UNCLASSIFIED

Y-1400
Mathematics and Computers

Y 3. At7
22/Y-1400

AEC
RESEARCH REPORTS

AEC RESEARCH AND DEVELOPMENT REPORT

CUBIC SPLINE, A CURVE FITTING ROUTINE

A. H. Fowler
C. W. Wilson

UNION CARBIDE NUCLEAR COMPANY
DIVISION OF UNION CARBIDE CORPORATION

Operating

- OAK RIDGE GASEOUS DIFFUSION PLANT
- OAK RIDGE Y-12 PLANT
- OAK RIDGE NATIONAL LABORATORY
- PADUCAH GASEOUS DIFFUSION PLANT

for the Atomic Energy Commission

Under U. S. Government Contract W7405 eng 26

metadc173349

UNCLASSIFIED

Printed in USA. Price: $2.00 Available from the
Office of Technical Services
U. S. Department of Commerce
Washington 25, D. C.

LEGAL NOTICE

This report was prepared as an account of Government sponsored work. Neither the United States, nor the Commission, nor any person acting on behalf of the Commission:

A. Makes any warranty or representation, expressed or implied, with respect to the accuracy, completeness, or usefulness of the information contained in this report, or that the use of any information, apparatus, method, or process disclosed in this report may not infringe privately owned rights; or

B. Assumes any liabilities with respect to the use of, or for damages resulting from the use of any information, apparatus, method, or process disclosed in this report.

As used in the above, "person acting on behalf of the Commission" includes any employee or contractor of the Commission, or employee of such contractor, to the extent that such employee or contractor of the Commission, or employee of such contractor prepares, disseminates, or provides access to, any information pursuant to his employment or contract with the Commission, or his employment with such contractor.

Date Issued: April 12, 1963

Report Number Y-1400

Mathematics and Computers
TID-4500 (19th Edition)

UNION CARBIDE NUCLEAR COMPANY
Division of Union Carbide Corporation

Y-12 PLANT

Contract W-7405-eng-26
With the U. S. Atomic Energy Commission

CUBIC SPLINE, A CURVE FITTING ROUTINE

A. H. Fowler
C. W. Wilson

Oak Ridge, Tennessee
September 5, 1962

Report Number Y-1400

Mathematics and Computers
TID-4500 (19th Edition)

Distribution:

Bailey, E. W.	
Ballenger, H. F.	(2)
Barker, B. E.	(San-Liv)
Beisheim, Harvey	(Dow-RF)
Bell, B. B.	
Bernander, N. K.	
Burkhart, L. E.	
Carlson, Robert	(McDonnell Aircraft Corp)
Carpenter, S. D.	(Mason and Hanger-Amarillo)
Cartledge, C. M.	(LASL)
Cowen, D. D.	(ORNL)
Dean, R. Y.	(GE-Hanford)
Emerson, E. L.	(San-Alb)
Erickson, Robert	(Mason and Hanger-Burlington)
Evans, G. W.	
Fair, G. M.	(Boeing/Aero Space)
Fortenbery, M. J.	
Fowler, A. H.	
Gabbard, J. L., Jr.	(ORGDP)
Gritzner, V. B.	
Harwell, W. L.	(ORGDP) (5)
Hemphill, L. F.	
Hensley, C. E.	
Hori, S.	(Armour Research Foundation)
Huber, A. P.	(ORGDP)
Johnson, W. B.	(North American Aviation)
Keller, C. A.	(AEC-ORO) (4)
Kite, H. T.	
Lewis, F. O.	(ORGDP)
Little, J. C.	

In addition, this report is distributed in accordance with the category, Mathematics and Computers as given in the "USAEC Standard Distribution Lists for Unclassified Scientific and Technical Reports", TID-4500 (19th Edition), February 1, 1963.

Lockhart, G. B.	
Malmquist, K. O., Jr.	(LRL)
Maytag, J. H.	(Dow-RF)
McLendon, J. D.	
Mills, B. G.	(Dow-RF)
Miskell, R. V.	
Mitchel, G. W.	
Murray, J. P.	(ORGDP)
Oliver, D. A.	(ACFI-Alb)
Patton, F. S.	
Pritchard, W. S.	(ORGDP)
Rader, D. H.	
Raminez, Jose	(Mason and Hanger-Amarillo)
Schwenn, M. F.	
Sestric, M. D., Jr.	(Bendix-KC)
Signor, R. W.	(GE-Huntsville)
Spies, R. J.	(LRL)
Stoner, H. H.	
Tilson, F. V.	
Uffelman, F. C.	
Visnaw, V. V.	(Bendix-KC)
Waters, J. L.	
Whitson, W. K.	
Wilkinson, P. E.	
Wilson, C. W.	(25)
Winkel, R. A.	(Paducah)
Zurcher, E.	
Zwayer, B. E.	(GE-Cin)
Y-12 Central Files	(5)
Y-12 Central Files	(Y-12RC)

ABSTRACT

A method of mathematically fitting a curve through a given ordered set of points has been developed and programmed in fortran computer language. The fitted curve, which is made up of a series of normalized cubic polynomials, very nearly approximates the curve generated by passing an infinitely thin spline through the sets of points and is, therefore, called a "cubic spline". The curve has the properties of continuous position, slope, and curvature. The restraints placed upon the cubic spline may be relaxed by applying an additional feature which permits the points to be adjusted by, at most, some preset value. This has the effect of smoothing the curve in the sense that it reduces the strain energy stored in it while preserving the general shape of the curve. Tool centers for numerically controlled tools may be found with the interpolation and offsetting methods included.

CONTENTS

SUMMARY

A routine has been developed for mathematically fitting a curve through a given ordered set of points. The fitted curve very nearly approximates the curve generated by passing an infinitely thin spline through the set of points and is given the name "cubic spline". The significant details concerning this program are as follows:

1. The procedure fits a smooth curve to a given set of points using normalized cubic polynomials; one cubic for each interval. Each cubic passes through the points at the beginning and end of its interval.

2. The cubic spline has the properties of continuous position, slope, and curvature. Continuous curvature is obtained through the use of an iterative scheme. An example of the results is shown in Appendix B, Example 1.

3. After a cubic spline has been passed through the given points, an optional smoothing procedure may be applied. This feature of the program adjusts the points slightly so as to reduce the stored strain energy thereby "smoothing" the curve. An example of the results is shown in Appendix B, Example 2.

4. The cubic spline technique as presented in this report is programmed in fortran language for use on an IBM 7090 computer. In addition to fitting the curve, this program provides the user with much additional information which may be used in definitive or diagnostic manner. These include lists of point and curve parameters and a curvature plot which reveals at a glance any unsuspected irregularities such as bumps, inflections, or local roughness.

5. The cubic spline routine has been incorporated into the APT (automatically programmed tools) computer system.(1)

6. Interpolated points may be found and offset to produce tool centers for numerically controlled machines. A list of both the interpolated points and the offset points is produced. An example of the results is shown in Appendix B, Example 3.

Given a set of points that "define a curve", it is felt that this routine accurately displays the true shape described by the

points, since no presumptions about the form of the curve are made and since the curve always passes exactly through the points. This is in contrast to the normal procedure in which the form of the fitted equation is predetermined and the technique is one of choosing the values of the coefficients that go into the equation such that the curve will pass near the points.

INTRODUCTION

Fairing curves through given sets of points with thin flexible splines of metal or plastic has been used for many years by mechanical designers as a standard method of curve fitting. Such a curve assumes a position of minimum strain energy and is generally considered "smooth". Through the development of continuous-path numerically controlled machining in recent years, a demand has been created for a practical method to fit such curves mathematically. The "cubic spline" technique described herein approximates a true spline curve. It is presented in fortran language for an IBM 7090 computer; however, the system is adaptable to any digital computer having the required capacity.

The procedure for executing the cubic spline fit, along with the necessary equations, is contained in the body of the report. For those who desire to run the programs, a listing of the necessary fortran statements is included in Appendix A with a description of each routine and its storage requirements.

Appendix B contains three test curves to show the results of the cubic spline procedure. The first example demonstrates the ability of the cubic spline method to approximate a given analytical expression. This ability is shown by a comparison of the slopes and curvatures of the cubic spline and the given expression. The second example shows the effect of the smoothing routine. Example 3 demonstrates the results of the interpolation and offset routines.

DISCUSSION

INTRODUCTION

Numerically controlled machine tools are now available which can machine complex shapes to extremely close tolerances. In order to use these machines efficiently the shape of the part must be precisely defined in mathematical terms. This often requires that a "smooth" curve be fitted to a set of points.

Other methods of curve fitting include the use of straight lines, general polynomials, or conic sections. These methods, however, have disadvantages. For example, the use of straight lines may give a rough and discontinuous curve unless a very large number of points is available. General polynomials tend to give an excessive number of inflections and become unwieldy if the number of points is large. Resolving this problem by a least-squares fit leads to a lack of control about how close the curve comes to the given points. Numerous other curve fitting routines are available but their use is usually limited to the skilled mathematician. A need exists for a procedure which can be used routinely by technicians.

The basic problem is that of passing a smooth curve through an ordered set of points. To be smooth, a curve should have continuous position, slope, and curvature; and, since curvature is an indication of roughness, the curvature should be small in magnitude and well behaved. A widely accepted measure of smoothness is the integral of curvature squared with respect to arc length:

$$\int K^2 \, ds.$$

When applied to elastic membranes, this integral is referred to as the strain energy. An infinitely thin elastic spline, under a given set of restaints, assumes a position of minimum strain energy; that is, it assumes the most relaxed position.(2) The resulting curve is accepted generally as being smooth. In effect then a curve is desired which approximates a spline, using a given set of points as restraints. The approach presented here is called a cubic spline fit. It consists of determining a set of cubic equations, one cubic for each interval, such that each cubic passes through the points defining the beginning and end of its interval and has a slope and curvature at these end points which match those of the adjacent cubics at these points.

For the sake of uniformity most of the variables in the following discussion have the same name as used in the fortran program. A complete list of the variables with their definitions is given in Table A-1 of Appendix A. Since a variable symbol may contain more than one character, multiplication is always indicated by an asterisk (*).

CALCULATION OF THE POINT PARAMETERS

The parameters defined in this and the next sections are shown in Figure 1. The given datapoints may be specified in either polar or cartesian form. Since both forms are used by the program, the following equations are used to convert from the given form to the other form:

$$R = \sqrt{X^2 + Y^2}$$

and

$$\theta = \text{arctangent } Y/X$$

or

$$X = R * \text{cosine } \theta$$

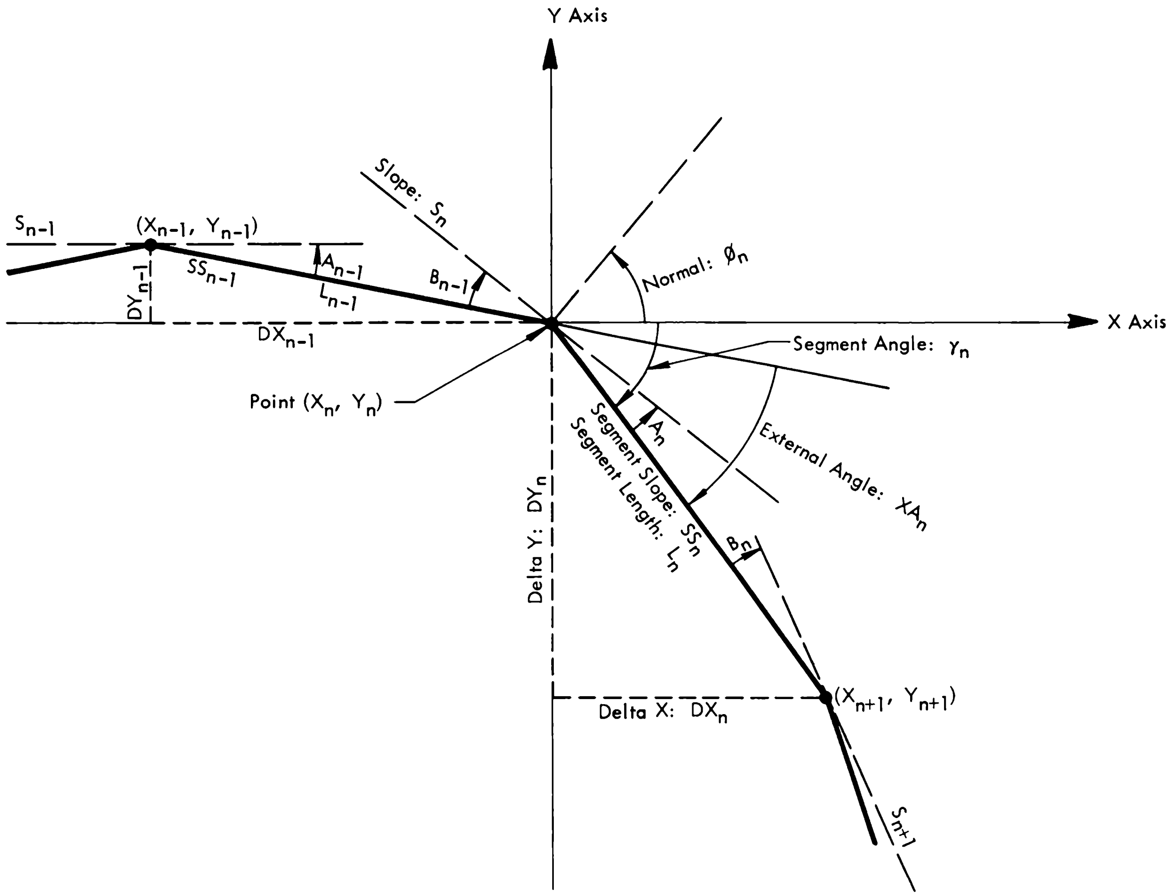

Figure 1. PARAMETER DESCRIPTION.

and

$$Y = R * \text{sine } \theta .$$

If the quantities

$$DX_n = X_{n+1} - X_n$$

and

$$DY_n = Y_{n+1} - Y_n$$

are found for the nth interval, then the slope of segment n is

$$SS_n = DY_n / DX_n$$

and the length of segment n is

$$L_n = \sqrt{DX_n^2 + DY_n^2} .$$

The segment angle, which is the angle the segment makes with the x axis, is

$$\gamma_n = \text{arctangent } SS_n \quad .$$

γ_n is not used in the cubic spline routine, but is useful in diagnostic work. The external angle is the angle between the segment under consideration and the previous segment; that is, it is a measure of the change in direction from one segment to the next. The tangent of the external angle is first found since the slopes of the two segments are known:

$$TXA_n = \frac{SS_n - SS_{n-1}}{1 + SS_n * SS_{n-1}} \quad .$$

CALCULATION OF THE CURVE PARAMETERS

Slopes

In order to produce the cubic spline through a given set of points, first it is necessary to have an estimate of the slope at each given point. Unless the slope, or the normal, is specified at a given point (X_n, Y_n), other than the first or last point of the contour, it is first approximated as the slope of the circle which passes through the point in question and the two adjacent points; namely, (X_{n-1}, Y_{n-1}) and (X_{n+1}, Y_{n+1}). The slope is found to be

$$S_n = \frac{(Y_{n+1} - Y_n) * (L_{n-1})^2 + (Y_n - Y_{n-1}) * (L_n)^2}{(X_{n+1} - X_n) * (L_{n-1})^2 + (X_n - X_{n-1}) * (L_n)^2} \quad . \tag{1}$$

