
            END IF
          END IF
C EVALUATION OF THE YI VALUE
          IF(NPO.LE.3)  THEN
            XX=XII-X0
            YI(II)=Y0+XX*(A1+XX*(A2+XX*A3))
          ELSE
            U=(XII-X0)/DX
            UC=1.0-U
            V=AA0*U*(-1.0+U**NPM1)+AA1*UC*(-1.0+UC**NPM1)
            YI(II)=Y0+DY*U+V
          END IF
C END OF SUBCASE 3
        END IF
   39 CONTINUE
      RETURN
C END OF GENERAL CASE
C SPECIAL CASES  --  FOUR DATA POINTS OR LESS
C PRELIMINARY PROCESSING FOR SPECIAL CASES
   50 X0=XD(1)
      Y0=YD(1)
      DO 51  ID=2,ND
        X(ID-1)=XD(ID)-X0
        Y(ID-1)=YD(ID)-Y0
   51 CONTINUE
      GO TO (90,60,70,80), ND
C SPECIAL CASE 1  --  TWO DATA POINTS
   60 A1=Y1/X1
      DO 61  II=1,NI
        YI(II)=Y0+A1*(XI(II)-X0)
   61 CONTINUE
      RETURN
C SPECIAL CASE 2  --  THREE DATA POINTS
   70 DLT=X1*X2*(X2-X1)
      A1=(X2*X2*Y1-X1*X1*Y2)/DLT
      A2=(X1*Y2-X2*Y1)/DLT
      A12=2.0*A2*X2+A1
      DO 71  II=1,NI
        XX=XI(II)-X0
        IF(XX.LE.0.0)  THEN
          YI(II)=Y0+A1*XX
        ELSE IF(XX.LT.X2) THEN
          YI(II)=Y0+XX*(A1+XX*A2)
        ELSE
          YI(II)=Y0+Y2+A12*(XX-X2)
        END IF
   71 CONTINUE
      RETURN
C SPECIAL CASE 3  --  FOUR DATA POINTS
   80 DLT=X1*X2*X3*(X2-X1)*(X3-X2)*(X3-X1)
      A1=(((X2*X3)**2)*(X3-X2)*Y1
     1   +((X3*X1)**2)*(X1-X3)*Y2
     2   +((X1*X2)**2)*(X2-X1)*Y3)/DLT
      A2=(X2*X3*(X2*X2-X3*X3)*Y1
```

```
     1   +X3*X1*(X3*X3-X1*X1)*Y2
     2   +X1*X2*(X1*X1-X2*X2)*Y3)/DLT
      A3=(X2*X3*(X3-X2)*Y1
     1   +X3*X1*(X1-X3)*Y2
     2   +X1*X2*(X2-X1)*Y3)/DLT
      A13=(3.0*A3*X3+2.0*A2)*X3+A1
      DO 81  II=1,NI
        XX=XI(II)-X0
        IF(XX.LE.0.0)  THEN
          YI(II)=Y0+A1*XX
        ELSE IF(XX.LT.X3) THEN
          YI(II)=Y0+XX*(A1+XX*(A2+XX*A3))
        ELSE
          YI(II)=Y0+Y3+A13*(XX-X3)
        END IF
   81 CONTINUE
      RETURN
C END OF SPECIAL CASES
C ERROR EXIT
   90 WRITE (*,99090) ND
      RETURN
   91 WRITE (*,99091) NI
      RETURN
   92 WRITE (*,99092) ID,XD(ID-1),XD(ID)
      RETURN
C FORMAT STATEMENTS
99090 FORMAT(1X/ ' ***   INSUFFICIENT DATA POINTS.'
     1  7X,'ND =',I3/
     2  ' ERROR DETECTED IN ROUTINE   UVIPIA'/)
99091 FORMAT(1X/ ' ***   NO DESIRED POINTS.'
     1  7X,'NI =',I3/
     2  ' ERROR DETECTED IN ROUTINE   UVIPIA'/)
99092 FORMAT(1X/ ' ***   TWO DATA POINTS IDENTICAL OR OUT OF ',
     1  'SEQUENCE.'/
     2  7X,'ID, XD(ID-1), XD(ID) =',I5,2F10.3/
     3  ' ERROR DETECTED IN ROUTINE   UVIPIA'/)
      END
```

APPENDIX B: INTERPOLATING FUNCTIONS IN A UNIT INTERVAL

In this appendix we examine the behavior of some interpolating functions in a unit interval, i.e., an interval between $x = 0$ and $x = 1$. Without loss of generality, we assume that $y(0) = 0$ and $y(1) = 0$. We consider only the interpolating functions that can be represented in a symmetric form with respect to x and 1-x, i.e., as the linear combination of f(x) and f(1-x), where f(x) is a basis function.

We consider, as a basis function, a function of x, f(x), that satisfies the following conditions:

$$x = 0: \quad f(0) = 0, \quad f'(0) = -t$$

$$x = 1: \quad f(1) = 0, \quad f'(1) = 1, \qquad \text{(B-1)}$$

$$0 \leq x \leq 1: \quad f''(x) \geq 0,$$

where t is a positive constant smaller than unity. Next, we consider, as an interpolating function, a function of x, y(x), that is a linear combination of f(x) and f(1-x) and is represented by

$$y(x) = C_0 \, f(x) + C_1 \, f(1-x). \qquad \text{(B-2)}$$

The C coefficients can be determined from the first derivatives of y(x) at $x = 0$ and $x = 1$, i.e., $y'(0)$ and $y'(1)$, by

$$C_0 = [t \, y'(0) + y'(1)] \, / \, (1 - t^2),$$

and (B-3)

$$C_1 = -[y'(0) + t \, y'(1)] \, / \, (1 - t^2).$$

It is clear from these relations that y(x) represented by (B-2) satisfies the conditions that $y(0) = 0$ and $y(1) = 0$.

Although we can consider an infinite number of functions as the basis function, we consider only three functions here. Each function is listed with its first derivative and the t value (which is minus the first derivative value at $x = 0$).

(a) nth-degree polynomial

$$f(x) = (x^n - x) / (n-1),$$

$$f'(x) = (n\, x^{n-1} - 1) / (n-1), \qquad \text{(B-4)}$$

$$t = 1 / (n-1).$$

(b) nth-degree and second-degree polynomials

$$f(x) = (x^n + x^2 - 2x) / n$$

$$= [(x^n - x) + (x^2 - x)] / n,$$

$$f'(x) = (n\, x^{n-1} + 2x - 2) / n, \qquad \text{(B-5)}$$

$$t = 2 / n.$$

(c) Exponential function

$$f(x) = \{[\exp(ax) - 1] - [\exp(a) - 1]\, x\} / b$$

$$= \{a^2(x^2 - x)/2! + a^3(x^3 - x)/3! + a^4(x^4 - x)/4! + \ldots\} / b,$$

$$f'(x) = [a \exp(ax) - \exp(a) + 1] / b, \qquad \text{(B-6)}$$

$$t = [\exp(a) - (1 + a)] / b,$$

$$b = (a - 1) \exp(a) + 1.$$

Perhaps the function in (a) is the simplest form we can ever consider as the basis function for our purpose. The function in (b) is a simple modification of (a) and is a sum of two terms, each similar to (a). When n equals three, two interpolating functions based on the two basis functions in (a) and (b) coincide with each other and reduce to a common third-degree polynomial which is the cubic Hermite interpolant represented by

$$y(x) = (x^3 - 2x^2 + x)\, y'(0) + (x^3 - x^2)\, y'(1). \tag{B-7}$$

The function in (c) is based on an exponential function but it can also be represented as a sum of an infinite number of terms, each similar to (a).

