A Consistently Well-Behaved Method of Interpolation

Russell W. Stineman

We're not an academic journal, but once in a while something serious and original comes in. This is apparently a real solution to a problem graphics guys know well; this paper gives a smooth slope-matching interpolating curve, without ever going wild.

Abstract

When a curve is represented in a digital computer by a table of points, many methods of interpolation have difficulty near an abrupt change in the slope of the original curve. The problem is evidenced by the interpolating curve having more inflection points than the actual function that it is intended to approximate.

Polynomial interpolation often gives wild results near an abrupt change of slope.

This paper presents a method of interpolation which generates a curve that will never have more inflection points than are clearly required by the given set of points. The interpolating curve passes through the tabulated points and exactly matches the given slopes at those points (except for one unlikely degenerate case which has a slope discontinuity at one of the given points).

When used to approximate a sine function, the method of this paper was found to be more accurate than spline interpolation. The amount of computation required to find an interpolated point is approximately the same as to evaluate a sixth-degree polynomial.

An appendix presents a suggested way to compute slopes at the given points when only the points are known. However, exact slopes should be used whenever possible.

Russell W. Stineman, Boeing Aerospace Company, P.O. Box 3999, Seattle, WA 98124.

Introduction

In digital computer computations, it is common to represent a curve by a set of points, with interpolation between points. However, as is well known, polynomial interpolation often gives wild results near an abrupt change of slope. Spline interpolation can also give unreasonable results.