The derivation of equation (1) is as follows. Consider the three points P_1, P_2, and P_3 on the contour with the translated origin at P_2, see Figure 2. If the center of the circle is $P_c(X_c, Y_c)$, then

$$(X_1 - X_c)^2 + (Y_1 - Y_c)^2 = r^2 \tag{2}$$

and

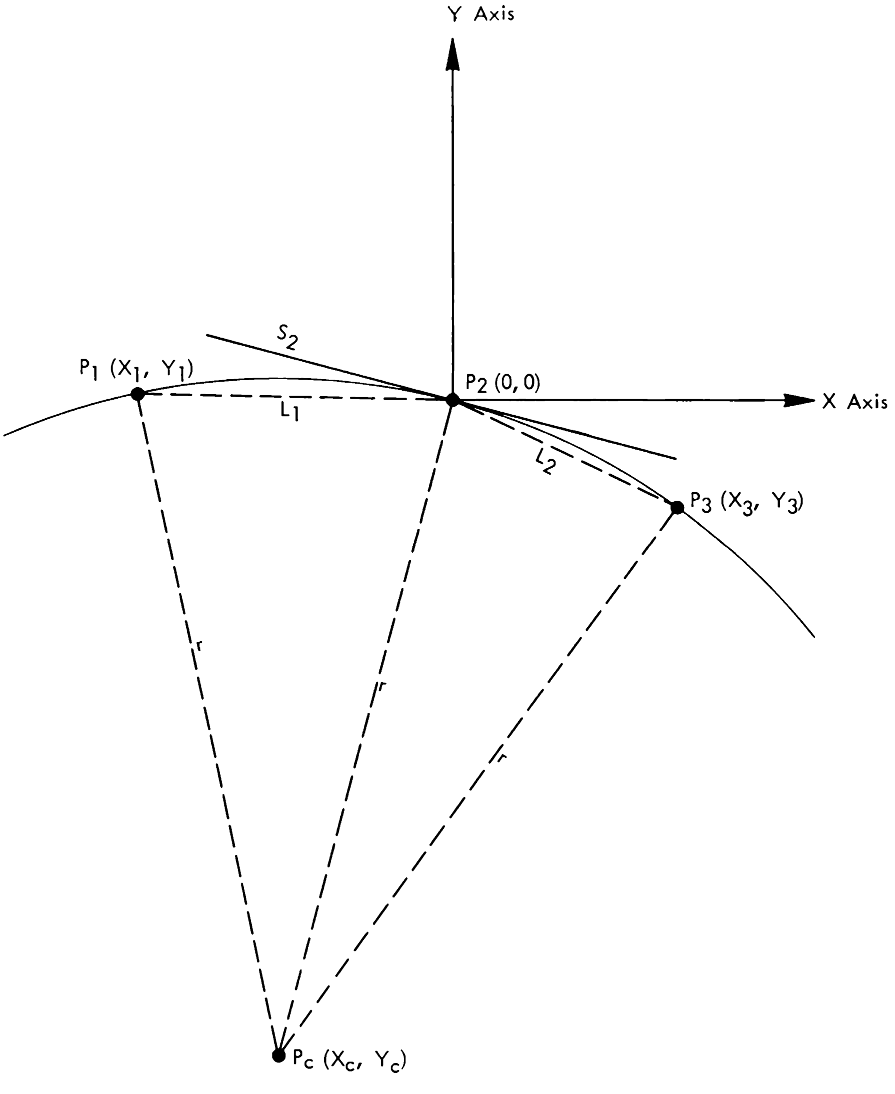

Figure 2. CALCULATION OF INITIAL SLOPES.

$$(X_3 - X_c)^2 + (Y_3 - Y_c)^2 = r^2 \ . \qquad (3)$$

Equation (2) may be expanded and rearranged to produce

$$X_1 * X_c + Y_1 * Y_c = \frac{X_1^2 + Y_1^2}{2}$$

or

$$X_1 * X_c + Y_1 * Y_c = \frac{L_1^2}{2} \ . \qquad (4)$$

Similarly, equation (3) may be written as

$$X_3 * X_c + Y_3 * Y_c = \frac{L_2^2}{2} \ . \qquad (5)$$

Equations (4) and (5) may be solved simultaneously to find

$$X_c = \frac{Y_3 * L_1^2 - Y_1 * L_2^2}{2 * (X_1 * Y_3 - X_3 * Y_1)} \qquad (6)$$

and

$$Y_c = \frac{-X_3 * L_1^2 + X_1 * L_2^2}{2 * (X_1 * Y_3 - X_3 * Y_1)} \ . \qquad (7)$$

Since the slope of the line connecting P_c and P_2 is Y_c/X_c, then

$$S_2 = -X_c/Y_c \ . \qquad (8)$$

If equations (6) and (7) are then substituted in equation (8),

$$S_2 = \frac{Y_3 * L_1^2 - Y_1 * L_2^2}{X_3 * L_1^2 - X_1 * L_2^2} \ . \qquad (9)$$

In terms of the incremental differences in successive coordinates, starting at P_1 and an arbitrary location of the origin at some point, P_n, on the contour, the following expressions are attained:

$$X_1 = DX_{n-1} = -(X_n - X_{n-1}),$$

$$X_3 = DX_n = X_{n+1} - X_n,$$

$$Y_1 = DY_{n-1} = (Y_n - Y_{n-1}),$$

and

$$Y_3 = DY_n = Y_{n+1} - Y_n \ .$$

If these expressions are substituted into equation (9), the expression for S_n, as shown in equation (1), is achieved.

Special conditions must be established for the first and last points of a contour. These slopes may be given as input data, in which case they act as additional restraints upon the spline to be fitted. If they are not given, then they are treated as variables and assumed to be the slope of the circle which passes through the end point and the adjacent point with the previously found slope at the latter point.

If the slope, S_n, is given as input or is calculated, then the normal $\emptyset_n$, is

$$\emptyset_n = \text{arctangent}\,(-1/S_n)\,.$$

If the normal is specified by input, then

$$S_n = -\text{cosine}\ \emptyset_n \,/\, \text{sine}\ \emptyset_n \ .$$

It should be noted that a slope, or normal, which is specified as input data will not be used unless that point is the first or last point of a contour. If slopes are known at several points along a contour, the contour may be broken at these points into several shorter contours, thus allowing the required slopes to remain fixed. Once initial slopes have been assigned to each point, four conditions (ie, end points and end slopes) have been established which must be satisfied by whatever expression is selected to define the curve between each pair of points.

Normalized Cubic

There are, of course, many expressions which could be used to meet the four established conditions. A third-degree polynomial was selected because it is the simplest expression which always exists and is never ambiguous. A general third-degree polynomial equation, which is of the form:

$$Ax^3 + Bx^2y + Cxy^2 + Dy^3 + Ex^2 + Fxy + Gy^2 + Hx + Iy + J = 0,$$

may be simplified by translation and rotation of the coordinate frame. If this normalizing procedure is applied to the four specified conditions by selecting a u-v coordinate frame such that the first point of a pair lies at the origin and the second point lies on the positive u axis, a normalized cubic is obtained which is of the form

$$v = C_1 * u^3 + C_2 * u^2 + C_3 * u + C_4 . \tag{10}$$

Let the slopes of the desired cubic be TA at the point (0, 0) and TB at the point (L, 0), where L is the distance between the two points. It is necessary to obtain expressions for C_1, C_2, C_3, and C_4 in terms of the known quantities L, TA, and TB. If $u = 0$, then it is seen in Figure 3 that $v = 0$. Hence, substituting these values in equation (10),

$$C_4 = 0.$$

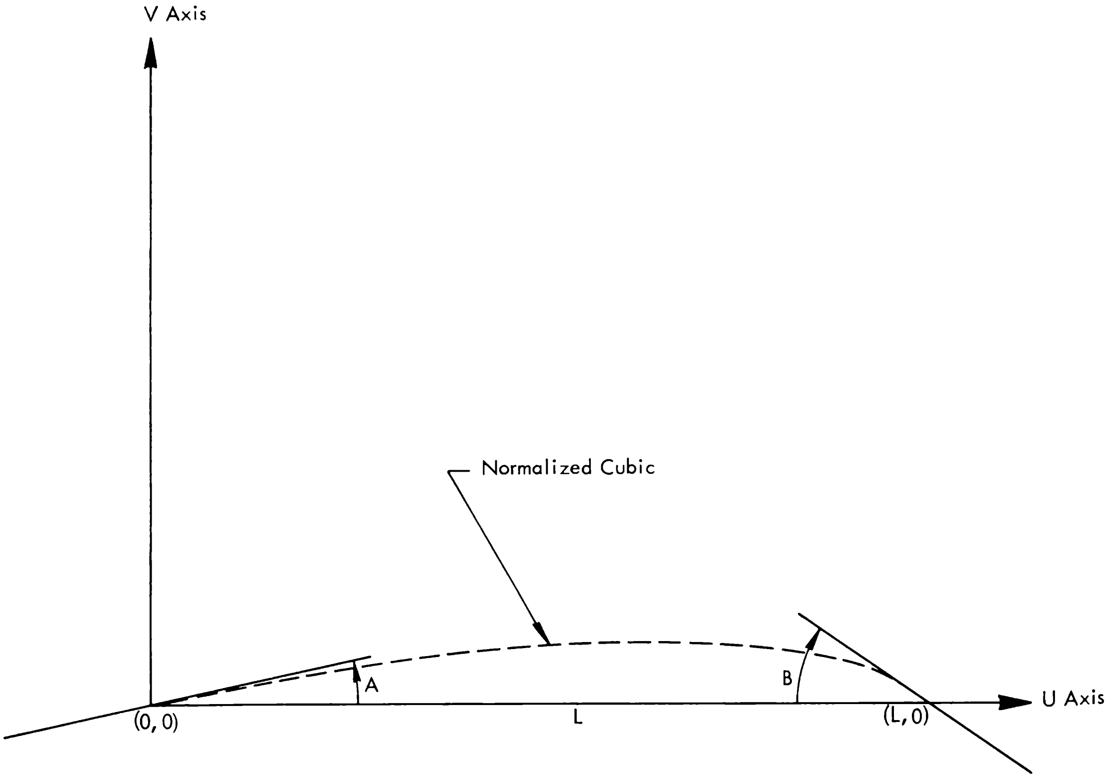

Figure 3. NORMALIZED CUBIC.

The slope, dv/du, of the cubic in question is the derivative of equation (10), or

$$\frac{dv}{du} = 3 * C_1 * u^2 + 2 * C_2 * u + C_3. \qquad (11)$$

When u = 0, dv/du = TA; therefore, substituting these values in equation (11),

$$C_3 = TA.$$

At the point (L, 0), where dv/du = TB, equation (10) becomes

$$C_1 * L^2 + C_2 * L + TA = 0, \qquad (12)$$

and equation (11) becomes

$$3 * C_1 * L^2 + 2 * C_2 * L + TA = TB . \qquad (13)$$

If equation (13) is solved for C_1,

$$C_1 = \frac{TB - TA - 2 * C_2 * L}{3 * L^2} ,$$

and substituted in equation (12), then

$$C_2 = - \frac{2 * TA + TB}{L}$$

and hence

$$C_1 = \frac{TA + TB}{L^2} .$$

If the expressions for C_1, C_2, C_3, and C_4 are now substituted in equation (10), the desired equation for segment n is

$$v = \left[\frac{TA + TB}{L^2}\right] * u^3 + \left[- \frac{2 * TA + TB}{L}\right] * u^2 + \left[TA\right] * u. \qquad (14)$$

It then follows that

$$\frac{dv}{du} = 3 * \left[\frac{TA + TB}{L^2}\right] * u^2 + 2 * \left[- \frac{2 * TA + TB}{L}\right] * u + TA \qquad (15)$$

and

$$\frac{d^2v}{du^2} = 6 * \left[\frac{TA + TB}{L^2}\right] * u + 2 * \left[-\frac{2 * TA + TB}{L}\right]. \qquad (16)$$

Obviously, the normalized cubic, as shown in equation (14), with only three coefficients is a much simpler form. It has the restrictions that the two points must not be coincident and the slopes must not be infinite in the normalized coordinate frame. However, these are trivial restrictions in practical curve-fitting problems. In practical cases the points are never coincident and are spaced sufficiently dense that the curve does not take radical excursions between any two adjacent points; that is, the change in slope is reasonably small. The normalized cubic has the additional property of minimizing the magnitude of the slope, which means that the curvature will be approximately linear.

The values of TA_n and TB_n are needed for each segment n to use in the normalized cubic equation, equation (14). Since A_n is the difference between the angle of the slope S_n, and the segment angle, γ_n, then

$$TA_n = \frac{S_n - SS_n}{1 + S_n * SS_n}.$$

Similarly, TB_n is the tangent of the difference between the angle of the slope, S_{n+1}, and the segment angle, γ_n, or

$$TB_n = \frac{S_{n+1} - SS_n}{1 + S_{n+1} * SS_n}.$$

The expression for the slope at the first point, if it was not specified as input data, may now be written:

$$S_1 = \frac{SS_1 + TA_1}{1 - SS_1 * TA_1},$$

where

$$TA_1 = - TB_1 .$$

Similarly, if the slope at the last point was not specified, then

$$S_n = \frac{SS_{n-1} + TB_{n-1}}{1 - SS_{n-1} * TB_{n-1}} ,$$

where

$$TB_{n-1} = - TA_{n-1} .$$

Curvature Matching

A sequence of normalized cubics meets the first two requirements for a smooth curve—continuous position and slope. However, if the curvatures of the two normalized cubics passing through a given point are evaluated, in general it is found that these curvatures are not equal. However, since the slopes are at our disposal, they may be manipulated to achieve the third requirement—continuous curvature.