Before we proceed, we introduce, as the fourth interpolating function, another function that does not fall in the same category as (a) through (c). The fourth interpolating function is a piecewise function composed of two second-degree polynomials joined together smoothly at x = 0.5, the center of the unit interval. It is represented as follows:

(d) Piecewise second-degree polynomials

$$\begin{aligned} 0 \le x \le 0.5:&\quad y(x) = a_1 x + a_2 x^2, \\ 0.5 \le x \le 1:&\quad y(x) = b_1(1 - x) + b_2(1 - x)^2, \end{aligned} \tag{B-8}$$

where

$$\begin{aligned} a_1 &= y'(0), \\ a_2 &= -[3y'(0) + y'(1)] / 2, \\ b_1 &= -y'(1), \\ \text{and} \quad b_2 &= [y'(0) + 3y'(1)] / 2. \end{aligned} \tag{B-9}$$

This function has an advantage over the third-degree polynomial (i.e., the interpolating function based on the basis function in (a) or (b) with n = 3) with respect to convexity, i.e., the property that y"(x) does not change its sign in the whole interval. Curves of the former are convex when y'(0)/y'(1) lies between -3 and -1/3, while curves of the latter are convex only when y'(0)/y'(1) lies between -2 and -1/2. The interpolating function in (d) is used by McAllister and Roulier (ACM Trans. Math. Software 7, pp. 331-347 and pp. 384-386, September 1981) and by others.

We now present the behavior of these four functions graphically in Figures B-1 through B-13. For simplicity, we present the behavior of the functions in a normalized form with respect to the first derivative of the function at x = 1, i.e., we assume that y'(1) = 1 and present the behavior of y(x) for various values of y'(0) between -1 and +1. In each figure, we plot 21 curves that correspond to the y'(0) values from +1.0 to -1.0 with a -0.1 step from the top to the bottom. The y'(1) value is unity for all curves. (Note that the scaling of the y axis is double that of the x axis.) The center curve that corresponds to y'(0) = 0 is plotted in a heavy line.

Curves for the function y(x) based on an nth-degree polynomial in (a) are plotted for the n values equal to 3, 4, 5, 6, 8, and 10 in Figures B-1 through B-6. These figures indicate that undulations are reduced by increasing n but each curve approaches a straight line at the same time.

Curves for the function y(x) based on an nth-degree and second-degree polynomials in (b) are plotted for n equal to 6 and 10 in Figures B-7 and B-8. If the curves for n = 3 were plotted, they would coincide with those plotted in Figure B-1. A curve of this function for an n value is very close to the curve of the function in (a) for a slightly smaller value of n. Curves of this function for n = 6 plotted in Figure B-7 are very close to those plotted in Figure B-3 that plots the function in (a) for n = 5. Curves of this function for n = 10 plotted in Figure B-8 almost fall between their respective curves plotted in Figures B-5 and B-6 that plot the function in (a) for n = 8 and 10.

Curves for the function y(x) based on an exponential function in (c) are plotted for the a value equal to 1, 2, 5 and 10 in Figures B-9 and B-12. These figures indicate that a curve of this function for an a value is very close to the curve of the nth-degree polynomial, i.e., the function in (a), for some n value. Note, however, that calculation time required for the exponential function is longer than that for the nth-degree polynomial.

Curves for the piecewise function composed of two second-degree polynomials described in (d) are plotted in Figure B-13. We notice that a few curves for the value of y'(0) around -0.2, look unnatural. Also, the "amplitude" of the top curve, i.e., the curve for y'(0) = +1, is too large in comparison with the top curve for the third-degree polynomial in Figure B-1. Because of these reasons, use of this piecewise function is not advisable regardless of the advantage of this function over the third-degree polynomial with respect to the property of convexity.

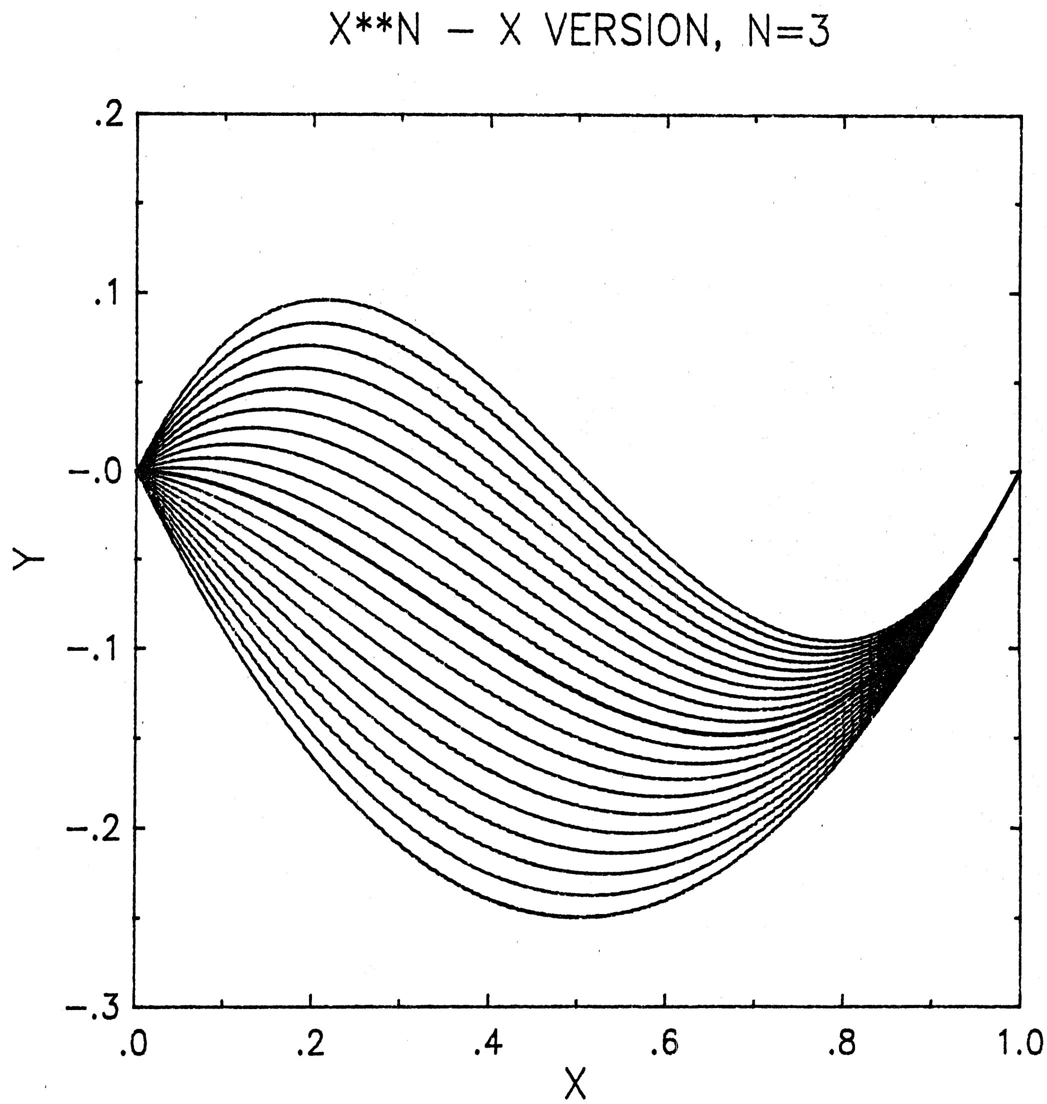


Figure B-1. Function based on an nth-degree polynomial with n = 3.