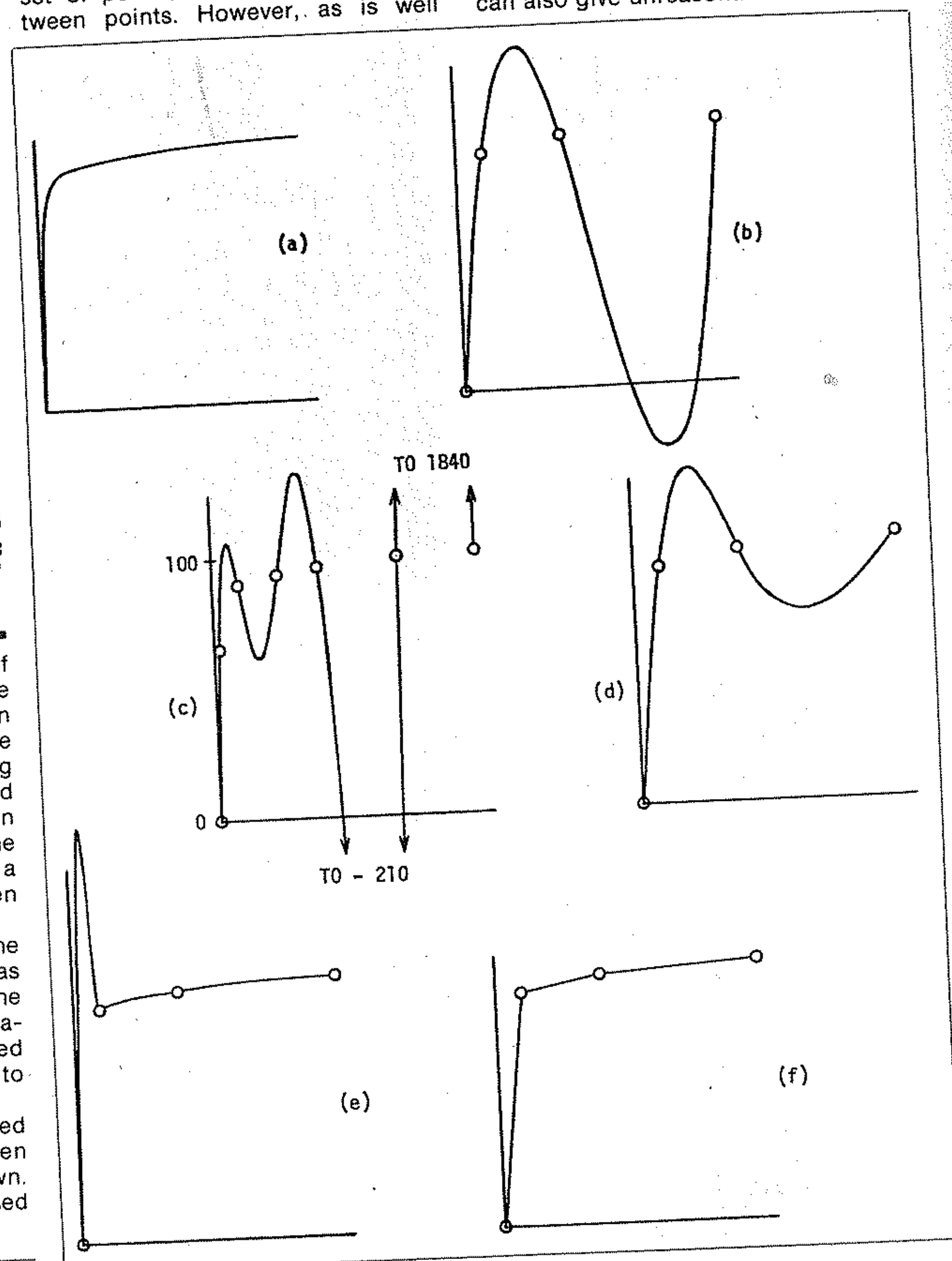

Figure 1

The problem is illustrated in Figure 1. Curve (a) shows a monotonically increasing curve with an abrupt decrease in slope. In (b), four points on curve (a) have been fitted by a cubic polynomial, with completely unsatisfactory results. In (c), three additional points have been taken from (a) and all seven points have been fitted by a 6th-degree polynomial, still with unsatisfactory results. In (d), the same four points as in (b) have been fitted by three piecewise cubic polynomials, chosen to preserve continuity of first and second derivatives at the interior points ("spline" interpolation), again unsatisfactory. Piecewise cubics are also used in (e), but the slopes at the given points are made equal to the slopes in (a), also unsatisfactory. In desperation, many analysts have used linear interpolation as in (f), accepting the need for a relatively large number of points to achieve a given accuracy.

The complete assurance that the procedure will never generate "wild" points makes it attractive as a general-purpose procedure.

What is needed is an interpolation procedure with the following properties:

a. If values of the ordinates of the specified points change monotonically, and the slopes of the line segments joining the points change monotonically, then the interpolating curve and its slope will change monotonically.

b. If the slopes of the line segments joining the specified points change monotonically, then the slope of the interpolating curve will change monotonically.

c. Suppose that the conditions in (a) or (b) are satisfied by a set of points, but a small change in the ordinate or slope at one of the points will result in conditions (a) or (b) being no longer satisfied. Then making this small change in the ordinate or slope at a point will cause no more than a small change in the interpolating curve.

An interpolation procedure that has the above properties is given in the next section of this article. The last of the three properties is discussed later in the paper in terms of an example.

Interpolation Procedure

In the following discussion, it is assumed that $x_j, y_j, y_j^!$, $j = 1, 2, \ldots, n$ are given where

x_j, y_j = rectangular coordinates of jth point on curve.

$y_j^!$ = slope of the curve at jth point.

$x_j < x_{j+1}$ for $j = 1, 2, \ldots, n-1$

If the slopes are not initially known, they may be calculated by the method described in the Appendix. Slopes thus calculated are consistent with achieving the objectives stated in the Introduction.

Given x such that $x_j \le x \le x_{j+1}$, the procedure for calculating y (the corresponding interpolated value) is the following. The slope of the line segment joining the two points is

$$s_j = \frac{y_{j+1} - y_j}{x_{j+1} - x_j} \qquad (1)$$

Values of s_j may be precomputed and stored along with the given points and slopes. On the line segment, the ordinate corresponding to x is

$$y_o = y_j + s_j (x - x_j) \qquad (2)$$

Next,

$$\Delta y_j = y_j + y_j^! (x - x_j) - y_o \qquad (3)$$

where Δy_j is the vertical distance from the point (x, y_o) to a line through (x_j, y_j) with slope $y_j^!$, as shown in Figure 2. Similarly,

$$\Delta y_{j+1} = y_{j+1} + y_{j+1}^! (x - x_{j+1}) - y_o \qquad (4)$$

is the vertical distance from the point (x, y_o) to a line thru (x_{j+1}, y_{j+1}) with slope $y_{j+1}^!$, also shown in Figure 2. The product $\Delta y_j \, \Delta y_{j+1}$ is then calculated and tested.