In order to match the curvature at point P_n, first it is necessary to derive an expression for the difference in curvature at P_n of the two cubics intersecting there. The general expression for the curvature, K, of a plane curve, v, with first derivative, dv/du, and second derivative, d^2v/du^2, is

$$K = \frac{\frac{d^2v}{du^2}}{\left[1 + \left[\frac{dv}{du}\right]^2\right]^{3/2}} .$$

Using equations (15) and (16), the curvature for the normalized cubic becomes

$$K = \frac{2 * C_1 * u - C_2}{\left[1 + \left[C_1 * u^2 - C_2 * u + TA\right]^2\right]^{3/2}} , \tag{17}$$

where

$$C_1 = \frac{3 * (TA + TB)}{L^2}$$

and

$$C_2 = \frac{2 * (2 * TA + TB)}{L} .$$

Therefore, at the point (0, 0) in segment n, where

$$\frac{dv}{du} = TA_n$$

and

$$\frac{d^2v}{du^2} = -\frac{4 * TA_n + 2 * TB_n}{L_n},$$

the curvature is

$$CA_n = -\frac{4 * TA_n + 2 * TB_n}{L_n * \left[1 + TA_n^{\,2}\right]^{3/2}} .$$

Similarly, at the point $(L_{n-1}, 0)$ in segment n-1, where

$$\frac{dv}{du} = TB_{n-1}$$

and

$$\frac{d^2v}{du^2} = \frac{2 * TA_{n-1} + TB_{n-1}}{L_{n-1}},$$

the curvature is

$$CB_{n-1} = \frac{2 * TA_{n-1} + TB_{n-1}}{L_{n-1} * \left[1 + TB_{n-1}^{\,2}\right]^{3/2}} .$$

The difference in curvature at P_n may be expressed as

$$DC_n = CA_n - CB_{n-1},$$

or, using the expressions just derived,

$$DC_n = -\frac{4 * TA_n + 2 * TB_n}{L_n * \left[1 + TA_n^{\,2}\right]^{3/2}} - \frac{2 * TA_{n-1} + TB_{n-1}}{L_{n-1} * \left[1 + TB_{n-1}^{\,2}\right]^{3/2}} . \qquad (18)$$

The required result is to make DC_n less than or equal to some arbitrarily small value. The procedure for matching the curvature at P_n is iterative. As each point is taken in turn, the slope of that point is adjusted until the two adjacent normalized cubics have matched curvature. Since the segment length, L_n, is fixed, then TA_n and TB_{n-1} must be altered to bring about the desired continuity. However, if TA_n is specified at a given point, then, in order to make the slope continuous, TB_{n-1} is uniquely determined. Hence, it is necessary to alter TA_n until DC_n is sufficiently small. The technique used is Newton's method of root approximations; that is,

$$X_{new} = X_{old} - f(X)/f'(X) \ .$$

If

$$f(X) = DC_n ,$$

then f' (X) is the derivative of equation (18) with respect to TA_n:

$$\frac{d(DC_n)}{d(TA_n)} = - \frac{2 * (1 + TXA_n^2) * \left[2 - TB_{n-1} * (3 * TA_{n-1} + 4 * TB_{n-1})\right]}{(1 + TXA_n * TA_n)^2 * L_{n-1} * (1 + TB_{n-1}^2)^{5/2}} - \frac{2 * \left[2 - TA_n * (3 * TB_n + 4 * TA_n)\right]}{L_n * (1 + TA_n^2)^{5/2}} \ .$$

The following approximation, DCP, is used in the program instead of the expression just derived:

$$DCP_n = - 4 * \left[\frac{1}{L_{n-1}} + \frac{1}{L_n}\right] \ .$$

This signed approximation provides rapid convergence and insures that the proper solution is found. Therefore, in the particular case under consideration, successive new values for TA_n are found according to the equation:

$$TA_n \text{ (new)} = TA_n \text{ (old)} + \frac{CA_n - CB_{n-1}}{4 * \left[\frac{1}{L_{n-1}} + \frac{1}{L_n}\right]} ,$$

until the resulting DC_n is sufficiently small. As this procedure is repeated for each point in turn, the match at nearby points is disturbed. Therefore, it is necessary to repeat the entire matching process. The procedure converges rapidly, and a match at all points is achieved in a few passes. The result is a cubic spline; that is, a series of normalized cubics which is continuous in position, slope, and curvature (see Example 1 in Appendix B).

STRAIN ENERGY

Along with the output lists of the cubic spline parameters (see Appendix A) is a plot of the curvature. The output is supplied primarily for diagnostic purposes and may be used to readily reconstruct any portion of the curve for detailed examination. The curvature plot reveals at a glance any unexpected irregularities such as bumps, inflections, and roughness.

It is with regard to the last characteristic, roughness, that a more detailed discussion is required. Since the purpose was to put a "smooth" curve through a set of points, then the question arises: "What is meant by a smooth curve being rough?". It should be noted that the cubic spline passes precisely through the given points. This is in contrast to the normal procedure in which the form of the equation is predetermined and the technique is one of choosing the values of the coefficients that go into the equation such that the curve will pass near the points. Indeed, no presumptions about the form of the curve have been made. Its shape is determined entirely by the points. Now it may be that the points were generated by a process that introduced local roughness; for example, measurements taken from a hand-sculptored clay model where the shape has been determined by pushing a little here and pulling a little there, or from an artist's freehand sketch. Experimentally derived data often exhibit this characteristic, as do points obtained by evaluating some known expression if care is not taken in the spacing of the points and in rounding off numbers. Such processes produce the overall or general shape but result in local roughness; that is, a curve with "high-frequency" components. Since the cubic spline passes exactly through the points, any intrinsic roughness is preserved, and this is displayed as a saw-tooth pattern in the curvature plot.

It is usually desirable to remove this local roughness; that is, to remove the high-frequency components while preserving the low-frequency components. For this, the strain energy, where the strain energy, E, of a curve, v, is defined as

$$E = \int K^2 \, ds$$

plays an important role. The strain energy,

$$E_n = \int_0^{L_n} K^2 \, du ,$$

where K is defined by equation (17), is calculated for each normalized cubic. The sum of all E_n is called the strain energy of the cubic spline. By far the largest contributors to the strain energy of a cubic spline are the high-frequency components. Therefore, a procedure was developed which gives the direction and an approximate distance within some specified maximum limit that a point needs to be moved to reduce the strain energy. The efficiency of the iterative procedure used to smooth out local roughness may be observed in the change of the strain energy of the fitted curve (see Example 2 in Appendix B).

POINT ADJUSTMENT

As has been stated, the saw-tooth pattern represents kinks in the curvature. The fact that the curvature of the normalized cubic is approximately linear may be used to decrease or even eliminate these kinks. If the opposite end points of two adjacent segments which represent a kink are treated as the end points of one segment, the normalized cubic fitted to this segment will miss the center point by an amount and direction which would reduce the kink. By moving each point the calculated distance in the proper direction; that is, by moving each point toward the normalized cubic through the two adjacent points, the kinks in the curvature are reduced, and hence the strain energy is reduced. The first and last points are never adjusted.

To determine the distance and direction that point P_n is to be moved, consider the points P_{n-1} and P_{n+1} and the slopes S_{n-1} and S_{n+1} of the cubic spline at these points (see Figure 4). As before, the parameters needed for the normalized cubic, equation (14), are the tangent of angle α, the tangent of angle β, and the distance between the two points. The length of the segment is

$$TL = \sqrt{(X_{n+1} - X_{n-1})^2 + (Y_{n+1} - Y_{n-1})^2} .$$

Angle α is the difference between the angle of the slope, S_{n-1}, and the angle of the segment. Since the slope of the segment is

$$SST = \frac{Y_{n+1} - Y_{n-1}}{X_{n+1} - X_{n-1}} ,$$

then the tangent of α is

$$TALPHA = \frac{S_{n-1} - SST}{1 + S_{n-1} * SST} .$$

Similarly, the tangent of angle β is

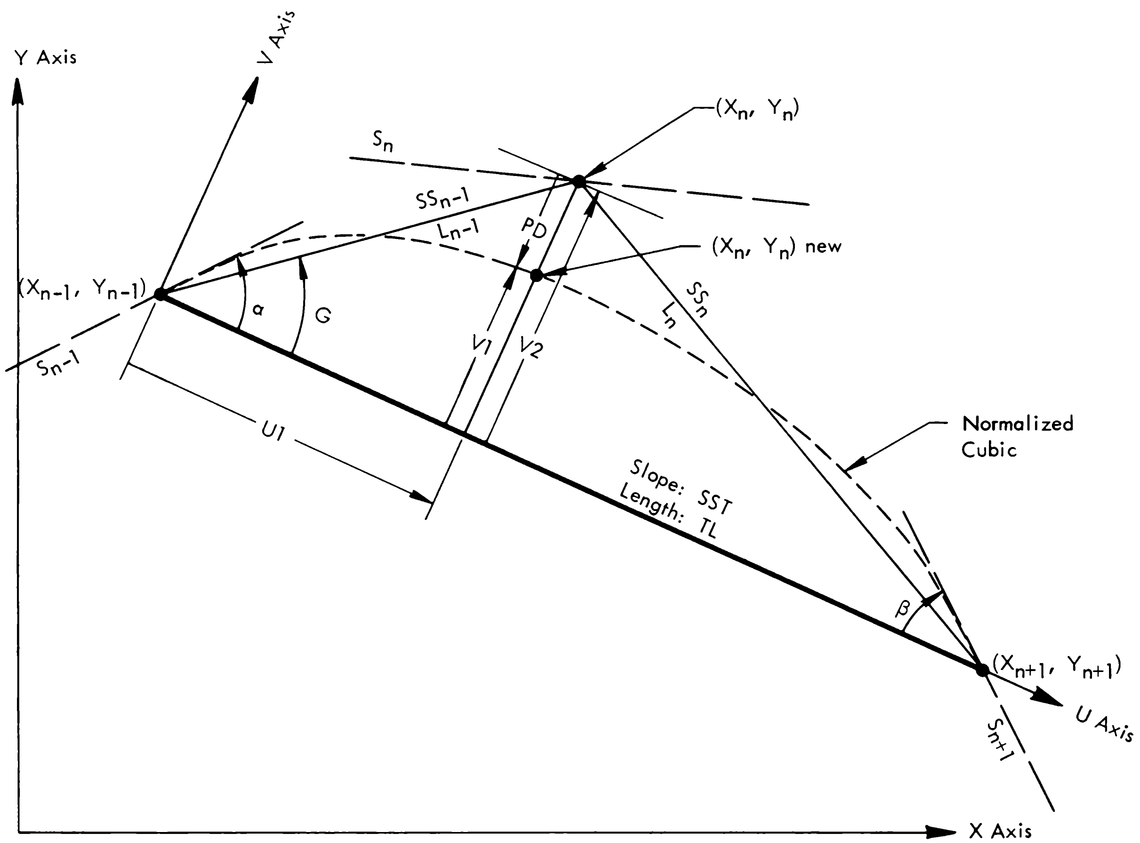

Figure 4. POINT MOVEMENT.

$$TBETA = \frac{S_{n+1} - SST}{1 + S_{n+1} * SST} \quad .$$

The distance the point is to be moved is the distance from P_n along a line through P_n perpendicular to the segment; that is, u axis, to the normalized cubic. In order to evaluate the cubic, first it is necessary to find at what point the perpendicular line intersects the u axis. From Figure 4 it may be seen that

$$\text{cosine } G = \frac{U1}{L_{n-1}} \, , \tag{19}$$

and, using the law of cosines,

$$L_n^2 = L_{n-1}^2 + TL^2 - 2 * TL * L_{n-1} * \text{cosine } G \, ,$$

or

$$\text{cosine } G = \frac{L_{n-1}^2 + TL^2 - L_n^2}{2 * TL * L_{n-1}} \quad . \tag{20}$$

Hence, if equations (19) and (20) are combined and solved for U1,

$$U1 = \frac{L_{n-1}^2 + TL^2 - L_n^2}{2 * TL} .$$

Thus, the point at which the cubic and the perpendicular line intersect may be found using equation (14):

$$V1 = \left[\frac{TALPHA + TBETA}{L^2}\right] * U1^3 - \left[\frac{2 * TALPHA + TBETA}{TL}\right] * U1^2 + TALPHA * U1 .$$

The coordinate V2 is found using the following equation which is derived from the standard equation for the distance from a point to a line:

$$V2 = \frac{(X_{n+1} - X_{n-1}) * (Y_n - Y_{n-1}) - (Y_{n+1} - Y_{n-1}) * (X_n - X_{n-1})}{TL} .$$

The predicted distance of movement, PD, is then

$$PD = V1 - V2 .$$

The program gives the user the option of specifying what fraction of PD the point is actually moved. Initially this fraction is set at 1/2 by the program. This value seems to make optimum use of the spline-fitting routine while not overcorrecting, since the effect of the point's movement upon adjacent cubics is not considered at this time. The coordinate, V1, of the new point is adjusted according to the fraction specified. The coordinates of the new point in the original x-y system are then found:

$$XN_n = X_{n-1} + \frac{U1 * (X_{n+1} - X_{n-1}) - V1 * (Y_{n+1} - Y_{n-1})}{TL}$$

and

$$YN_n = Y_{n-1} + \frac{U1 * (Y_{n+1} - Y_{n-1}) + V1 * (X_{n+1} - X_{n-1})}{TL} .$$

Four types of point movement, which may vary from point to point, are allowed in the program; namely, (1) x and y movement perpendicular to the line segment as just discussed, (2) movement parallel to the x axis, (3) movement parallel to the y axis, and (4) radial movement. If the x and y method is selected, there is only one remaining check required. The distance, TDM, from the original point to the newly found point is calculated. If this distance is greater than the maximum allowable movement, TOLRAD, specified by the user, then the new point must be readjusted. The point (XN_n, YN_n) is adjusted to be the intersection of the line connecting the original point (XO_n, YO_n) and the new point (XN_n, YN_n) with the circle with radius equal to TOLRAD and center at (XO_n, YO_n) (see Figure 5). The new adjusted point then has the coordinates

$$XNA_n = XO_n + \frac{TOLRAD}{TDM} * (XN_n - XO_n)$$

and

$$YNA_n = YO_n + \frac{TOLRAD}{TDM} * (YN_n - YO_n) \ .$$

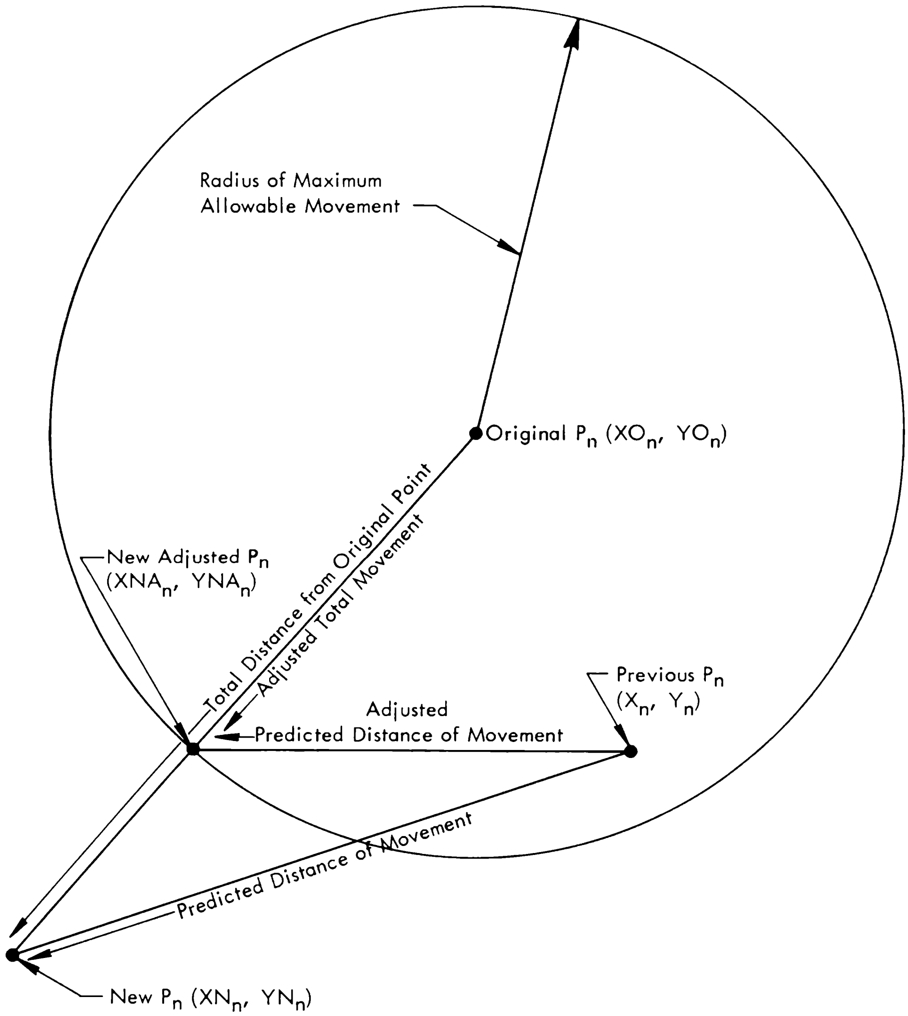

Figure 5. **ADJUSTMENT OF NEW POINT BASED ON TOTAL DISTANCE MOVED.**

If the method of moving the point parallel to the x axis is selected, the new coordinates are

$$XN_n = X_n + PD$$

and

$$YN_n = Y_n \ .$$

If the difference between XN_n and XO_n is greater than TOLRAD, then the new point is moved toward the original point until it is at the maximum distance. Similarly, if the movement is selected to be parallel to the y axis, the new coordinates are

$$XN_n = X_n$$

and

$$YN_n = Y_n + PD \ ,$$

and the corresponding adjustment is made if required. If radial movement is selected, the radius is changed by the amount PD and then adjusted if it differs too much from the original radius. In order to make most efficient use of the last three methods, points should be moved parallel to the x axis only if the curve is almost parallel to the y axis; points should be moved parallel to the y axis only if the curve is almost parallel to the x axis, and radial movement should be selected only if the contour is essentially circular.