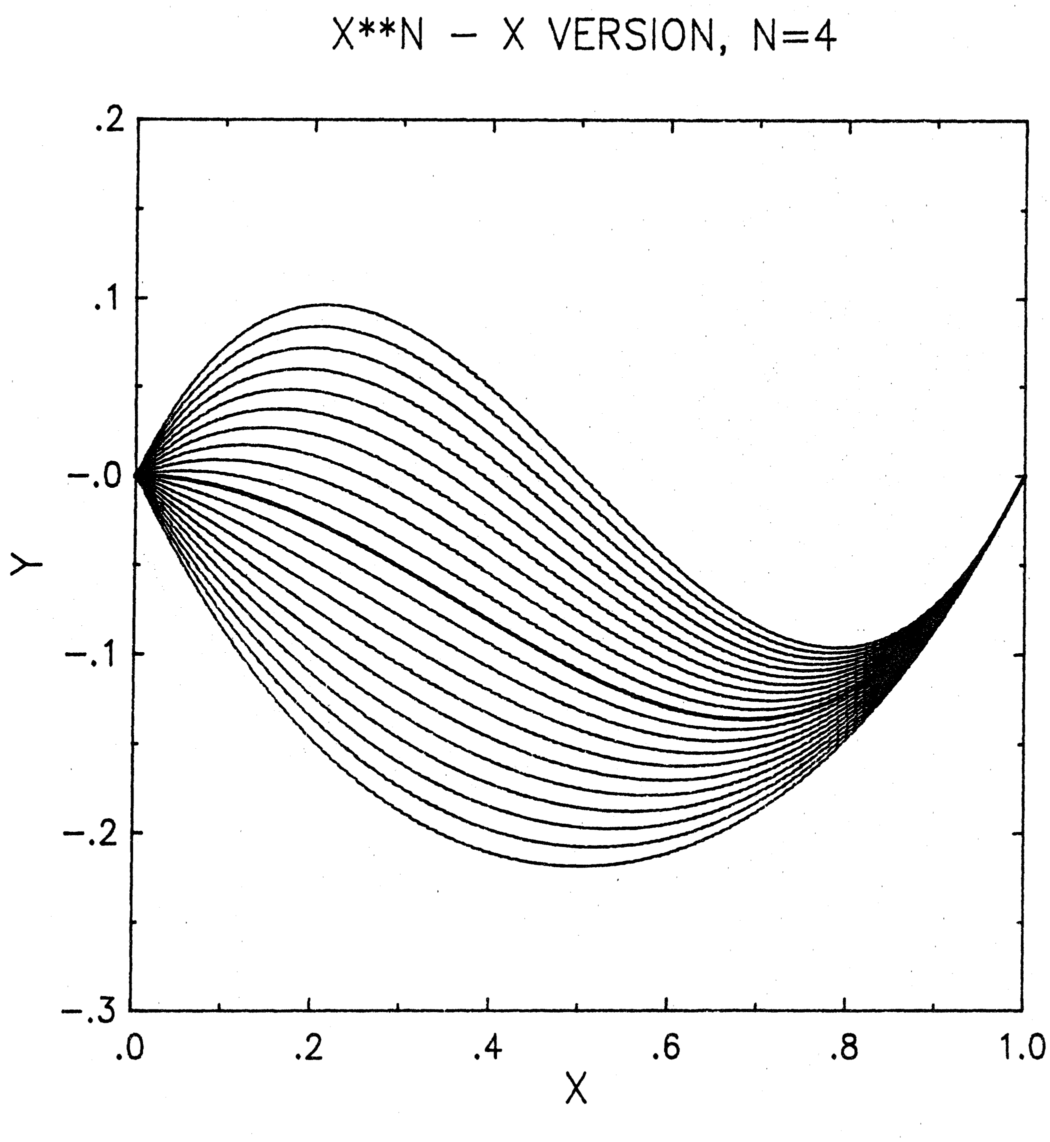


Figure B-2. Function based on an nth-degree polynomial with n = 4.

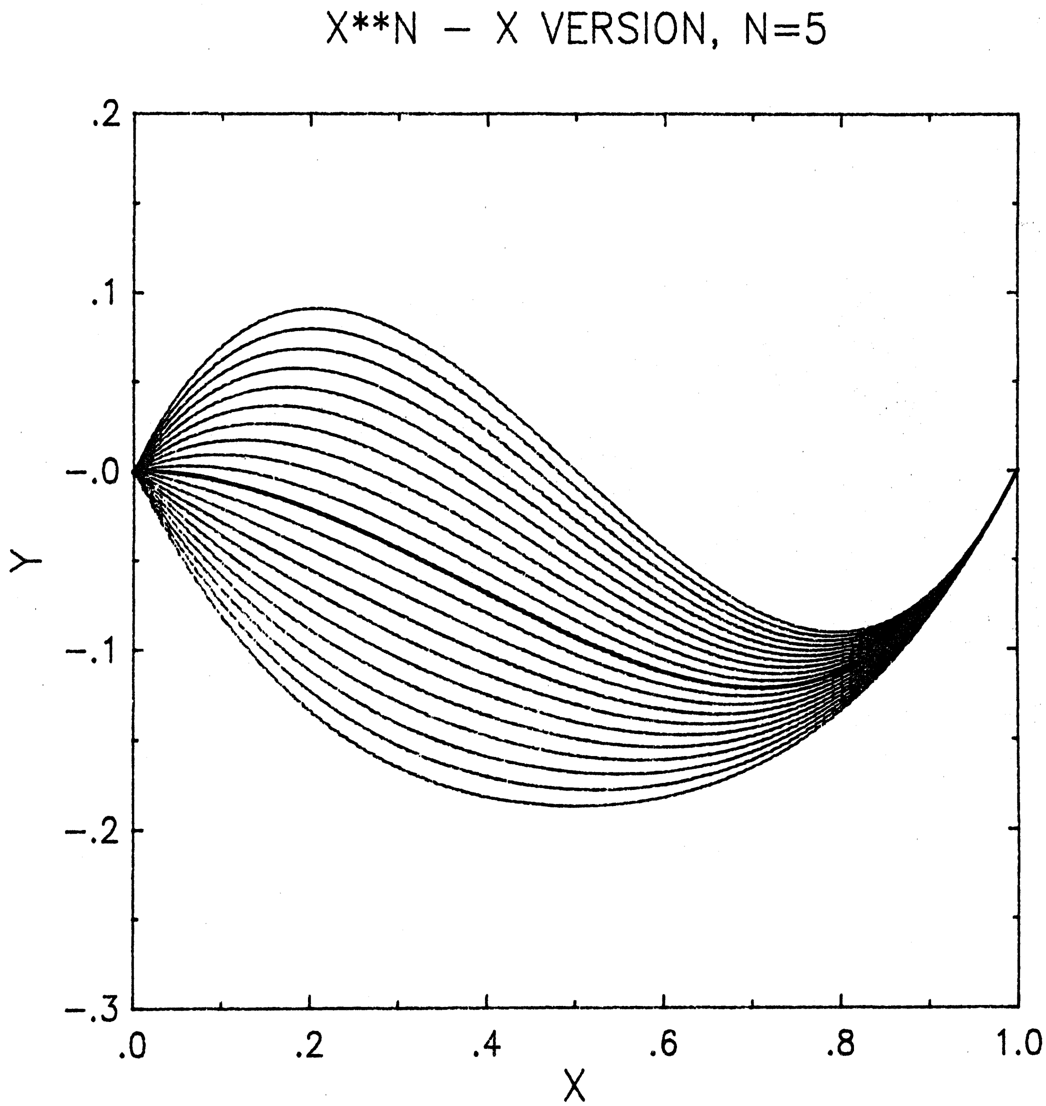


Figure B-3. Function based on an nth-degree polynomial with n = 5.