If $y_j^! = s_j$, then the line through point (x_j, y_j) with slope $y_j^!$ will coincide with the line segment joining points (x_j, y_j) and (x_{j+1}, y_{j+1}), and $\Delta y_j = 0$. Similarly, if $y_{j+1}^! = s_j$, then $\Delta y_{j+1} = 0$. If either or both Δy_j and Δy_{j+1} are zero, then the product $\Delta y_j \, \Delta y_{j+1} = 0$, and y is simply

$$y = y_o \qquad (5)$$

If $\Delta y_j \, \Delta y_{j+1} = 0$, but Δy_j and Δy_{j+1} are not both zero, then the interpolating curve will have a slope discontinuity. For example, if $\Delta y_j \neq 0$, then the slope for $x_j < x < x_{j+1}$ will be $y' = s_j = y_{j+1}^!$. But as $x \to x_j$ from the left, $y' \to y_j^! \neq s_j$. This degenerate case is the only way a slope discontinuity can occur.

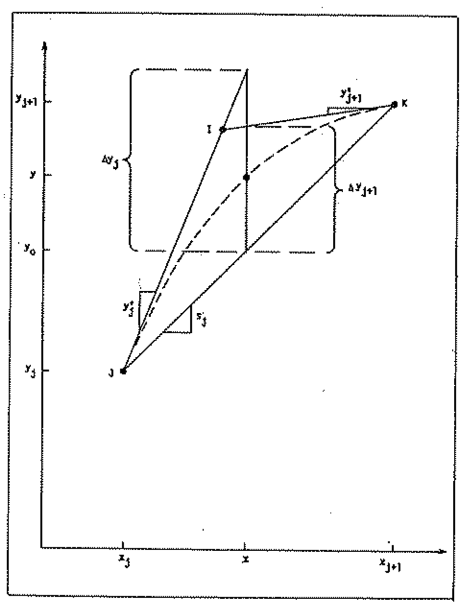

Figure 2

If $\Delta y_j \, \Delta y_{j+1} > 0$, then (as in Figure 2) Δy_j and Δy_{j+1} have the same sign, and

$$y = y_o + \frac{\Delta y_j \, \Delta y_{j+1}}{\Delta y_j + \Delta y_{j+1}} \qquad (6)$$

Equation (6) always determines the point (x,y) inside the triangle IJK of Figure 2. The slope of the interpolating curve matches the given slopes at the given points. The slope changes monotonically between the given points, so the interpolating curve is always concave toward the line segment joining the given points.

If $\Delta y_j \, \Delta y_{j+1} < 0$, then the geometry is like Figure 3, and there must be an inflection point between x_j and x_{j+1}. In this case,

$$y = y_o + \frac{\Delta y_j \, \Delta y_{j+1} \, (x - x_j + x - x_{j+1})}{(\Delta y_j - \Delta y_{j+1}) \, (x_{j+1} - x_j)} \qquad (7)$$

Equation (7) always determines the point (x,y) inside the quadrilateral JIKL of Figure 3, where the vertical distance LO equals the vertical distance OI. The slope of the interpolating curve matches the given slopes at the given points. The interpolating curve intersects line segment JK at its midpoint.

The rationale for equation (7) may be understood by considering the case where $y_j^!$ is significantly greater than s_j, the slope of line segment JK, but $y_{j+1}^!$ is nearly equal to s_j, Figure 2 or 3. Regardless of whether $y_{j+1}^!$ is greater or less than s_j, points I and L will be very close to point J and the interpolating curve will be very close to line segment JK. Thus, a change of $y_{j+1}^!$ from slightly more than s_j to slightly less than s_j will cause only a slight change in the interpolating curve. This example illustrates the third requirement given in the Introduction.

Equations (6) and (7) fall in the general area of rational interpolation. However, the desirable properties of this method of interpolation stem from the particular form of (6) and (7). In general, rational interpolation does not have such properties.