It must be recognized that moving a point influences conditions in adjacent segments; however, satisfactory results are obtained when these side effects are ignored during the point-movement routine. After all points have been moved the proper amount, a new cubic spline is calculated and the strain energy of the new cubic spline is found. The decrease in strain energy between successive splines is a monotonically decreasing series; that is, each new cubic spline has a lower strain energy than all previous splines and the strain energy decreases by a smaller amount with each new cubic spline. This suggests that the iteration (that of moving points and passing cubic splines through the new points) be terminated when the fractional decrease in strain energy between two successive passes; that is,

$$\frac{\text{Energy (previous)} - \text{Energy (new)}}{\text{Energy (previous)}} \ ,$$

is less than some specified amount. As a protection, the iteration may also be stopped after a given number of passes. The number of passes may be specified by the user; however, it is initialized at 100.

INTERPOLATION

The usual reason for fitting a curve through a set of points is to define the region between the points in order that intermediate points may be calculated. The method presented here finds points on the cubic spline spaced uniformly along the chord connecting two adjacent given points. The distances between the interpolated points in each segment depend basically upon the parameter, TOLER. TOLER is the maximum allowable normal distance from the broken line passing through the given and interpolated points to the cubic spline.

Assume that the curve under consideration is a circular arc, as shown in Figure 6. Since

$$D1 = \frac{\sqrt{4 * R^2 - D2^2}}{2} ,$$

then

$$D3 = R * \left(1 - \sqrt{1 - \frac{D2^2}{4 * R^2}}\right).$$

Using the approximation

$$\sqrt{\frac{1 - D2^2}{4 * R^2}} \approx 1 - \frac{D2^2}{8 * R^2} ,$$

$$D3 \approx TOLER = \frac{D2^2}{8 * R} .$$

For a given value of TOLER, the maximum length of the chord between two adjacent points is

$$D2 = \sqrt{8 * TOLER * R} .$$

Since $R = 1/C_1$, where C_1 is the curvature of the circle, then the expression for D2 may also be written as

$$D2 = \sqrt{\frac{8 * TOLER}{C_1}} .$$

Therefore, the number of points, K_n, needed in a given segment n is

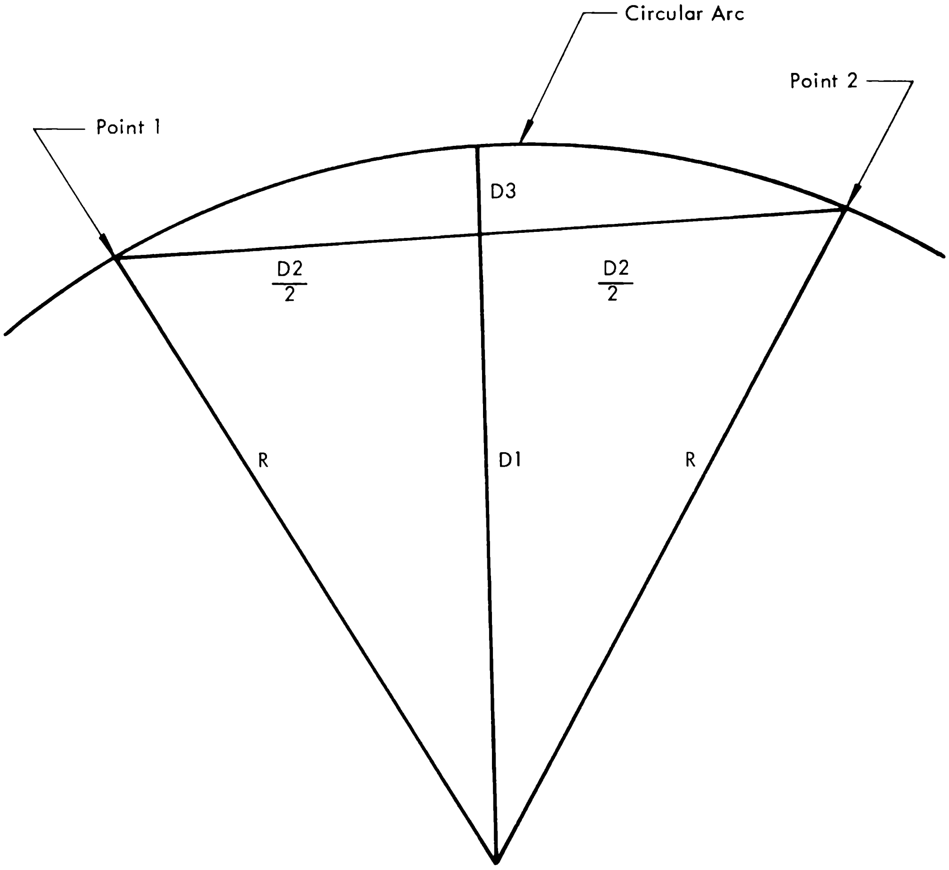

Figure 6. SCALLOP TOLERANCE D3.

$$K_n = \frac{L_n}{\sqrt{\frac{8 * TOLER}{C_1}}} .$$

Since the curvature of the cubic spline is essentially linear in a given segment, C_1 is selected as the larger of CA_n and CB_n, thus assuring sufficient points will be found.

To find K_n intermediate points spaced uniformly along the u axis of segment n, let

$$U_J = L_n * \frac{J}{K_n + 1},$$

where J designates the point along the segment being calculated and U_J is the u coordinate of that point. The v coordinate, V_J, of point J may then be found by substituting U_J in equation (14), Page 20. If the terms are rearranged, then

$$V_J = L_n * \left(\frac{J}{K_n + 1}\right) * \left(\frac{K_n + 1 - J}{K_n + 1}\right) * \left[TA_n * \left(\frac{K_n + 1 - J}{K_n + 1}\right) - TB_n * \left(\frac{J}{K_n - 1}\right)\right].$$

Using the equations

$$X_J = X_n + U_J * \text{cosine}\,\gamma_n - V_J * \text{sine}\,\gamma_n$$

and

$$Y_J = Y_n + U_J * \text{sine}\,\gamma_n + V_J * \text{cosine}\,\gamma_n,$$

where

$$\text{cosine}\,\gamma_n = \frac{X_{n+1} - X_n}{L_n}$$

and

$$\text{sine}\,\gamma_n = \frac{Y_{n+1} - Y_n}{L_n} \quad ,$$

the coordinates of point J become

$$X_J = X_n + (X_{n+1} - X_n) * \left(\frac{J}{K_n + 1}\right) - (Y_{n+1} - Y_n) * C_2$$

and

$$Y_J = Y_n + (Y_{n+1} - Y_n) * \left(\frac{J}{K_n + 1}\right) + (X_{n+1} - X_n) * C_2 \,,$$

where

$$C_2 = \left(\frac{J}{K_n + 1}\right) * \left(\frac{K_n + 1 - J}{K_n + 1}\right) * \left[TA_n * \left(\frac{K_n + 1 - J}{K_n + 1}\right) - TB_n * \left(\frac{J}{K_n + 1}\right)\right] \,.$$

OFFSET POINTS

It is frequently desired to find points offset from a fitted cubic spline. Since it may be that the amount of offset for a given curve may vary, a provision has been included in the routine whereby the offset, RJ, at a point is the sum, OFFSET, of a component, OFFVAR, which may vary from point to point, and a component, OFFCON, which is constant for all points. The offset at the K_n intermediate points in segment n is made to vary linearly between the offsets at P_n and P_{n+1}. Hence, at point J in segment n, the offset is

$$RJ = OFFSET_n + \left(\frac{J-1}{K_n+1}\right) * (OFFSET_{n+1} - OFFSET_n) .$$

The technique used to find the point offset from point J consists of finding the intersection of the two lines parallel to the segments intersecting at J at a distance equal to the offset RJ from the segments. It should be noted that the offset point found by this procedure is not the point normally offset a distance RJ from the point J. It is, however, that point through which the center of a tool must pass. The edge of a tool with a radius of RJ whose center is passed along the straight lines connecting these offset points would describe the broken line, inscribed in the cubic spline, determined by the interpolation procedure described in the last section (see Figure 7). These two types of offset points would coincide if and only if the point J lies on a straight line connecting the two adjacent points.

Let P_1(XQ1, YQ1), P_2(XQ2, YQ2), and P_3(XQ3, YQ3) be three consecutive points, given or interpolated, on the cubic spline (see Figure 8). The following parameters are required in order to find the offset point, PTC:

$$U11 = XQ2 - XQ1 ,$$

$$V11 = YQ2 - YQ1 ,$$

$$QL1 = \sqrt{U11^2 + V11^2} ,$$

$$U12 = XQ3 - XQ2 ,$$

$$V12 = YQ3 - YQ2 ,$$

and

$$QL2 = \sqrt{U12^2 + V12^2} .$$

The steps required to find PTC are as follows:

1. Find the coordinates of points PQ1 and PQ2.

2. Find the midpoint, PC, of the line connecting PQ1 and PQ2.

3. Find the x and y components, GXP and GYP, respectively, of the distance D_5 from P_2 to PC.

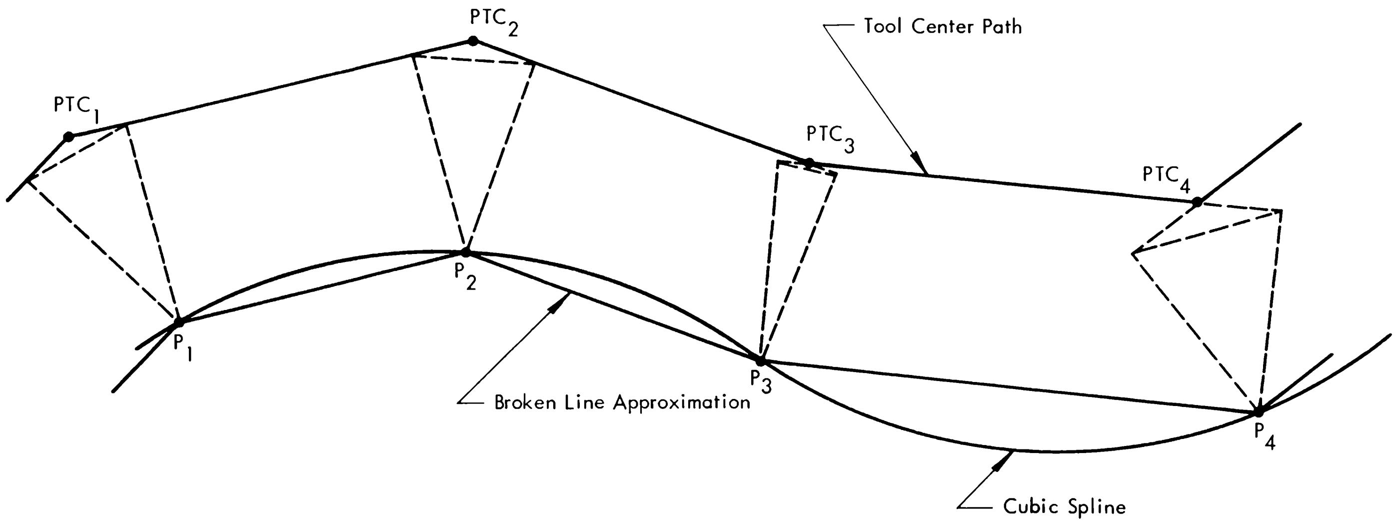

Figure 7. CONSTRUCTION OF THE TOOL CENTER PATH FROM THE BROKEN LINE APPROXIMATION TO THE CUBIC SPLINE.

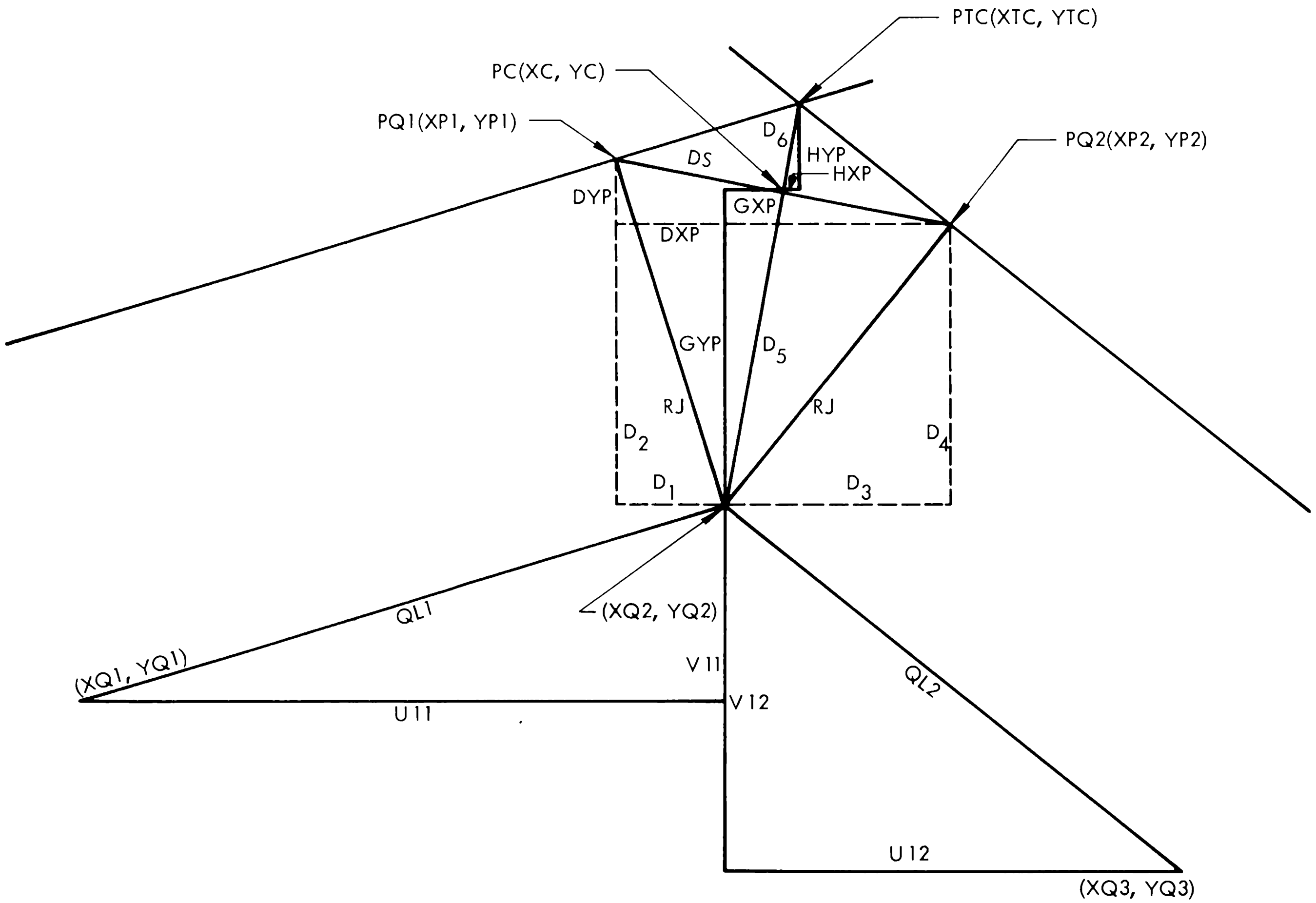

Figure 8. DETAILED CALCULATION FOR A TOOL CENTER.