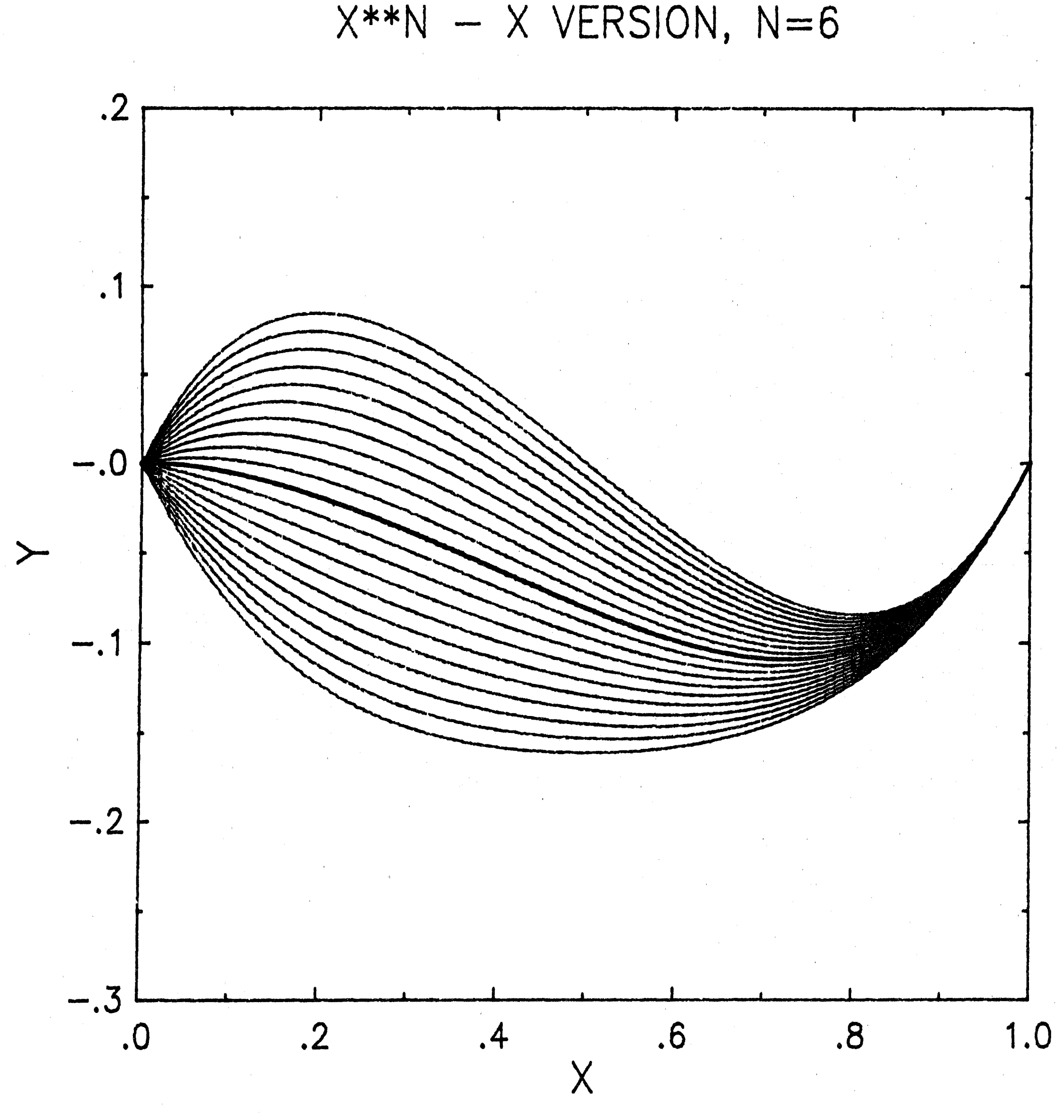


Figure B-4. Function based on an nth-degree polynomial with n = 6.

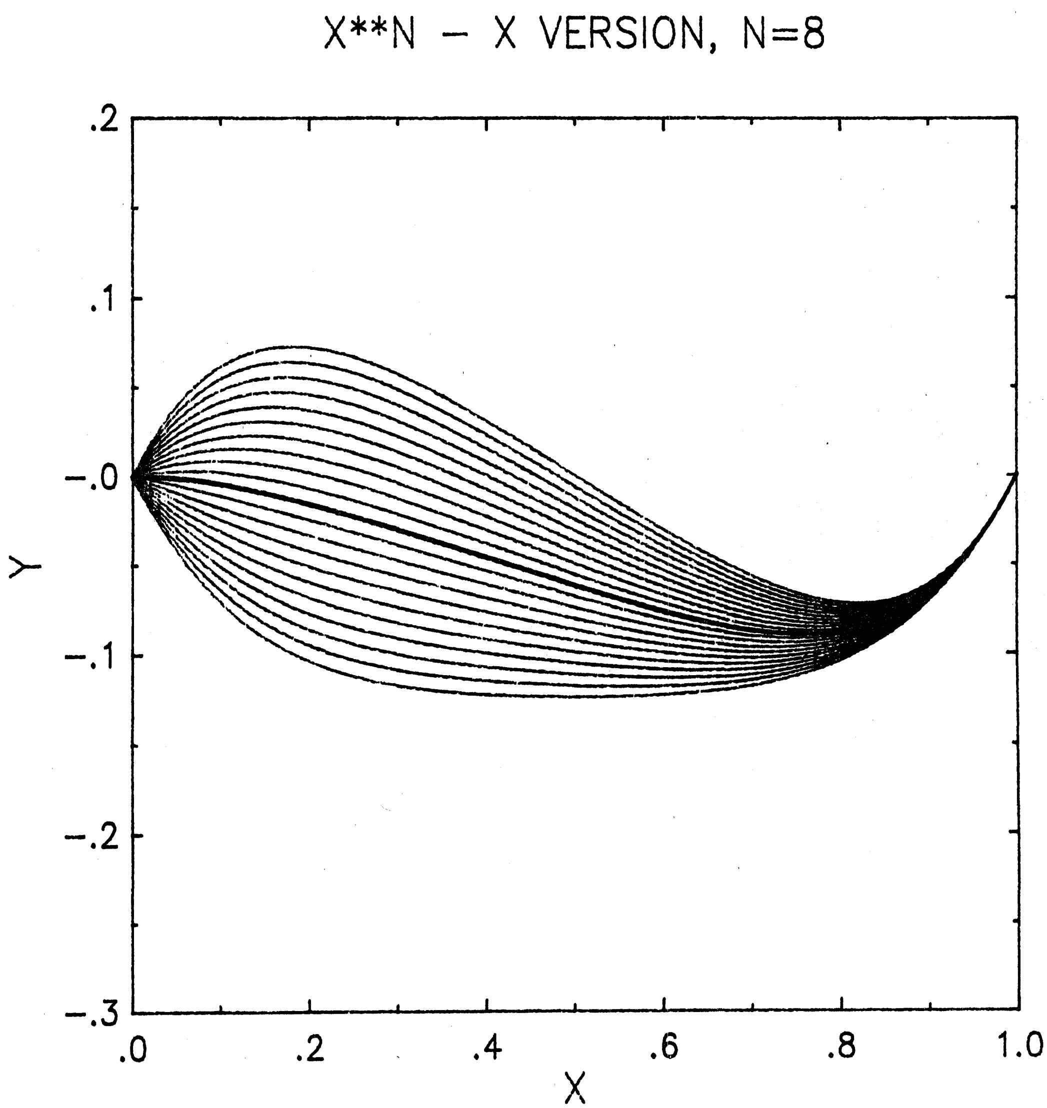


Figure B-5. Function based on an nth-degree polynomial with n = 8.