The curve in Figure 1(a) was calculated by the above method of interpolation, given the four points shown in Figure 1(b), and with slopes calculated by the method given in the Appendix.

Accuracy

The accuracy of the given interpolation procedure may be illustrated

Interpolation, cont'd...

by fitting the function

$$y = \sin x \qquad (8)$$

No attempt is made to get an optimum fit. Rather, the values chosen for x_j are 0, 45, and 90 degrees, and the corresponding values of y_j and y'_j are computed exactly using equation (8). The resulting interpolated curve deviates from sin x by a maximum of .00333, at x = 24 degrees.

By contrast, the maximum error using linear interpolation is .0704, at 68 degrees. From another standpoint, to achieve a maximum error of no more than .00333, linear interpolation requires that x be given at intervals of 9 degrees.

An example given in reference [1] considers one full cycle of the function given in equation (8). The points x_j are selected at 45-degree intervals (that is, x_j = 0, 45, 90,..., 360 degrees), and the values of y_j are calculated exactly by (8). Interpolation is then done with a fifth-degree spline. That is, piecewise fifth-degree polynomials are found such that the interpolating curve and its first four derivatives are continuous. The interpolating curve also exactly matches the first and third derivatives of equation (8) at x = 0 and at x = 360 degrees. The interpolating curve deviates from sin x by a maximum of .0372, at 25 and at 335 degrees.

The same 9 points as in the above example from reference (1) were fitted by the method of this paper, using slopes calculated by the method given in the Appendix, rather than exact slopes. In this case, the interpolating curve deviates from sin x by a maximum of .0766, at 18 and at 342 degrees. Using exact slopes, the maximum error is .00333, at 24, 156, 204, and336 degrees. This shows the importance of using accurate slopes at the given points, if the slopes are known.

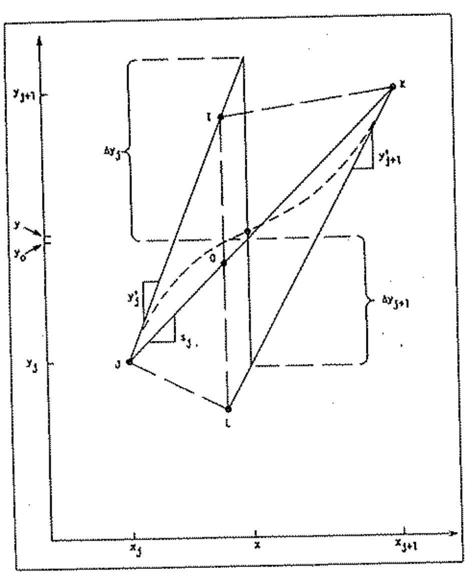

Figure 3

Conclusions

A procedure has been presented for interpolating between tabulated points. This procedure completely avoids the problems which various forms of polynomial interpolation, including spline interpolation, have near an abrupt change of slope. The procedure is especially recommended for such applications as the voltage-current curve of a semiconductor. However, the complete assurance that the procedure will never generate "wild" points makes it attractive as a general-purpose procedure.

The procedure uses only ordinary arithmetic operations (that is, no trigonometric, exponential, or similar functions need be evaluated. The number of operations is approximately equivalent to the evaluation of a sixth-degree polynomial. Memory required for data is four words per point.

In desperation, many analysts have used linear interpolation, accepting the need for a relatively large number of points to achieve a given accuracy.

When fitting a known function, the procedure given in this paper is at least as accurate as spline interpolation, provided that accurate values are available for the slopes of the function at the given points. For a given accuracy, the spacing of tabulated points may be significantly greater than for linear interpolation.

Appendix. Calculation of Slopes

Given the points x_j, y_j, j = 1, 2,..., n, the problem addressed in this Appendix is to compute slopes, y'_j, consistent with the requirements stated in the Introduction. It is assumed that the interpolating procedure given in this paper will be used.