4. Find the distance D_6 from PC to PTC.

5. Find the x and y components, HXP and HYP, respectively, of the distance D_6.

6. Find the coordinates XTC and YTC by adding the proper components to XQ2 and YQ2.

The coordinates XP1 and YP1 of point PQ1 may be found using the similar triangles defined by the lines D_1, D_2, and RJ and the lines U11, V11, and QL1, as shown in Figure 8.

It may be seen that

$$\frac{D_1}{RJ} = -\frac{V11}{QL1}$$

or

$$D_1 = -\frac{V11 * RJ}{QL1}$$

and

$$\frac{D_2}{RJ} = \frac{U11}{QL1}$$

or

$$D_2 = \frac{U11 * RJ}{QL1} .$$

Hence,

$$XP1 = XQ2 - \frac{V11 * RJ}{QL1}$$

and

$$YP1 = YQ2 + \frac{U11 * RJ}{QL1} .$$

Similarly, using the similar triangles defined by the lines D_3, D_4, and RJ and the lines U12, V12, and QL2,

$$XP2 = XQ2 - \frac{V12 * RJ}{QL2}$$

and

$$YP2 = YQ2 + \frac{U12 * RJ}{QL2} \quad .$$

The components of DS may now be stated as

$$DXP = XP2 - XP1$$

and

$$DYP = YP2 - YP1 \quad .$$

Since the triangle defined by the points P_2, PQ1, and PQ2 is isosceles, then the line from P_2 to PC is the perpendicular bisector of the line, DS, from PQ1 to PQ2. Therefore, the x and y components of the line D_5 are

$$GXP = -\frac{RJ}{2} * \left(\frac{V12}{QL2} - \frac{V11}{QL1}\right) \tag{21}$$

and

$$GYP = \frac{RJ}{2} * \left(\frac{U12}{QL2} - \frac{U11}{QL1}\right) , \tag{22}$$

respectively.

Since the distance from PQ1 to PC is DS/2 and the right triangles defined by the points PQ1, PC, and P_2 and the points PQ1, PC, and PTC are similar, then

$$\frac{D_6}{\frac{DS}{2}} = \frac{\frac{DS}{2}}{D_5} \quad ,$$

where

$$DS = \sqrt{DXP^2 + DYP^2} \quad .$$

And, using the first right triangle (PQ1, PC, and P_2),

$$D_5 = \sqrt{RJ^2 - \left(\frac{DS}{2}\right)^2} \quad .$$

That is,

$$D_6 = \frac{DS^2}{2 * \sqrt{4 * RJ^2 - DS^2}} \quad .$$

By using the similar triangles defined by the lines DYP, DXP, and DS and the lines D_6, HXP, and HYP, the x and y components of D_6 may be expressed as

$$HXP = - \frac{D_6 * DYP}{DS} \qquad (23)$$

and

$$HYP = \frac{D_6 * DXP}{DS} \quad . \qquad (24)$$

Therefore, the coordinates of PTC may be expressed as

$$XTC = XQ2 - GXP - HXP \qquad (25)$$

and

$$YTC = YQ2 + GYP + HYP \ , \qquad (26)$$

where GXP, GYP, HXP, and HYP are defined by equations (21), (22), (23), and (24), respectively.

Actually, equations (25) and (26) must be altered to allow for the direction of offset desired. This is accomplished through the use of the parameter, SIDE. If the points are considered in the order in which they are given, SIDE is set equal to +1 if offset to the left is desired or to -1 if offset to the right is desired. The equations become

$$XTC = XQ2 - (GXP + HXP) * SIDE$$

and

$$YTC = YQ2 + (GYP + HYP) * SIDE \ .$$

REFERENCES

(1) Automatically Programmed Tools, Computer System for Numerically Controlled Tools, Sponsored by Aerospace Industries Association, Washington, D. C.

(2) Walsh, J. L., Ahlberg, J. H., and Hilson, E. N., "Best Approximation Properties of the Spline Fit", Journal of Mathematics and Mechanics, II, p 225 (1962).

(3) SHARE Distribution No. 1017, Argonne National Laboratory, Argonne, Illinois, (Richard F. King).

APPENDIX A

THE COMPUTER PROGRAM

PROGRAM INFORMATION

The cubic spline program requires six routines to be processed. All are written in fortran language. A seventh optional routine may be added by the user. The cubic spline program, excluding the input-output routines and the clock routine, required 8970 words of computer storage, as compiled by the present fortran compiler. The number of words required by each routine is shown in parenthesis in each of the discussions. The routines are listed starting on Page 51 in the same order as they are described in the paragraphs that follow:

Main Program (7587 words) - A general flow chart of the program is shown in Figure A-1. The circled numbers refer to statement numbers in the fortran program. A dictionary of the variables used in the program is included on Page 63. The definition and formats of the input variables are listed as comment cards in the fortran lists in Table A-1. Two tapes are required to run the cubic spline program. The program specifies Tape 10 as the input tape and Tape 6 as the output tape. Contours, which have fifty points and are reasonably smooth, take an average of about ten seconds to run on an IBM 7090 computer. The program allows as many as 200 points in each contour. The output includes the following items:

1. A list of the input cards.

2. A list of the point parameters including polar and cartesian coordinates, the segment lengths, the angles the segments make with the x axis, and the distances the points have been moved.

3. A list of the curve parameters including: the slope, normal, and alpha angle at each given point, the tangent of the angle between the slope and the segment line and the curvature at each end of each cubic, the difference in curvature at each point, and the strain energy of each segment.

4. A plot of the curvature.

5. Lists 2, 3, and 4 after all point moving is completed.

Subroutine GRAPH (166 words) - This routine tests the values stored in the first argument to determine if a parameter should be plotted. If it is to be plotted, the routine then finds the symbol used to designate that parameter and the printer column in which to plot it. Next, the routine assembles a format statement which plots the required values properly and lists the value of the first parameter. The format is returned in the second argument. Each time GRAPH is called the format for one line of the plot is produced.

Subroutine HEAD (114 words) - This routine writes a heading at the top of each page. The second argument specifies the tape to be used. The first argument is a flag specifying which of three items should be done. If the flag equals zero, the page counter is set to zero, the time is obtained from the computer clock, and the page counter is indexed. If it is negative, the time is obtained and the page counter is indexed. Finally, if it is positive, the page counter is indexed. The routine then writes the contour title, the time, the date, and the page number on the specified tape in an appropriate format.

Function VALUE (67 words) - This routine is called by routine RIEMAN to obtain the value of the function at each point. The function to be integrated in this case is the square of the curvature. The first argument contains the point at which the function is to be evaluated, and the second argument is an unused index.

Subroutine TAD - HEAD calls a routine named TAD which obtains the time and date from the computer clock. As each computer clocking system differs slightly, this routine has not been included. The program is fixed to run without TAD. If a routine TAD is added, the indicated changes should be made in HEAD (see Table A-1, section 3). The first argument of TAD is the time. HEAD expects the time as a five-digit integer. The first two digits are the hour, the second two are the minutes, and the last would represent the seconds to the nearest ten if a zero were added after it. The second argument of TAD is two words containing the date in alphanumeric form. In the first word, HEAD expects the month, a slash, the day, and a slash. The second word should contain the year and four blanks.

Function RIEMAN (321 words) - This routine requires a functional value, f, at a specified number of points, estimates the second derivative of f at each point, and then places integration intervals so as to achieve the desired accuracy. The integation is done in one step. The routine is SHARE distribution number 1017.[3] The first two arguments are the lower and upper bounds for the integration, respectively. The third argument specifies the number of points at which the program is to estimate the second derivative of the function and must be greater than two. The fourth argument is the absolute limit of error for the integration. The fifth argument is an index passed on to the function subprogram VALUE, but is not used by the cubic spline program.

Subroutine ANGLE (83 words) - This routine divides the second argument by the third argument and then finds the arctangent of the quotient. It then tests the second and third arguments to adjust the angle to between zero and 2π radians. The answer is returned in the first argument. ANGLE is used only to adjust the polar coordinate, θ.

Subroutine TC (632 words) - This routine is called by the main program if either a Type 13 or 14 card is in the data. The routine calculates interpolated points based on TOLER and offsets these points the required amount. It produces a list of the interpolated points and the offset points. The Type 13 card should precede the curve points, and Type 14 may appear anywhere in the data. The value in the latter card is used for all following points until a new card of that type is read. Note that the value will be zero until the first Type 14 card is read.

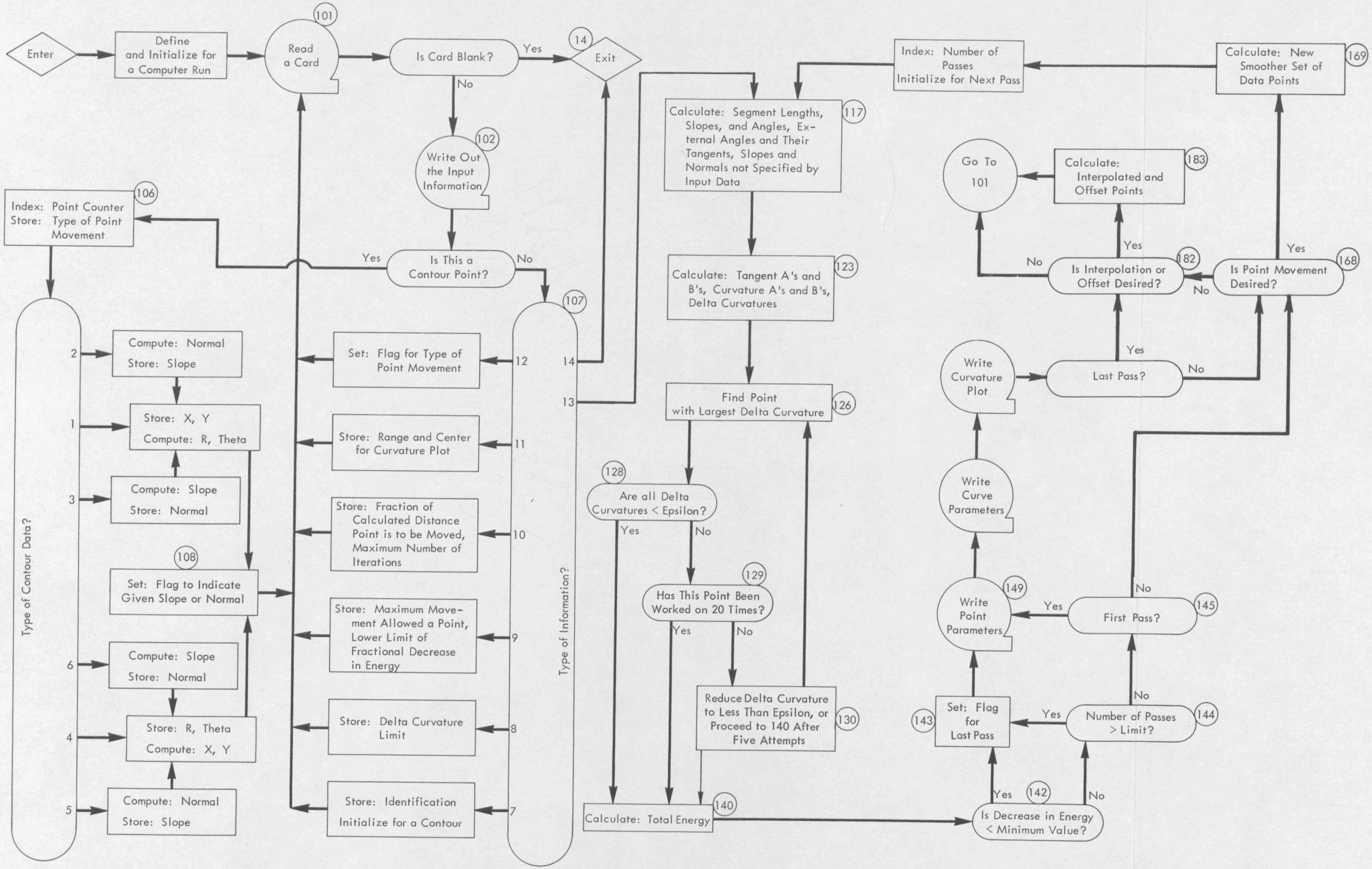

Figure A-1. MAIN PROGRAM FLOW CHART.

Table A-1

FORTRAN PROGRAMS

MAIN PROGRAM

```
C                CARD COLUMNS                                                          INITIAL
C23        11-19       21-29       31-39       41-70                                   VALUES
C
C01          X           Y
C02          X           Y         SLOPE
C03          X           Y         NORMAL
C04          R         THETA
C05          R         THETA       SLOPE
C06          R         THETA       NORMAL
C07                                          IDENTIFICATION                           (BLANK)
C08        EPSILON                           DELTA CURVATURE LIMIT                    (0.001)
C09        MAX MOVE    VALUE                 MAXIMUM POINT MOVEMENT ALLOWED            (0.0)
C                                            DELTA ENERGY LIMIT                        (0.0)
C10        FRACTION    ITER                  FRACTION OF PREDICTED DISTANE             (0.5)
C                                            LIMIT OF ITERATIONS FOR MOVING PTS        (100)
C11         RANGE     CENTER                 CURVATURE PLOT SCALES       (BASED ON DATA)
C12        TYP MOVE                          TYPE OF MOVEMENT-                         (1.0)
C                                              RADIAL # 1.0
C                                              PARALLEL TO X-AXIS # 2.0
C                                              PARALLEL TO Y-AXIS # 3.0
C                                              X AND Y # 4.0
C13         OFFFIX      TOLER      SIDE        FIXED PART OF OFFSET                      (0.0)
C                                             INTERPOLATION SCALLOP TOLERANCE          (10.0)
C                                             OFFSET SIDE- RIGHT#-1
C                                                             LEFT#1
C14         OFFVAR                            VARIABLE PART OF OFFSET                   (0.0)
C15                                           END OF CONTOUR DATA
C16                                           END OF COMPUTER RUN
C
C      NOTE- TYPE 7 MUST BE THE FIRST CARD FOR EACH CONTOUR
C            TYPE 15 MUST FOLLOW EACH CONTOUR
C
      CRVAF(Z1,Z2,Z3)#-(4.0*Z1+2.0*Z2)/(Z3*SQRTF((1.0+Z1**2)**3))           I    001
      CRVBF(Z1,Z2,Z3)# (2.0*Z1+4.0*Z2)/(Z3*SQRTF((1.0+Z2**2)**3))           I    002
      CRVF(Z1)#50.0*(Z1-CENTER)/CURVRG+51.50001                             I    003
      SMALF(Z1)#SIGNF(MAX1F(ABSF(Z1),1.0E-12),Z1)                           I    004
      TANMF(S1,S2)#(S1-S2)/SMALF(1.0+S1*S2)                                 I    005
      TANPF(S1,S2)#(S1+S2)/SMALF(1.0-S1*S2)                                 I    006
      DIMENSION RN(200),MOVTYP(200),FNT(106),T(5),TITLE(5),M(7),TH(200),I    007
     1R(200),X(200),Y(200),XL(200),GA(200),XA(200),S(200),PHI(200),D(2),I    008
     2AL(200),CA(200),CB(200),DC(200),EF(200),SS(200),TA(200),XLSQ(200),I    009
     3TB(200),TXA(200),LOOP(200),TABLE(8),XN(200),YN(200),TDM(200)       I    010
     4,OFFSET(200)                                                       I    011
      COMMON C1,C2,C3,TA,TB,XL,X,Y,PHI,CA,CB,OFFSET,N,TOLER,SIDE,TITLE  I    0115
      FRAC#0.5                                                          I    012
      ITER#100                                                          I    013
      EPS#0.001                                                         I    014
      TOLRAD#0.0                                                        I    015
      TOLFAC#0.0                                                        I    016
      CONV#57.295779513                                                 I    017
      HALFPI#1.570796327                                                I    018
      TABLE(1)#0.05                                                     I    019
      TABLE(2)#0.1                                                      I    020
      TABLE(3)#0.2                                                      I    021
      TABLE(4)#0.5                                                      I    022
      TABLE(5)#1.0                                                      I    023
      TABLE(6)#2.0                                                      I    024
      TABLE(7)#5.0                                                      I    025
      TABLE(8)#10.0                                                     I    026
```