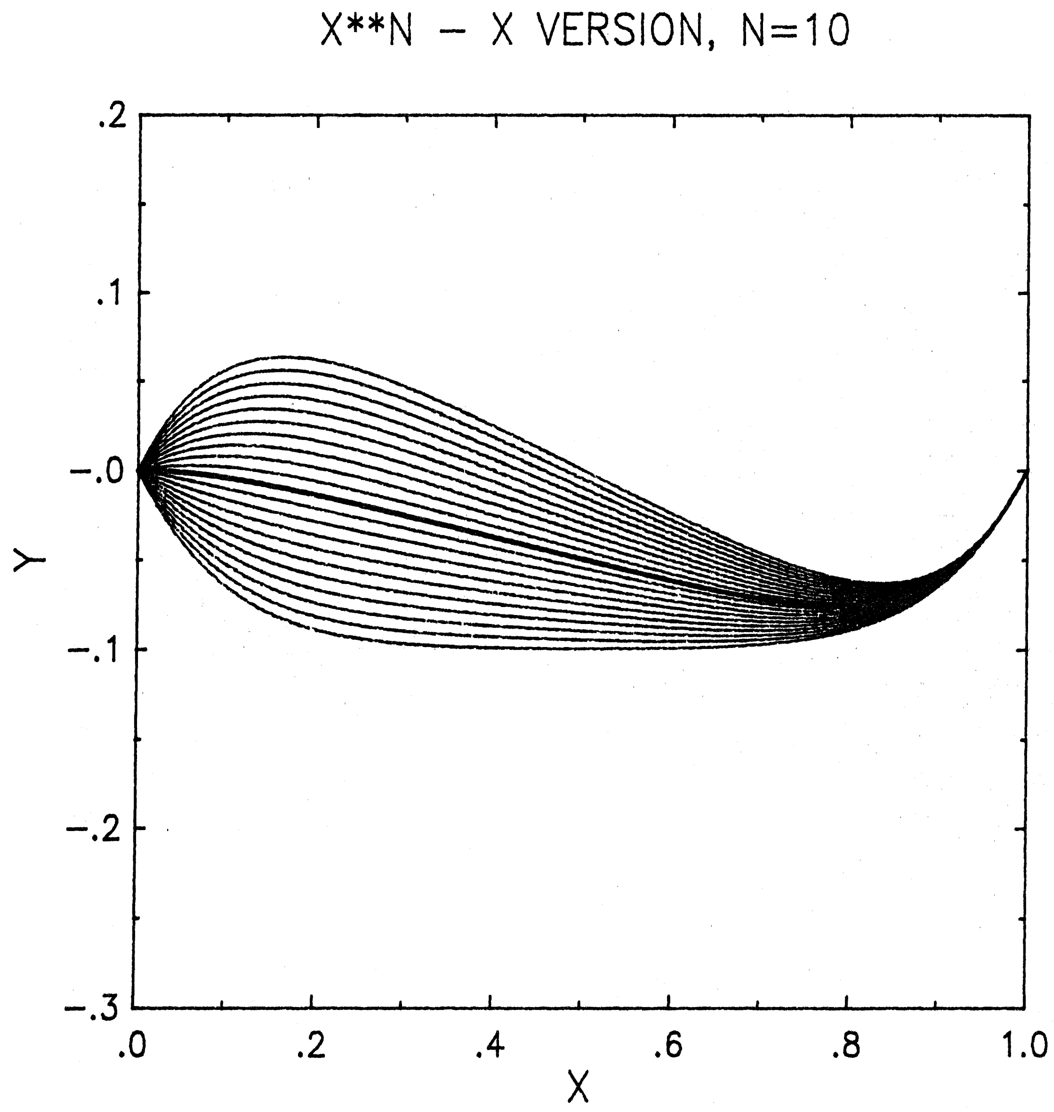


Figure B-6. Function based on an nth-degree polynomial with n = 10.

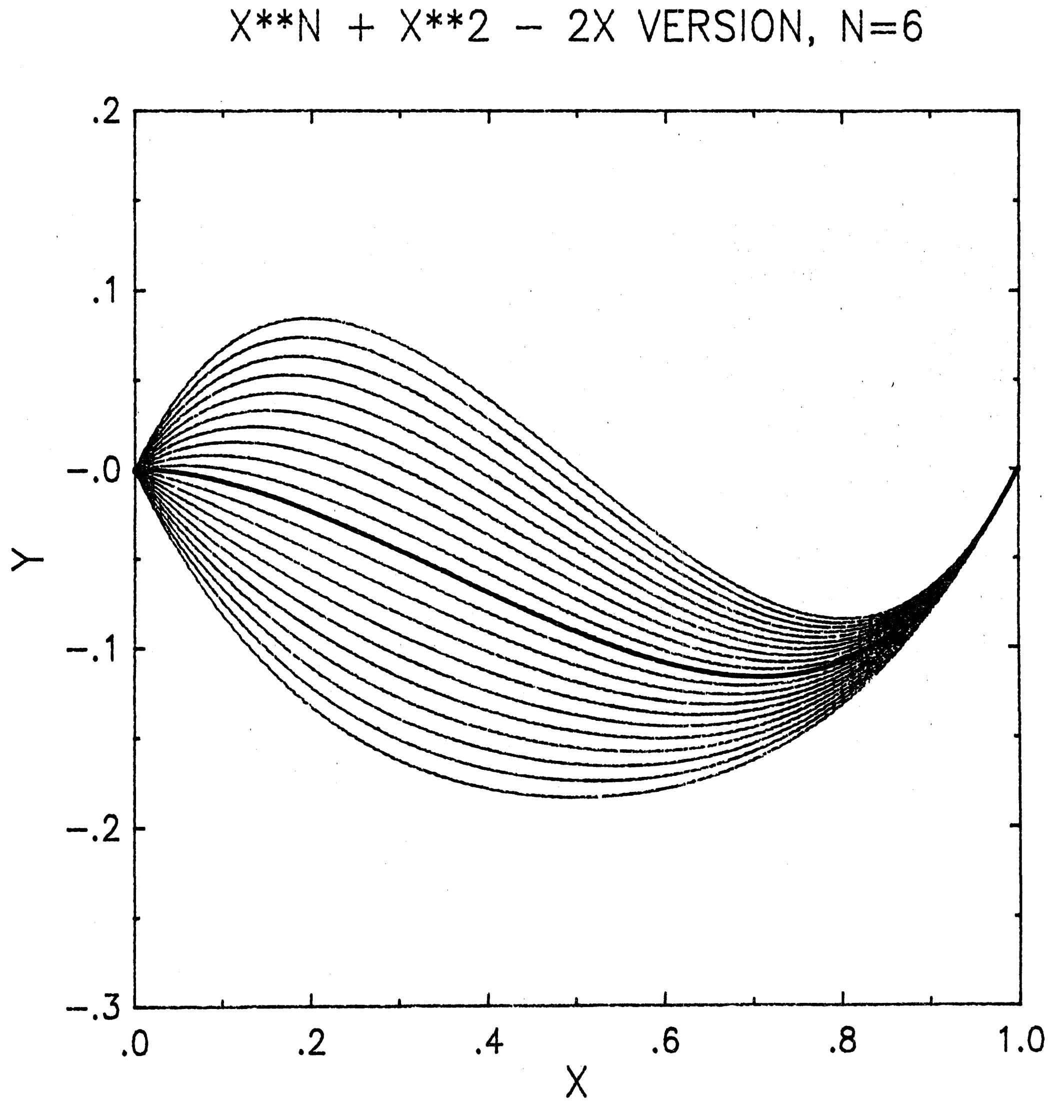


Figure B-7. Function based on an nth-degree and second-degree polynomials with n = 6.