In Figure 4, let I, J, and K be any three consecutive points. Point J may be above or below the line segment joining I and K, as shown in Figures 4(a) and 4(b), respectively. The requirements of the Introduction are satisfied if y'_j has a value between the slopes of the line segments IJ and JK. That is, for Figure 4(a), it is necessary that

$$\text{slope (IJ)} > y'_j > \text{slope (JK)} \qquad (9a)$$

while for Figure 4(b),

$$\text{slope (IJ)} < y'_j < \text{slope (JK)} \qquad (9b)$$

Another point is that if, for example, line segment IJ is much shorter than JK, it may easily be seen that a smoother overall interpolating curve will result if y'_j is nearly equal to the slope of IJ.

All of the above considerations are satisfied by setting y_j equal to the slope at point J of a circle through points I, J, and K. Thus

$$y'_j = \frac{(y_j-y_i)\,((x_k-x_j)^2 + (y_k-y_j)^2) + (y_k-y_j)\,((x_j-x_i)^2 + (y_j-y_i)^2)}{(x_j-x_i)\,((x_k-x_j)^2 + (y_k-y_j)^2) + (x_k-x_j)\,((x_j-x_i)^2 + (y_j-y_i)^2)} \qquad (10)$$

Equation (10) takes care of all interior points. The end points require special attention. From the requirements of the Introduction, it may be seen that the slope at an end point must have the same sign as the line segment from the end point to the next point. In line with the use of equation (10), one might try drawing a circle through the first (or last) three points. However, if either point I or K of Figure 4 is an end point, the particular arrangement of the points causes the slope of the circle at I or K to have the wrong sign. The sign of the slope is not necessarily wrong at the end point, but it may be wrong.

The problem divides into two cases. To simplify the notation, let M and subscript m designate either point I or point K, whichever is an end point, and let s be the slope of the line segment joining points J and M. The first case occurs when s is "steeper" than y'_j. In this case, a parabola through

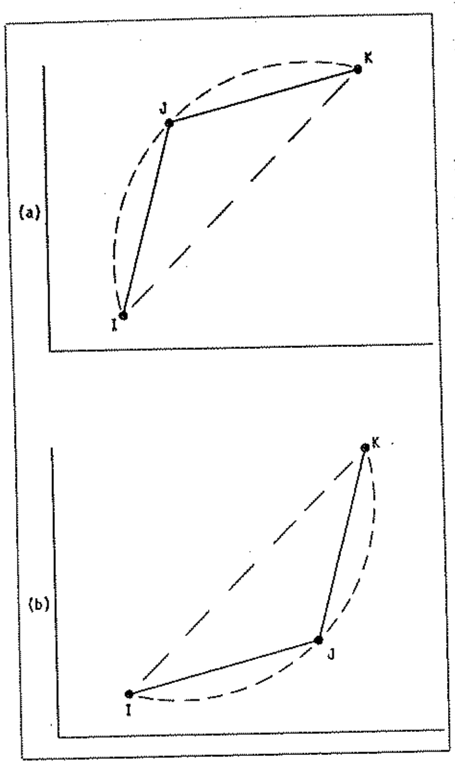

Figure 4

J and M with slope y'_j at J has a slope at M which meets the requirements and is reasonable. Stated more precisely, if $s > 0$ and $s > y'_j$, or if $s < 0$ and $s < y'_j$, then

$$y'_m = s + (s - y'_j) = 2s - y'_j \quad (11a)$$

The second case occurs if neither condition for the first case is satisfied.

The procedure uses only ordinary arithmetic operations. The number of operations is approximately equivalent to the evaluation of a sixth-degree polynomial. Memory required for data is four words per point.

In this case, the term in parenthesis in equation (11a) is multiplied by a factor between zero and one which assures that y'_m is always the same sign as s. The result is

$$y'_m = s + \frac{|s|\,(s-y'_j)}{|s| + |s-y'_j|} \quad (11b)$$

It should be understood that the slopes calculated by equations (10) and (11) are not independent of the scaling of the variables. For best results, x and y should be scaled to have roughly equal ranges, before calculating slopes. □

References

A. C. Ahlin, *On Smooth Interpolation by Continuously Connected piecewise Polynomials*, Rendiconti del Circolo Matematico di Palermo, Serie II, Tomo XX, 1971, pp 229-253.

"This system is not very friendly."