Table A-1 (Cont'd)

```
101 READ INPUT TAPE 10,1000,NN,A,B,C,(T(I),I#1,5)                    1   027
    IF(NN)14,14,102                                                  1   028
102 IF(NN-7)103,7,103                                                1   029
103 IF(LINES)104,104,105                                             1   030
104 LINES#55                                                         1   031
    CALL HEAD (1,6,TITLE)                                            1   032
105 WRITE OUTPUT TAPE 6,1001,NN,A,B,C,(T(I),I#1,5)                   1   033
    LINES#LINES-1                                                    1   034
    IF(NN-7)106,7,107                                                1   035
106 N#N+1                                                            1   036
    MOVTYP(N)#MOVE                                                   1   037
    OFFSET(N)#OFFFIX+OFFVAR                                          1   038
107 GO TO (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16),NN                1   0385
  2 PHI(N)#ATANF(-1.0/SMALF(C))                                      1   039
    S(N)#C                                                           1   040
    GO TO 1                                                          1   041
  3 C#C/CONV                                                         1   042
    S(N)#-COSF(C)/SMALF(SINF(C))                                     1   043
    PHI(N)#C                                                         1   044
  1 X(N)#A                                                           1   045
    Y(N)#B                                                           1   046
    R(N)#SQRTF(A*A+B*B)                                              1   047
    CALL ANGLE (TH(N),B,A)                                           1   048
    GO TO 108                                                        1   049
  5 PHI(N)#ATANF(-1.0/SMALF(C))                                      1   050
    S(N)#C                                                           1   051
    GO TO 4                                                          1   052
  6 C#C/CONV                                                         1   053
    S(N)#-COSF(C)/SMALF(SINF(C))                                     1   054
    PHI(N)#C                                                         1   055
  4 R(N)#A                                                           1   056
    B#B/CONV                                                         1   057
    TH(N)#B                                                          1   058
    X(N)#A*COSF(B)                                                   1   059
    Y(N)#A*SINF(B)                                                   1   060
108 NP#NN                                                            1   061
    IF(N-1)109,109,101                                               1   062
109 NTYPE1#XABSF(XMODF(NN,3)-1)                                      1   063
    GO TO 101                                                        1   064
  7 DO 110 I#1,5                                                     1   065
110 TITLE(I)#T(I)                                                    1   066
    CALL HEAD(0,6,TITLE)                                             1   067
    LINES#55                                                         1   068
    MOVE#1                                                           1   069
    DO 111 J#1,7                                                     1   070
111 M(J)#0                                                           1   071
    DO 112 J#1,200                                                   1   072
    LOOP(J)#0                                                        1   073
    TDM(J)#0.0                                                       1   074
112 MOVTYP(J)#1                                                      1   075
    EFSUM#1.0E20                                                     1   076
    N#0                                                              1   077
    ISW1#0                                                           1   078
    ISW3#0                                                           1   0781
    TOLER#10.0                                                       1   0782
    SIDE#1.0                                                         1   0783
    OFFFIX#0.0                                                       1   0784
    OFFVAR#0.0                                                       1   0785
    TOLRAD#0.0                                                       1   0786
    GO TO 101                                                        1   079
```

Table A-1 (Cont'd)

```
   8 EPS#A
     GO TO 101
   9 TOLRAD#A
     TOLFAC#B
     GO TO 101
  10 FRAC#A
     ITER#B+0.5
     GO TO 101
  11 CURVRG#A
     CENTER#B
     ISW1#1
     M(7)#CRVF(0.0)
     GO TO 101
  12 IF(A)113,113,114
 113 A#1.0
 114 MOVE#A+0.01
     GO TO 101
  13 ISW3#1
     OFFFIX#A
     TOLER#B
     IF(C)101,101,1141
1141 SIDE#C
     GO TO 101
  14 ISW3#1
     OFFVAR#A
     GO TO 101
  15 IND#N-1
     ISW2#0
     NTYPEN#XABSF(XMODF(NP,3)-1)
 117 XLSUM#0.0
     DO 119 J#1,IND
     DY1#DY2
     DX1#DX2
     DX2#X(J+1)-X(J)
     DY2#Y(J+1)-Y(J)
     XLSQ(J)#DX2*DX2+DY2*DY2
     XL(J)#SQRTF(XLSQ(J))
     XLSUM#XLSUM+XL(J)
     SS(J)#DY2/SMALF(DX2)
     GA(J)#ATANF(SS(J))
     IF(J-1)119,119,118
 118 TXA(J)#TANMF(SS(J),SS(J-1))
     XA(J)#ATANF(TXA(J))
     S(J)#(XLSQ(J-1)*DY2+XLSQ(J)*DY1)/SMALF(XLSQ(J-1)*DX2+XLSQ(J)*DX1)
     PHI(J)#ATANF(-1.0/SMALF(S(J)))
 119 CONTINUE
     IF(NTYPE1)121,120,121
 120 TA(1)#-TANMF(S(2),SS(1))
     S(1)#TANPF(SS(1),TA(1))
     PHI(1)#ATANF(-1.0/SMALF(S(1)))
 121 IF(NTYPEN)123,122,123
 122 TB(IND)#-TANMF(S(IND),SS(IND))
     S(N)#TANPF(SS(IND),TB(IND))
     PHI(N)#ATANF(-1.0/SMALF(S(N)))
 123 DO 124 J#1,N
     GA(J)#GA(J)*CONV
 124 XA(J)#XA(J)*CONV
     DO 125 J#1,IND
     TA(J)#TANMF(S(J),SS(J))
     TB(J)#TANMF(S(J+1),SS(J))
```

Table A-1 (Cont'd)

```
      CA(J)#CRVAF(TA(J),TB(J),XL(J))
      CB(J)#CRVBF(TA(J),TB(J),XL(J))
  125 DC(J)#CA(J)-CB(J-1)
  126 DCMAXP#0.0
      DCMAX#0.0
      DO 128 J#2,IND
      DCMAX#MAXIF(DCMAX,ABSF(DC(J)))
      IF(DCMAX-DCMAXP)128,128,127
  127 DCMAXP#DCMAX
      L#J
  128 CONTINUE
      IF(DCMAX-EPS)140,140,129
  129 LOOP(L)#LOOP(L)+1
      IF(LOOP(L)-20)130,140,130
  130 LL#L-1
      DO 131 LOOL#1,5
      DCP#-4.0*(XL(LL)+XL(L))/(XL(LL)*XL(L))
      TA(L)#TA(L)-DC(L)/DCP
      TB(LL)#TANPF(TA(L),TXA(L))
      CB(LL)#CRVBF(TA(LL),TB(LL),XL(LL))
      CA(L)#CRVAF(TA(L),TB(L),XL(L))
      DC(L)#CA(L)-CB(LL)
      IF(ABSF(DC(L))-EPS)133,131,131
  131 CONTINUE
  133 IF(LL-1)134,134,136
  134 IF(NTYPE1)136,135,136
  135 TA(LL)#-TB(LL)
      S(LL)#TANPF(SS(LL),TA(LL))
      PHI(LL)#ATANF(-1.0/SMALF(S(LL)))
      CB(LL)#CRVBF(TA(LL),TB(LL),XL(LL))
      DC(L)#CA(L)-CB(LL)
  136 IF(L-IND)139,137,137
  137 IF(NTYPEN)139,138,139
  138 TB(L)#-TA(L)
      S(N)#TANPF(SS(L),TB(L))
      PHI(N)#ATANF(-1.0/SMALF(S(N)))
      CA(L)#CRVAF(TA(L),TB(L),XL(L))
      DC(L)#CA(L)-CB(LL)
  139 CA(LL)#CRVAF(TA(LL),TB(LL),XL(LL))
      CB(L)#CRVBF(TA(L),TB(L),XL(L))
      S(L)#TANPF(SS(L),TA(L))
      PHI(L)#ATANF(-1.0/SMALF(S(L)))
      DC(LL)#CA(LL)-CB(L-2)
      DC(L+1)#CA(L+1)-CB(L)
      GO TO 126
  140 DC(1)#0.0
      DC(N)#0.0
      EP#EFSUM
      EF(N)#0.0
      EFSUM#0.0
      DO 141 J#1,IND
      A2#XL(J)
      C1#3.0*(TA(J)+TB(J))/(A2*A2)
      C2#2.0*(2.0*TA(J)+TB(J))/A2
      C3#TA(J)
      EF(J)#RIEMAN(0.0,A2,5,0.00001,1)
  141 EFSUM#EFSUM+EF(J)
      IF(ISW2-1000)142,146,146
  142 IF((EP-EFSUM)/EP-TOLFAC)143,143,144
  143 ISW2#1000
```

Table A-1 (Cont'd)

```
      GO TO 117
 144  IF(ISW2-ITER)145,146,146
 145  IF(ISW2)149,149,168
 146  ISW2#1000
      DO 148 J#1,N
      IF(MOVTYP(J)-1)148,148,147
 147  R(J)#SQRTF(X(J)*X(J)+Y(J)*Y(J))
      CALL ANGLE (TH(J),Y(J),X(J))
 148  CONTINUE
 149  LINES#0
      DO 152 J#1,N
      IF(LINES)150,150,151
 150  CALL HEAD (1,6,TITLE)
      WRITE OUTPUT TAPE 6,1002
      LINES#53
 151  THD#TH(J)*CONV
      WRITE OUTPUT TAPE6,1003,J,THD,R(J),X(J),Y(J),XL(J),GA(J),XA(J),
     1TDM(J)
 152  LINES#LINES-1
      WRITE OUTPUT TAPE 6,1004,XLSUM
      LINES#0
      DO 157 J#1,N
      IF(LINES)153,153,154
 153  CALL HEAD (1,6,TITLE)
      WRITE OUTPUT TAPE 6,1005
      LINES#53
 154  PHIDEG#PHI(J)*CONV
      AL(J)#PHI(J)-TH(J)+4.0*HALFPI
1541  IF(ABSF(AL(J))-HALFPI)156,156,155
 155  AL(J)#AL(J)-2.0*HALFPI
      GO TO 1541
 156  ALDEG#AL(J)*CONV
      WRITE OUTPUT TAPE 6,1006,J,S(J),PHIDEG,ALDEG,TA(J),TB(J),CA(J),
     1CB(J),DC(J),EF(J)
 157  LINES#LINES-1
      WRITE OUTPUT TAPE 6,1007,EFSUM
      DO 166 J#1,IND
      IF(J-1)158,158,164
 158  CALL HEAD (1,6,TITLE)
      LINES#1
      IF(ISW1)159,159,163
 159  CMIN#1.0E10
      CMAX#-1.0E10
      DO 160 L#1,IND
      CMIN#MIN1F(CMIN,CA(L),CB(L))
 160  CMAX#MAX1F(CMAX,CA(L),CB(L))
      CEN2#(CMAX-CMIN)/2.0
      DO 161 L#1,7
      IF(CEN2-TABLE(L))162,162,161
 161  CONTINUE
      L#8
 162  CURVRG#TABLE(L)
      CEN1#10.0/CURVRG
      CENTER#CEN2+CMIN
      CENTER#INTF(CENTER*CEN1+SIGNF(0.5,CENTER))/CEN1
 163  CURTI1#CENTER-CURVRG
      CURTI2#CENTER-0.5*CURVRG
      CURTI4#CENTER+0.5*CURVRG
      CURTI5#CENTER+CURVRG
      WRITE OUTPUT TAPE 6,1008,CURTI1,CURTI2,CENTER,CURTI4,CURTI5
```

Table A-1 (Cont'd)

```
      WRITE OUTPUT TAPE 6,1009
      M(1)#0
      GO TO 165
  164 M(1)#CRVF(CB(J-1))
  165 M(2)#CRVF(CA(J))
      M(7)#CRVF(0.0)
      CALL GRAPH (M,FNT)
      WRITE OUTPUT TAPE 6,FNT,J,CA(J)
  166 CONTINUE
      M(1)#CRVF(CB(IND))
      M(2)#0
      M(7)#CRVF(0.0)
      CALL GRAPH (M,FNT)
      WRITE OUTPUT TAPE 6,FNT,J,CB(IND)
      WRITE OUTPUT TAPE 6,1009
      DO 167 J#1,N
      RN(J)#R(J)
      XN(J)#X(J)
  167 YN(J)#Y(J)
      IF(ISW2-1000)168,182,182
  168 IF(TOLRAD)182,182,169
  169 X1#X(1)
      Y1#Y(1)
      X2#X(2)
      Y2#Y(2)
      DO 180 L#2,IND
      XMX#X(L+1)-X1
      YMY#Y(L+1)-Y1
      TLSQ#XMX*XMX+YMY*YMY
      TL#SQRTF(TLSQ)
      FAC#YMY/SMALF(XMX)
      A#TANMF(S(L-1),FAC)
      B#TANMF(S(L+1),FAC)
      U1#(TLSQ+XL(L-1)**2-XL(L)**2)/(2.0*TL)
      V2#(XMX*(Y2-Y1)-YMY*(X2-X1))/TL
      V1#(A+B)*(U1**3)/TLSQ-(2.0*A+B)*(U1*U1)/TL+A*U1
      V1#V1+SIGNF(ABSF(V1-V2)-ABSF(V1-V2)*FRAC,V2-V1)
      X(L)#X1+(U1*XMX-V1*YMY)/TL
      Y(L)#Y1+(U1*YMY+V1*XMX)/TL
      PD#V1-V2
      MOVE#MOVTYP(L)
      GO TO (172,175,177,170),MOVE
  170 TDM(L)#SQRTF((X(L)-XN(L))**2+(Y(L)-YN(L))**2)
      IF(TDM(L)-TOLRAD)1711,1711,171
  171 FAC#TOLRAD/TDM(L)
      X(L)#XN(L)+FAC*(X(L)-XN(L))
      Y(L)#YN(L)+FAC*(Y(L)-YN(L))
      TDM(L)#TOLRAD
 1711 TDM(L)#SIGNF(TCM(L),Y(L)-YN(L))
      GO TO 179
  172 PD#SIGNF(PD,SQRTF(X(L)*X(L)+Y(L)*Y(L))-R(L))
      R(L)#R(L)+PD
      TDM(L)#R(L)-RN(L)
      IF(ABSF(TDM(L))-TOLRAD)174,174,173
  173 TDM(L)#SIGNF(TOLRAD,TDM(L))
      R(L)#RN(L)+TDM(L)
  174 X(L)#R(L)*COSF(TH(L))
      Y(L)#R(L)*SINF(TH(L))
      GO TO 179
  175 PD#SIGNF(PD,X(L)-X2)
```