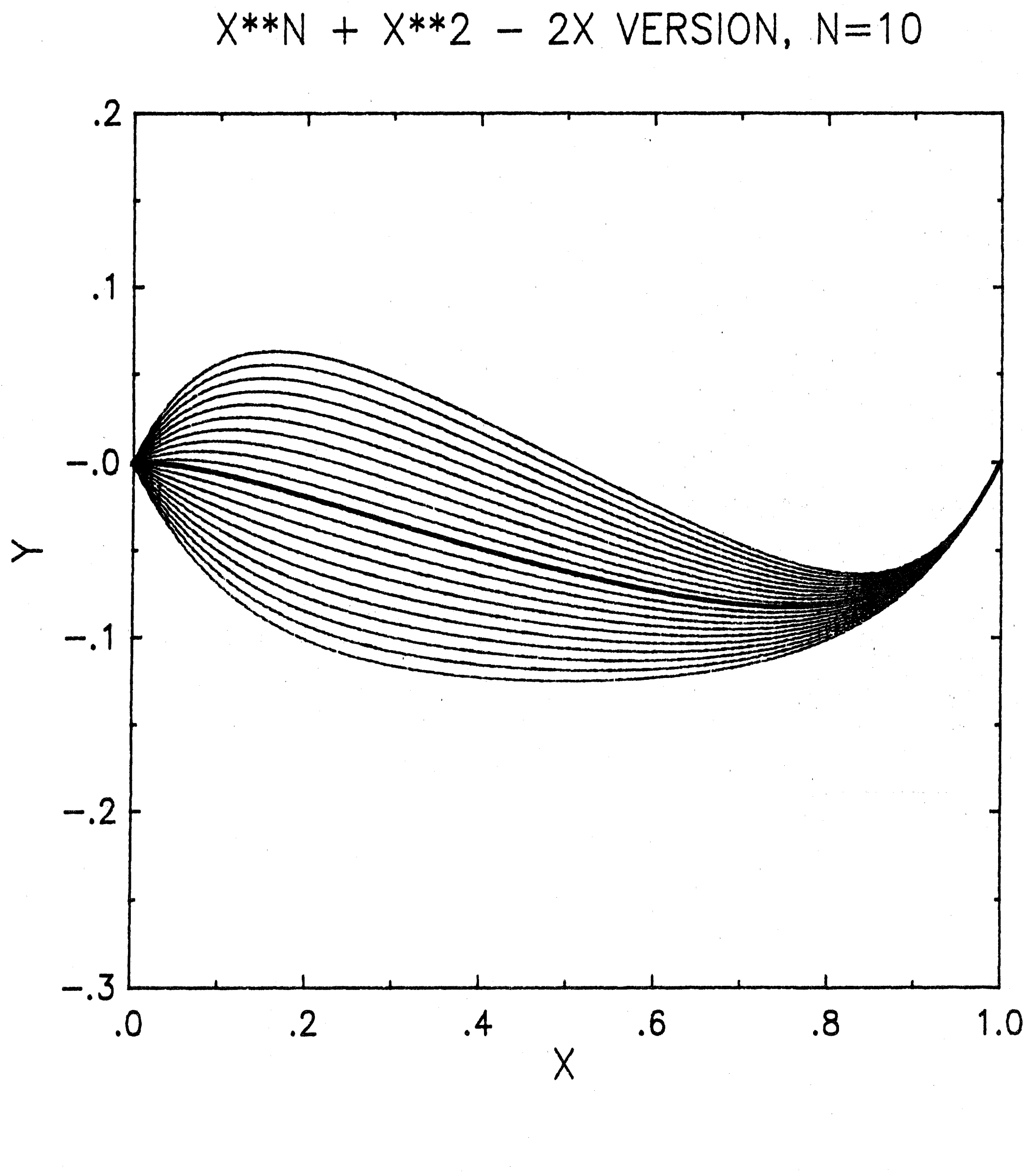


Figure B-8. Function based on an nth-degree and second-degree polynomials with n = 10.

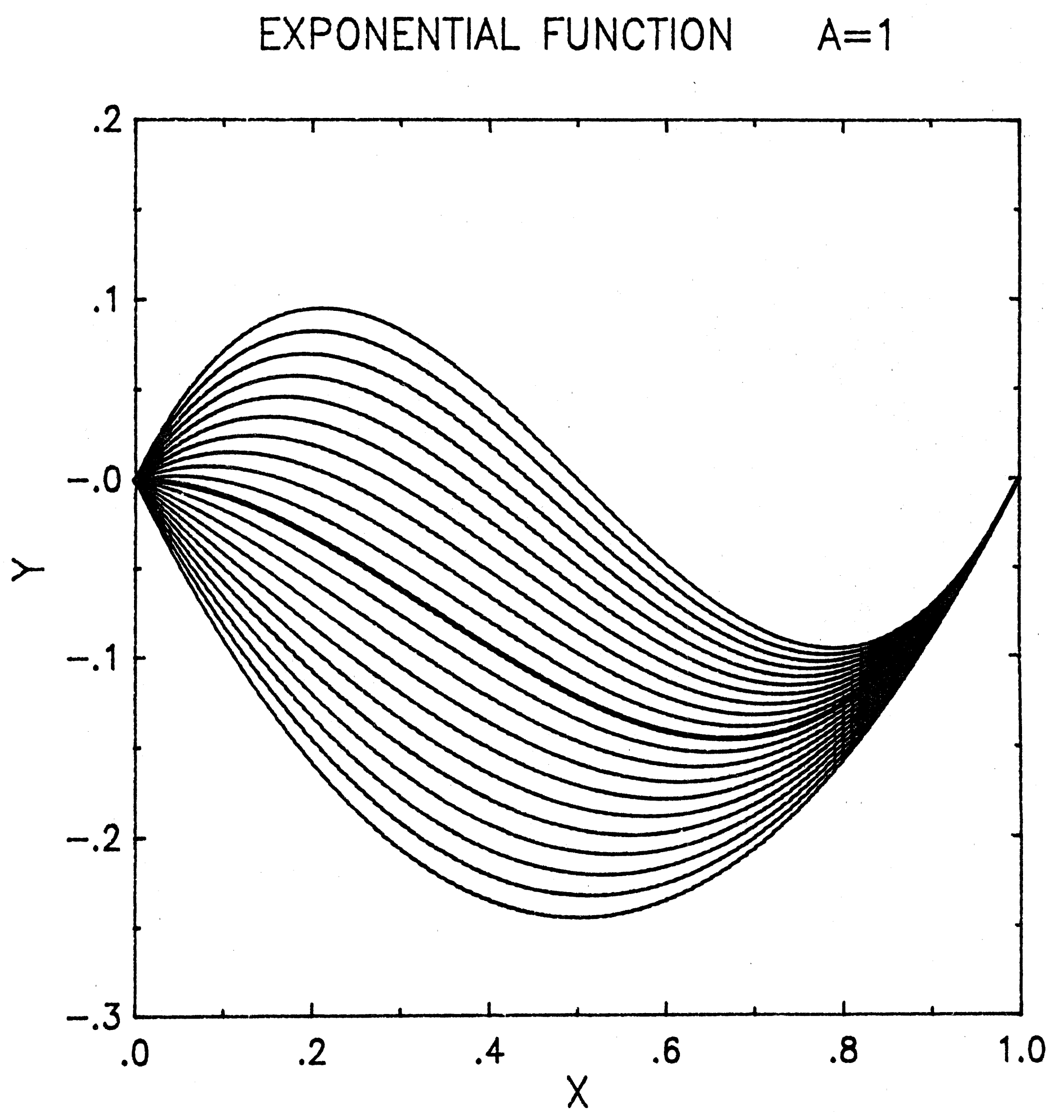


Figure B-9. Function based on an exponential function exp(ax) with a = 1.

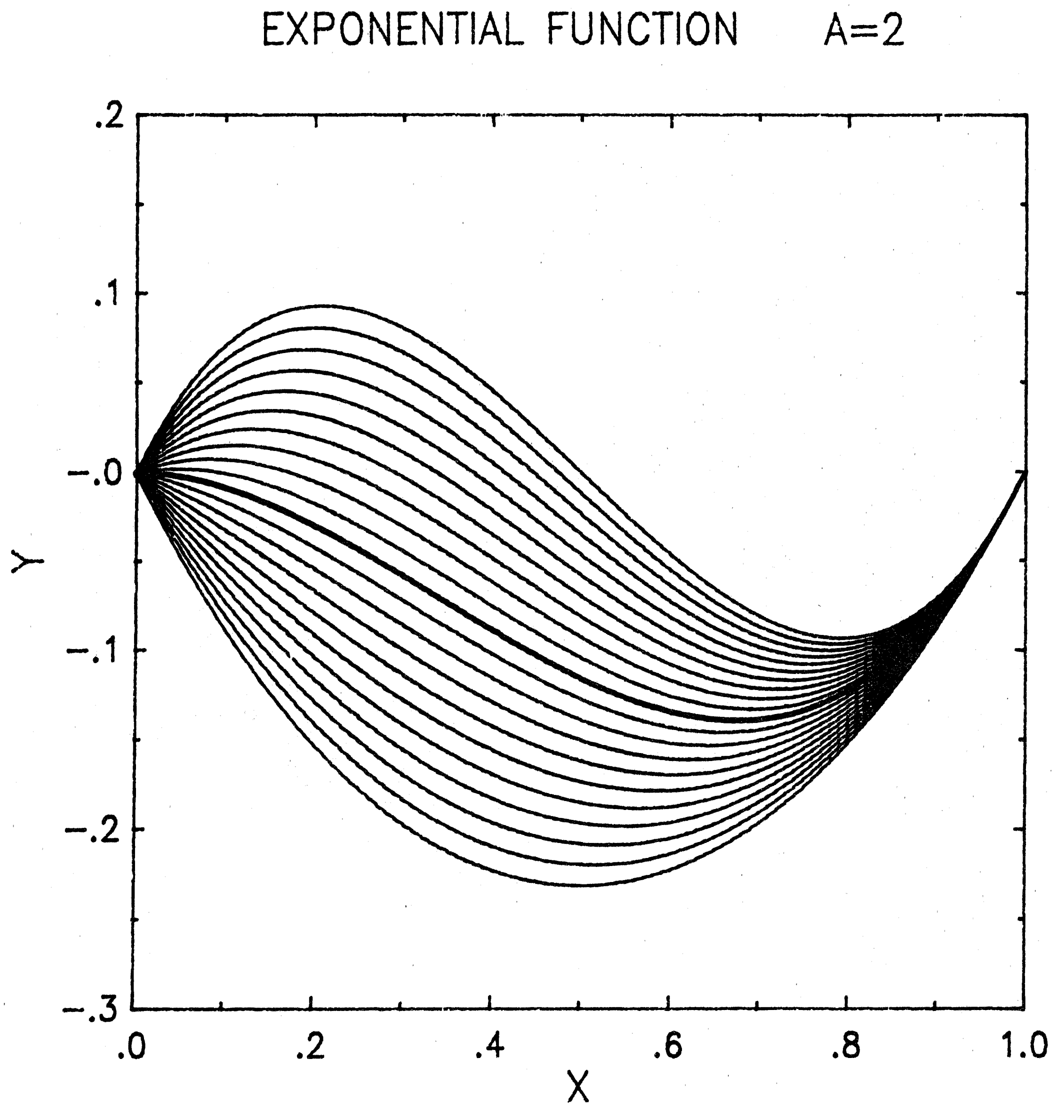


Figure B-10. Function based on an exponential function exp(ax) with a = 2.

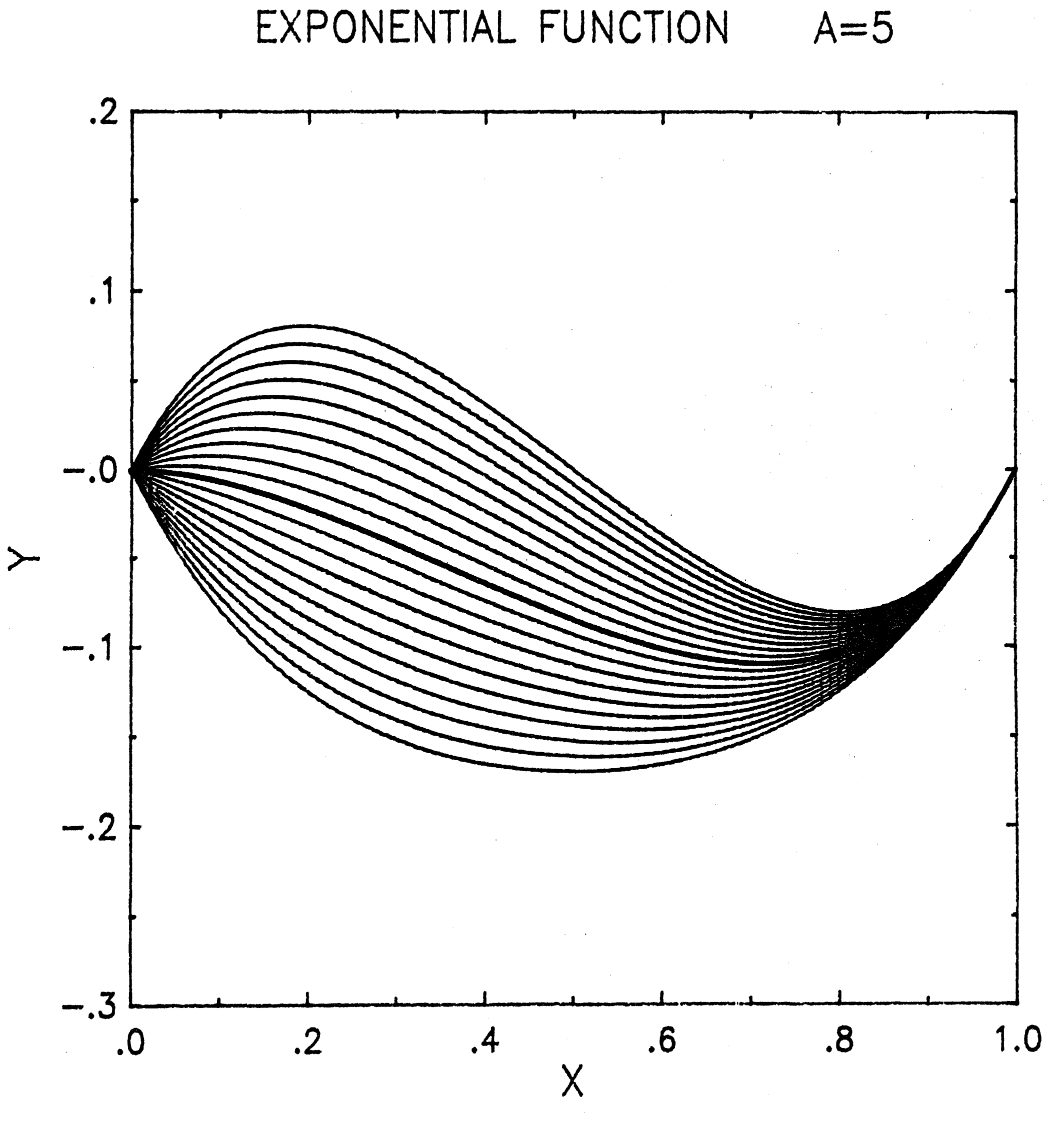


Figure B-11. Function based on an exponential function exp(ax) with a = 5.

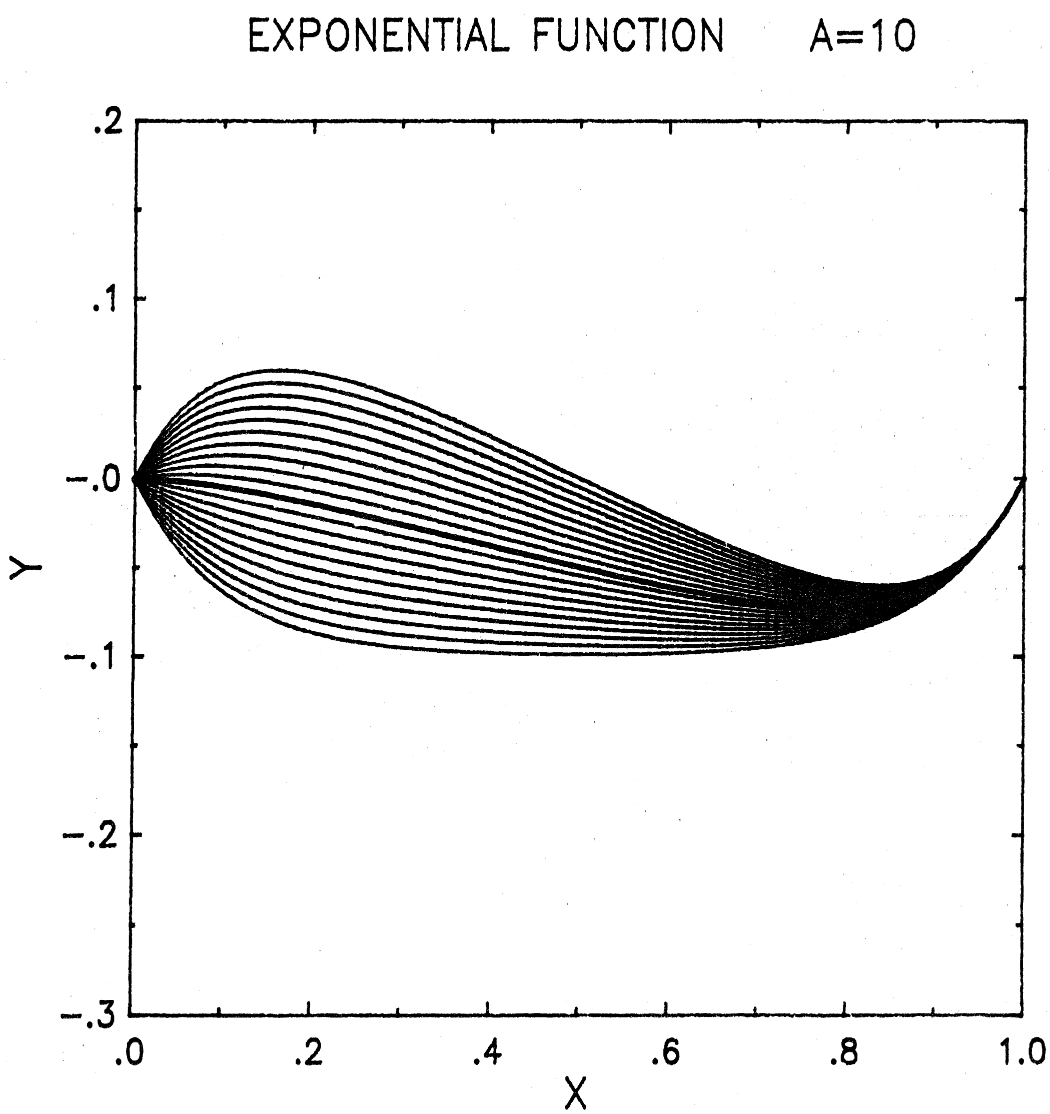


Figure B-12. Function based on an exponential function exp(ax) with a = 10.