Table A-1 (Cont'd)

```
      X(L)#X2+PD
      Y(L)#Y2
      TDM(L)#X(L)-XN(L)
      IF(ABSF(TDM(L))-TOLRAD)179,179,176
  176 TDM(L)#SIGNF(TOLRAD,TDM(L))
      X(L)#XN(L)+TDM(L)
      GO TO 179
  177 PD#SIGNF(PD,Y(L)-Y2)
      Y(L)#Y2+PD
      X(L)#X2
      TDM(L)#Y(L)-YN(L)
      IF(ABSF(TDM(L))-TOLRAD)179,179,178
  178 TDM(L)#SIGNF(TOLRAD,TDM(L))
      Y(L)#YN(L)+TDM(L)
  179 X1#X2
      Y1#Y2
      X2#X(L+1)
      Y2#Y(L+1)
  180 CONTINUE
      DO 181 L#1,200
  181 LOOP(L)#0
      ISW2#ISW2+1
      GO TO 117
  182 IF(ISW3)101,101,183
  183 CALL TC
      GO TO 101
   16 CALL EXIT
 1000 FORMAT(I3,7X,3(F9.6,1X),5A6)
 1001 FORMAT(1H I3,3F15.6,5X,5A6)
 1002 FORMAT(120H0 NUM          THETA          RADIUS          X-CORD
     1  Y-CORD        SEG LENGTH        SEG ANGLE       EXT ANGLE   MOVED  )
 1003 FORMAT(1H I4,F15.4,4F15.6,2F15.4,F10.6)
 1004 FORMAT(1H0 51X,13HTOTAL LENGTH# F15.6)
 1005 FORMAT(115H0 NUM        SLOPE               NORMAL        ALPHA   TAN ANGLE
     1A   TAN ANGLE B    CURV A        CURV B    DELTA CURV    ENERGY      )
 1006 FORMAT(1H I4,E15.6,2F12.4,2F12.7,4F12.4)
 1007 FORMAT(1H0 88X,13HTOTAL ENERGY# F14.4)
 1008 FORMAT(1H0 16X,F7.4,3(18X,F7.4),14X,F7.4)
 1009 FORMAT(120H     CURVATURE      .+.......................+............
     1..............+.........................+.............................+.)
      END
```

GRAPH

```
      SUBROUTINE GRAPH (M,FNT)
      DIMENSION M(7),FOR(08),FNT(106)
      IF(ISET)1,1,2
    1 ISET#1
B     FNT(1)#743104732611
B     FNT(2)#330473046773
B     FNT(3)#606060606060
B     FOR(1)#013067606060
B     FOR(2)#013054606060
B     FOR(3)#013020606060
B     FOR(4)#013000606060
B     FOR(5)#013053606060
B     FOR(6)#013013606060
B     FOR(7)#013033606060
B     FOR(8)#013060606060
B     FNT(106)#013033346060
      FNT(4)#FOR(7)
```

Table A-1 (Cont'd)

```
    2 DO 3 I#5,105                                                          2   018
    3 FNT(I)#FOR(8)                                                         2   019
      DO 7 I#1,7                                                            2   020
      IF(M(I))4,5,6                                                         2   021
    4 M(I)#5                                                                2   022
      GO TO 7                                                               2   023
    5 M(I)#0                                                                2   024
      GO TO 7                                                               2   025
    6 M(I)#XMINOF(M(I),101)+4                                               2   026
    7 CONTINUE                                                              2   027
      DO 9 I#1,6                                                            2   028
      I1#I+1                                                                2   029
      DO 9 I2#I1,7                                                          2   030
      IF(M(I)-M(I2))9,8,9                                                   2   031
    8 M(I2)#0                                                               2   032
    9 CONTINUE                                                              2   033
      DO 11 I#1,7                                                           2   034
      IF(M(I))11,11,10                                                      2   035
   10 J#M(I)                                                                2   036
      FNT(J)#FOR(I)                                                         2   037
   11 CONTINUE                                                              2   038
      RETURN                                                                2   039
      END                                                                   2   040
```

<u>HEAD</u>

```
      SUBROUTINE HEAD(IFLAG,NOTAPE,TITLE)                                   3   001
      DIMENSION D(2),TITLE(5)                                               3   002
      IF(IFLAG)2,1,3                                                        3   003
    1 IPAGE#0                                                               3   004
C    TAD IS A ROUTINE BASED ON THE OAK RIDGE COMPUTOR CLOCK. T CONTAINS THE TIME
C       IN THE FORM- 23174. THE FIVE CARDS FOLLOWING WOULD CHANGE THIS TO THE
C       FORM- 23 HRS 17 MIN 40 SEC . D CONTAINS THE DATE IN THE FORM- 08/16/62.
C       THE APPROPRIATE SUBROUTINE SHOULD BE SUPPLIED TO OBTAIN THE DATE AND
C       TIME. HOWEVER, IT WILL RUN WITHOUT THE CLOCK ROUTINE. IF TAD IS ADDED,
C       REMOVE C FROM COLUMN ONE OF CARD 5 AND REMOVE CARD AFTER 5.
C   2 CALL TAD(T,D)                                                         3   005
    2 GO TO 3
B     T1#(T*7777CCC00000)+60305162                                          3   006
B     T2#(T*7777C0)+6060600C00060                                           3   007
B     T3#4431456C6060                                                       3   008
B     T4#(T*77)+6C6060606CC0                                                3   009
B     T5#6062252360                                                         3   010
    3 IPAGE#IPAGE+1                                                         3   011
      WRITE OUTPUT TAPE NOTAPE,4,(TITLE(I),I#1,5),(D(I),I#1,2),T1,T2,T3,    3   012
     1T4,T5,IPAGE                                                           3   013
    4 FORMAT(1H15A6,10X,5HDATE-A8,A2,10X,7HTIME-  2A6,A3,2A6,10X,           3   014
     15HPAGE-I5)                                                            3   015
      RETURN                                                                3   016
      END                                                                   3   017
```

<u>VALUE</u>

```
      FUNCTION VALUE(X,II)                                                  4   001
      COMMON C1,C2,C3                                                       4   002
      VALUE#SQRTF(1.0+(C1*X*X-C2*X+C3)**2)                                  4   003
      VALUE#(2.0*C1*X-C2)**2/(VALUE**5)                                     4   004
      RETURN                                                                4   005
      END                                                                   4   006
```

Table A-1 (Cont'd)

RIEMAN

```
      FUNCTION RIEMAN (A1, A2, N, EPS, II)                               6  002
      DIMENSION A(3), F(3)                                               6  003
  100 DIVA#(A2 - A1)/FLOATF(N)                                           6  004
  101 DO 103 I # 1,3                                                     6  005
  102 A(I) # A1 + (FLOATF(2*I-1)/2.0)*DIVA                               6  006
  103 F(I) # VALUE (A(I),II)                                             6  007
  104 END # A1                                                           6  008
  105 RIEMAN # 0.0                                                       6  009
  199 DO 233 I # 1,N                                                     6  010
 1991 RESULT # 0.0                                                       6  011
  200 ERROR # (((F(3) - 2.0*F(2) + F(1))*DIVA)/24.0)*FLOATF(N)           6  012
  201 ERR # ABSF(ERROR/EPS)                                              6  013
  202 IF(ERR - 1.0) 218,218,203                                          6  014
  203 K # SQRTF(ERR) + 1.0                                               6  015
      IF(K-32765)204,204,1                                               6  016
    1 IF(N-16382)2,2,3                                                   6  017
    2 N#N*2                                                              6  018
      GO TO 100                                                          6  019
    3 CALL PDUMP (RIEMAN,A(1),0)                                         6  020
  204 L # K/2                                                            6  021
  205 K # 2*L+1                                                          6  022
  206 DIVR # DIVA/FLOATF(K)                                              6  023
  207 DIVR2 # DIVR/2.0                                                   6  024
  208 DO 211 J # 1,L                                                     6  025
  209 ADD # VALUE (END + DIVR2, II)*DIVR                                 6  026
  210 RESULT # RESULT + ADD                                              6  027
  211 END # END + DIVR                                                   6  028
  212 END # END + DIVR                                                   6  029
  213 DO 216 J # 1,L                                                     6  030
  214 ADD # VALUE (END + DIVR2, II)*DIVR                                 6  031
  215 RESULT # RESULT + ADD                                              6  032
  216 END # END + DIVR                                                   6  033
  217 GO TO 220                                                          6  034
  218 END # END + DIVA                                                   6  035
  219 DIVR # DIVA                                                        6  036
  220 IF(I-1) 222,222,221                                                6  037
  221 IF (I - (N-1)) 224,229,229                                         6  038
  222 RESULT # RESULT + F(1)*DIVR                                        6  039
  223 GO TO 233                                                          6  040
  224 RESULT # RESULT + F(2)*DIVR                                        6  041
  225 F(1) # F(2)                                                        6  042
  226 F(2) # F(3)                                                        6  043
  227 F(3) # VALUE(END+(3./2.)*DIVA,II)                                  6  044
  228 GO TO 233                                                          6  045
  229 IF (I-N) 230,232,232                                               6  046
  230 RESULT # RESULT + F(2)*DIVR                                        6  047
  231 GO TO 233                                                          6  048
  232 RESULT # RESULT + F(3)*DIVR                                        6  049
  233 RIEMAN # RIEMAN + RESULT                                           6  050
  234 RETURN                                                             6  051
      END                                                                6  052
```

ANGLE

```
      SUBROUTINE ANGLE (A,TOP,BOT)                                       7  001
      IF(BOT)5,1,6                                                       7  002
    1 IF(TOP)2,3,4                                                       7  003
    2 A#4.71238898                                                       7  004
      RETURN                                                             7  005
    3 A#0.0                                                              7  006
      RETURN                                                             7  007
    4 A#1.57079633                                                       7  008
```

Table A-1 (Cont'd)

```
      RETURN
    5 FAC#3.14159265
      GO TO 9
    6 IF(TOP)7,8,8
    7 FAC#6.28318531
      GO TO 9
    8 FAC#0.0
    9 A#ATANF(TOP/BOT)+FAC
      RETURN
      END
```

TC

```
      SUBROUTINE TC
      DIMENSION X(200),Y(200),XL(200),PHI(200),CA(200),CB(200),TA(200),
     1TB(200),OFFSET(200),TITLE(5)
      COMMON C1,C2,C3,TA,TB,XL,X,Y,PHI,CA,CB,OFFSET,N,TOLER,SIDE,TITLE
      LINES#0
      IND#N-1
      ISW1#1
      DO 1 I#1,N
      IF(OFFSET(I))2,1,2
    1 CONTINUE
      ISW1#0
    2 DO 23 L#1,N
      IF(L-N)4,10,10
    4 J#0
      FK#INTF(XL(L)/SQRTF(8.0*TOLER/MAXIF(ABSF(CA(L)),ABSF(CB(L)),
     11.0E-6)))
      IF(L-1)5,5,9
    5 QL1#1.0
      IF(ABSF(COSF(PHI(1)))-.707)6,7,7
    6 FAC#SIGNF(1.0,X(2)-X(1))
      U11#FAC*ABSF(SINF(PHI(1)))
      V11#-FAC*SIGNF(COSF(PHI(1)),PHI(1))
      GO TO 8
    7 FAC#SIGNF(1.0,Y(2)-Y(1))
      U11#-FAC*SINF(PHI(1))
      V11#FAC*COSF(PHI(1))
    8 XQ2#X(1)
      YQ2#Y(1)
    9 J#J+1
      FJ#J
      RJ#OFFSET(L)+((FJ-1.0)/(FK+1.0))*(OFFSET(L+1)-OFFSET(L))
      FAC#((FK+1.0-FJ)/(FK+1.0))*(FJ/(FK+1.0))*((TA(L)*(FK+1.0-FJ)/
     1(FK+1.0))-(TB(L)*FJ/(FK+1.0)))
      XQ3#X(L)+((X(L+1)-X(L))*FJ/(FK+1.0))-((Y(L+1)-Y(L))*FAC)
      YQ3#Y(L)+((Y(L+1)-Y(L))*FJ/(FK+1.0))+((X(L+1)-X(L))*FAC)
      U12#XQ3-XQ2
      V12#YQ3-YQ2
   10 QL2#SQRTF(U12*U12+V12*V12)
      DYP#RJ*(U12/QL2-U11/QL1)
      DXP#RJ*(V12/QL2-V11/QL1)
      DS2#DYP*DYP+DXP*DXP
      GYP#0.5*RJ*(U12/QL2+U11/QL1)
      GXP#0.5*RJ*(V12/QL2+V11/QL1)
      FAC#SQRTF(DS2)*0.25/ SQRTF(RJ*RJ -0.25*DS2)
      HYP#SIGNF(FAC*DXP,GYP)
      HXP#SIGNF(FAC*DYP,GXP)
      XTC#XQ2-(GXP+HXP)*SIDE
      YTC#YQ2+(GYP+HYP)*SIDE
      U11#U12
```

Table A-1 (Cont'd)

```
      V11#V12
      QL1#QL2
      XQ1#XQ2
      YQ1#YQ2
      XQ2#XQ3
      YQ2#YQ3
      IF(LINES)15,11,15
   11 CALL HEAD (1,6,TITLE)
      IF(ISW1)12,12,13
   12 WRITE OUTPUT TAPE 6,1005
      GO TO 14
   13 WRITE OUTPUT TAPE 6,1007
   14 LINES#49
   15 IF(ISW1)17,17,16
   16 WRITE OUTPUT TAPE 6,1006,L,J,XQ1,YQ1,XTC,YTC
      GO TO 18
   17 WRITE OUTPUT TAPE 6,1006,L,J,XQ1,YQ1
   18 LINES#LINES-1
      IF(J-XFIXF(FK))19,9,19
   19 IF(L-IND)23,20,23
   20 FK#0.0
      J#1
      IF(ABSF(COSF(PHI(L+1)))-.707)21,22,22
   21 FAC#SIGNF(1.0,X(L+1)-X(L))
      U12#FAC*ABSF(SINF(PHI(L+1)))
      V12#-FAC*SIGNF(COSF(PHI(L+1)),PHI(L+1))
      GO TO 23
   22 FAC#SIGNF(1.0,Y(L+1)-Y(L))
      U12#-FAC*SINF(PHI(L+1))
      V12#FAC*COSF(PHI(L+1))
   23 CONTINUE
      RETURN
 1005 FORMAT(26HOSEG   PT       CURVE POINTS / 13X,1HX 11X,1HY )
 1006 FORMAT(2I4,2F12.6,4X,2F12.6)
 1007 FORMAT(26HOSEG   PT       CURVE POINTS 16X 13HOFFSET POINTS / 13X
     11HX 11X,1HY 15X,1HX 11X,1HY )
      END
```