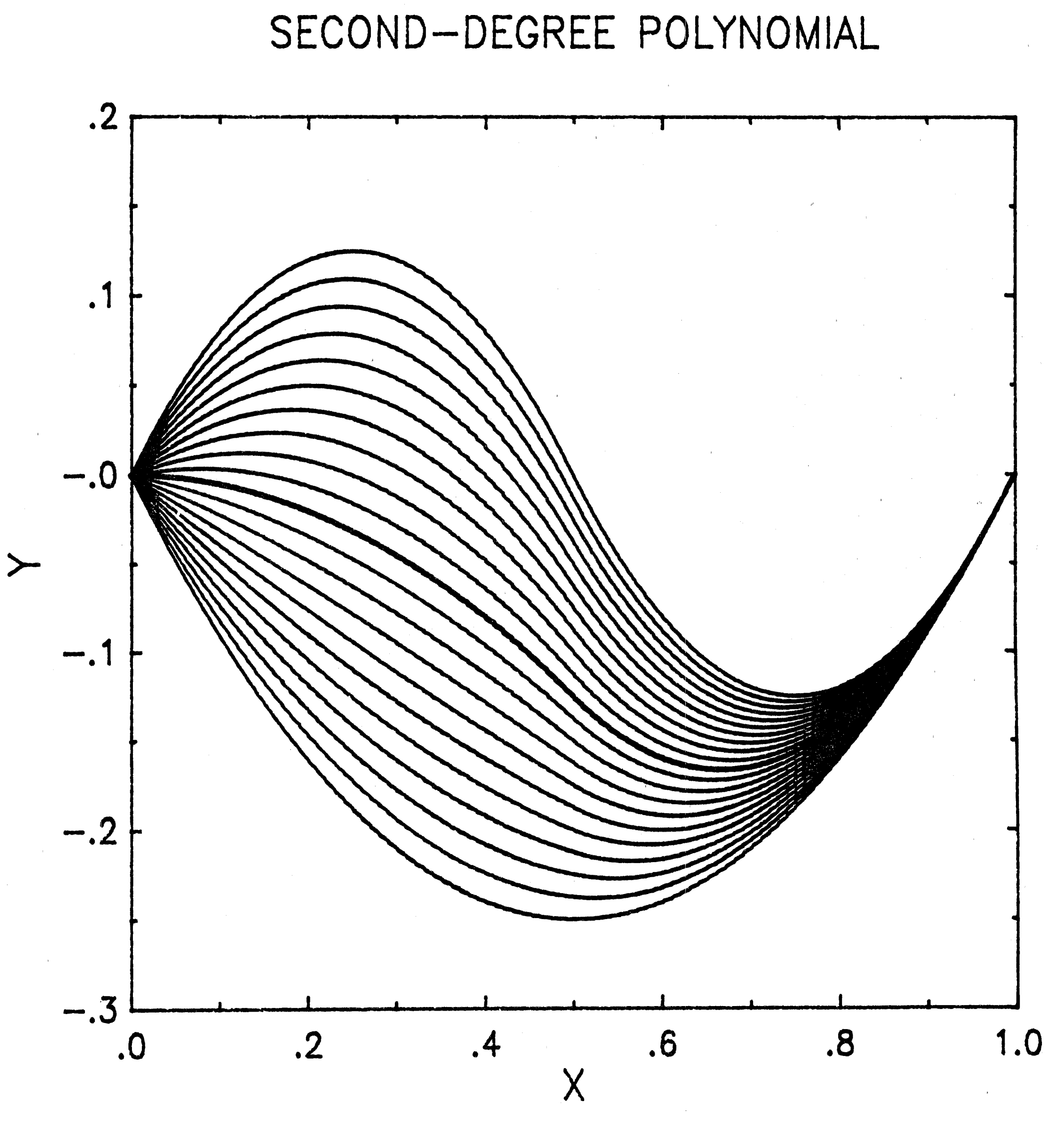


Figure B-13. Piecewise function composed of two second-degree polynomials.

FORM **NTIA-29** (4-80)

U.S. DEPARTMENT OF COMMERCE
NAT'L. TELECOMMUNICATIONS AND INFORMATION ADMINISTRATION

BIBLIOGRAPHIC DATA SHEET

	1. PUBLICATION NO.	2. Gov't Accession No.	3. Recipient's Accession No.
	NTIA Report 86-208		

4. TITLE AND SUBTITLE A Method of Univariate Interpolation That Has The Accuracy of A Third-Degree Polynomial	5. Publication Date November 1986 6. Performing Organization Code NTIA/ITS.N2
7. AUTHOR(S) Hiroshi Akima	9. Project/Task/Work Unit No.
8. PERFORMING ORGANIZATION NAME AND ADDRESS National Telecommunications & Information Admin. Institute for Telecommunication Sciences 325 Broadway Boulder, CO 80303	10. Contract/Grant No.
11. Sponsoring Organization Name and Address National Telecommunications & Information Admin. Herbert C. Hoover Building 14th & Constitution Avenue, N.W. Washington, D.C. 20230	12. Type of Report and Period Covered 13.

14. SUPPLEMENTARY NOTES

15. ABSTRACT *(A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here.)*

A method of interpolation that accurately interpolates data values that satisfy a function is said to have the accuracy of that function. The desired or required properties for a univariate interpolation method are reviewed, and the accuracy of a third-degree polynomial is found to be one of the desirable properties. A method of univariate interpolation having the accuracy of a third-degree polynomial while retaining the desirable properties of the method developed earlier by Akima (J. ACM 17, pp. 589-602, 1970) has been developed. The newly developed 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 data points. The method estimates the first derivative of the interpolating function (or the slope of the curve) at each given data point from the coordinates of seven data points. The resultant curve looks natural in many cases when the method is applied to curve fitting. The method is presented with examples. Possible use of a higher-degree polynomial in each interval is also examined.

16. Key Words *(Alphabetical order, separated by semicolons)*

curve fitting; interpolation; polynomial; second-degree polynomial; third-degree polynomial; univariate interpolation

17. AVAILABILITY STATEMENT	18. Security Class. *(This report)*	20. Number of pages
☒ UNLIMITED.	UNCLASSIFIED	72
☐ FOR OFFICIAL DISTRIBUTION.	19. Security Class. *(This page)* UNCLASSIFIED	21. Price:

☆ U.S. Government Printing Office: 1980—678-495/529