DICTIONARY OF VARIABLES

Variable	Definition	Variable	Definition
A	First field of data on input cards	M(N)	Table of values of curvature to be plotted on one line
A2	Upper limit of integral used in energy calculation	MOVE	Flag used to designate type of movement to be made at next point
AL(N)	Table of alpha angles (normal - theta)	MOVTYP(N)	Table designating type of movement for each point
ALDEG	Alpha angle in degrees	N	Number of points
B	Second field of data on input cards	NN	Indicates type of data on an input card
C	Third field of data on input cards	NP	Temporary storage used during input
CA(N)	Table of curvature of the normalized cubic at beginning of each segment	NTYPE1	Indicates if slope at first point was specified by input
CB(N)	Table of curvature of the normalized cubic at the end of each segment	NTYPEN	Indicates if slope at last point was specified by input
CEN1	Parameter calculated and used in the curvature plot section	OFFFIX	Fixed portion of the offset
CEN2	Parameter calculated and used in the curvature plot section	OFFSET(N)	Table of offset distances
CENTER	Value of curvature at the center of the curvature plot	OFFVAR	Variable portion of the offset
CMAX	Maximum value of curvature to be plotted	PD	Distance a point is to be moved
CMIN	Minimum value of curvature to be plotted	PHI(N)	Table of normals at each point (radians)
CONV	Conversion factor to change the units of expression for angles	PHIDEG	Normal at a point (degrees)
CURTI1	Value of curvature at the left edge of the plot	R(N)	Table of radius coordinates
CURTI2	Value of curvature at the center of the left side of the plot	RN(N)	Table of original radius coordinates
CURTI4	Value of curvature at the center of the right side of the plot	S(N)	Table of slopes at each point
CURTI5	Value of curvature at the right edge of the plot	SIDE	Direction of offset
CURVRG	Difference in curvature between the center and the edge of the plot	SS(N)	Table of segment slopes
D(N)	Table containing the date on which the computer run is made	T(N)	Table of the descriptive information in the last data card
DC(N)	Table of the difference in curvature at each point	TA(N)	Table of tangents of the angle between the segment and the curve slope at the first point in each segment
DCMAX	Maximum difference in curvature		
DCMAXP	Parameter used in finding the largest difference in curvature	TABLE(N)	Table of headings for curvature plot
DCP	Approximation of the derivative of the difference in curvature with respect to the tangent of the slope angle	TB(N)	Table of tangents of the angle between the segment and the curve slope at the last point in each segment
DX1	Delta X between the first two of three successive points	TDM(N)	Table of distance each point has been moved
DX2	Delta X between the second two of three successive points	TH(N)	Table of theta coordinates
DY1	Delta Y between the first two of three successive points	THD	Theta coordinate in degrees
DY2	Delta Y between the second two of three successive points	TITLE(N)	Table of descriptive information from the type seven card
EF(N)	Table of the strain energy for each normalized cubic	TL	Segment length from a given point to the second point following
EFSUM	Strain energy of a cubic spline	TLSQ	Square of TL
EP	Strain energy of the previous cubic spline	TOLER	Interpolation scallop tolerance
EPS	Maximum allowable difference in curvature	TOLFAC	Minimum fractional decrease in strain energy allowed
FAC	Temporary storage	TOLRAD	Maximum distance a point may be moved
FNT(N)	Format for one line of the plot, produced by Subroutine GRAPH	TXA(N)	Table of tangents of the angles between adjacent segments
FRAC	Fraction of the predicated distance a point is moved	U1	Coordinate used in point movement
GA(N)	Table of the slope of each segment (degrees)	V1	Coordinate used in point movement
HALFPI	Constant equal $\pi/2$	V2	Variable used in point movement
I	Index	X(N)	Table of X coordinates
IND	Number of segments	X1	Coordinate of the first point in a series
ISW1	Indicates if center and range of curvature plot were input data	X2	Coordinate of the second point in a series
ISW2	Indicates number of passes made at adjusting points	XA(N)	Table of angles between adjacent segments
ISW3	Indicates if interpolation and offseting are desired	XL(N)	Table of segment lengths
ITER	Maximum number of passes allowed to adjust points	XLSQ(N)	Table of squares of XL
J	Index	XLSUM	Sum of the segment lengths
L	Index	XMX	Delta X between two given points
LINES	Number of lines remaining to be printed on a given output page	XN(N)	Table of original X coordinates
LL	Index	Y(N)	Table of Y coordinates
LOOL	Number of attempts made to match the curvature of a given point on a given pass	Y1	Coordinate of the first point in a series
		Y2	Coordinate of the second point in a series
LOOP(N)	Table of number of passes made at each point to match curvature	YMY	Delta Y between two given points
		YN(N)	Table of original Y coordinates

APPENDIX B

TEST CURVES

SLOPE AND CURVATURE COMPARISON (Example 1)

In order to evaluate the difference between the slopes obtained by the normalized cubic with continuous curvature method and the slopes at points on a known curve, a test case consisting of 28 points from the sine curve, Y = 2 sine X, was examined (see Figure B-1). For purposes of comparison, the slopes, curvatures, and other parameters were calculated in the following ways:

Method 1 - Directly from the equation of the sine curve.

Method 2 - By the cubic spline program with slopes given at the first and last points.

Method 3 - By the cubic spline program with no slopes given.

The coordinates of the points used and the related point parameters are shown in Table B-1. The curve parameters obtained by the three methods are shown in Tables B-2, B-3, and B-4. The curvature plot obtained by the third method is shown in Figure B-2. The differences between the slopes found by the first method and those found by the other methods are plotted in Figure B-3. Similarly, the curvature differences are plotted in Figure B-4.

The sine curve was selected as a test curve because it has several features which normally give trouble in curve fitting routines. First, the points are not evenly spaced. As may be seen in Table B-1, the segment lengths vary from 0.177 to 0.389. Second, the radius of curvature spans a large range of values; namely, from 0.5 to infinity. Third, the curvature changes significantly between points. There is as much as a 17 percent change in some segments. Fourth, the curve contains an inflection.

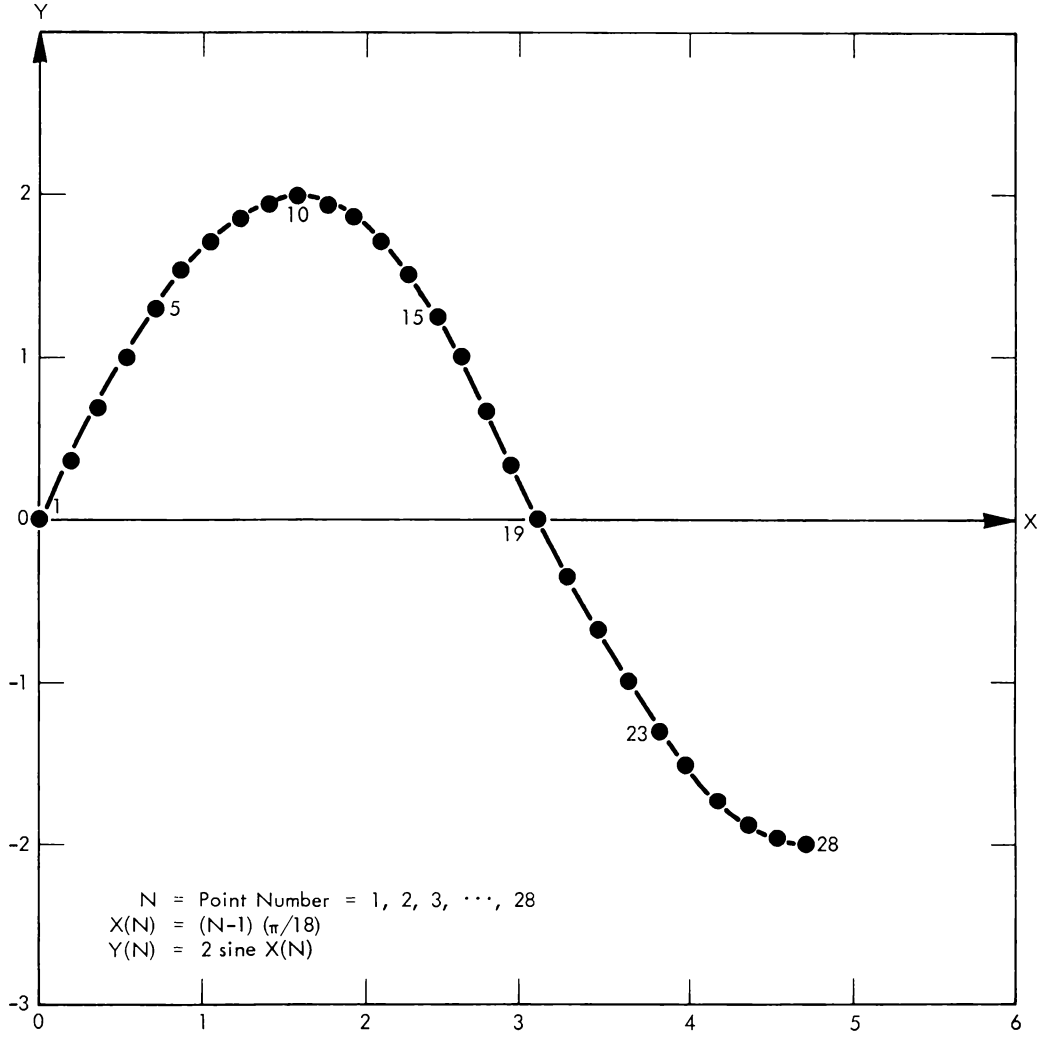

Figure B-1. 28 POINTS FROM SINE CURVE.

Table B-1

POINT PARAMETERS

SINE CURVE- METHOD 2 DATE- 08/26/62 TIME- 08 HRS 55 MIN 30 SEC PAGE- 2

NUM	THETA	RADIUS	X-CORD	Y-CORD	SEG LENGTH	SEG ANGLE	EXT ANGLE	MOVED
1	0.	0.	0.	0.	0.388689	63.3185	0.	0.
2	63.3185	0.388689	0.174533	0.347300	0.379283	62.6022	-0.7163	0.
3	62.9648	0.767957	0.349066	0.684040	0.360960	61.0842	-1.5180	0.
4	62.3635	1.128785	0.523599	1.000000	0.334690	58.5687	-2.5155	0.
5	61.4959	1.462909	0.698132	1.285580	0.302033	54.6998	-3.8689	0.
6	60.3344	1.763182	0.872664	1.532080	0.265431	48.8871	-5.8127	0.
7	58.8430	2.024019	1.047197	1.732060	0.228396	40.1671	-8.7200	0.
8	56.9733	2.241583	1.221730	1.879380	0.196482	27.3406	-12.8264	0.
9	54.6672	2.414323	1.396263	1.969620	0.177157	9.8742	-17.4664	0.
10	51.8540	2.543108	1.570796	2.000000	0.177157	-9.8742	-19.7485	0.
11	48.4551	2.631649	1.745329	1.969620	0.196482	-27.3406	-17.4664	0.
12	44.3895	2.686622	1.919862	1.879380	0.228396	-40.1671	-12.8265	0.
13	39.5907	2.717816	2.094395	1.732060	0.265431	-48.8871	-8.7200	0.
14	34.0290	2.737755	2.268928	1.532080	0.302033	-54.6998	-5.8127	0.
15	27.7502	2.761017	2.443461	1.285580	0.334690	-58.5687	-3.8689	0.
16	20.9055	2.802479	2.617993	1.000000	0.360960	-61.0842	-2.5155	0.
17	13.7638	2.875085	2.792526	0.684040	0.379283	-62.6022	-1.5180	0.
18	6.6762	2.987316	2.967059	0.347300	0.388689	-63.3185	-0.7163	0.
19	0.	3.141593	3.141593	0.	0.388689	-63.3185	-0.0001	0.
20	354.0212	3.334262	3.316126	-0.347300	0.379283	-62.6022	0.7163	0.
21	348.9126	3.557050	3.490658	-0.684040	0.360960	-61.0842	1.5180	0.
22	344.7390	3.799161	3.665191	-1.000000	0.334690	-58.5687	2.5155	0.
23	341.4889	4.049222	3.839724	-1.285580	0.302033	-54.6998	3.8689	0.
24	339.1102	4.296688	4.014257	-1.532080	0.265431	-48.8871	5.8127	0.
25	337.5350	4.532769	4.188790	-1.732060	0.228396	-40.1671	8.7200	0.
26	336.6974	4.750858	4.363323	-1.879380	0.196482	-27.3406	12.8264	0.
27	336.5371	4.946872	4.537856	-1.969620	0.177157	-9.8742	17.4664	0.
28	337.0030	5.119239	4.712389	-2.000000	0.	0.	0.	0.

TOTAL LENGTH# 7.899366

Table B-2

CURVE PARAMETERS - METHOD 1

Num	Slope	Normal	Alpha	Tan Angle A	Tan Angle B	Curv A	Curv B	Delta Curv
1		-26.56505				0.00000		
2		-26.91751				-0.03222		
3		-28.01695				-0.07090		
4		-30.00000				-0.12500		
5		-33.13263				-0.20992		
6		-37.87799				-0.35461		
7		-45.00000				-0.61237		
8		-55.62630				-1.05674		
9		-70.84807				-1.66034		
10		90.00000				-2.00000		
11		70.84807				-1.66034		
12		55.62630				-1.05674		
13		45.00000				-0.61237		
14		37.87799				-0.35461		
15		33.13263				-0.20992		
16		30.00000				-0.12500		
17		28.01695				-0.07090		
18		26.91751				-0.03222		
19		26.56505				0.00000		
20		26.91751				0.03222		
21		28.01695				0.07090		
22		30.00000				0.12500		
23		33.13263				0.20992		
24		37.87799				0.35461		
25		45.00000				0.61237		
26		55.62630				1.05674		
27		70.84807				1.66034		
28		90.00000				2.00000		

Table B-3

CURVE PARAMETERS
(Sine Curve - Method 2)

SINE CURVE- METHOD 2 DATE- 08/26/62 TIME- 08 HRS 55 MIN 30 SEC PAGE- 3

NUM	SLOPE	NORMAL	ALPHA	TAN ANGLE A	TAN ANGLE B	CURV A	CURV B	DELTA CURV	ENERGY
1	0.2000000E 01	-26.5651	-26.5651	0.0020317	-0.0041206	0.0003	-0.0320	0.	0.0001
2	0.196961E 01	-26.9175	89.7639	0.0083818	-0.0107739	-0.0316	-0.0694	0.0004	0.0010
3	0.187953E 01	-28.0151	89.0202	0.0157220	-0.0189114	-0.0694	-0.1224	-0.0000	0.0034
4	0.173211E 01	-29.9992	87.6373	0.0249999	-0.0296377	-0.1216	-0.2045	0.0008	0.0091
5	0.153231E 01	-33.1289	85.3752	0.0379141	-0.0448701	-0.2046	-0.3421	-0.0000	0.0231
6	0.128598E 01	-37.8693	81.7963	0.0566714	-0.0677822	-0.3416	-0.5904	0.0005	0.0595
7	0.100033E 01	-44.9906	76.1664	0.0847162	-0.1013267	-0.5900	-1.0170	0.0004	0.1540
8	0.684234E 00	-55.6188	67.4080	0.1235040	-0.1447909	-1.0171	-1.6387	-0.0001	0.3675
9	0.346320E-00	-70.8980	54.4348	0.1624618	-0.1740647	-1.6378	-2.0043	0.0009	0.6267
10	0.146113E-07	-90.0000	38.1460	0.1740647	-0.1624633	-2.0043	-1.6379	0.0000	0.6267
11	-0.346322E-00	70.8979	22.4429	0.1447894	-0.1235031	-1.6387	-1.0171	-0.0008	0.3675
12	-0.684232E 00	55.6188	11.2293	0.1013276	-0.0847166	-1.0171	-0.5900	0.0001	0.1540
13	-0.100033E 01	44.9906	5.3999	0.0677816	-0.0567134	-0.5901	-0.3423	-0.0000	0.0596
14	-0.128609E 01	37.8669	3.8380	0.0448283	-0.0378810	-0.3418	-0.2044	0.0005	0.0231
15	-0.153220E 01	33.1308	5.3806	0.0296707	-0.0250905	-0.2044	-0.1224	-0.0000	0.0091
16	-0.173247E 01	29.9940	9.0886	0.0188207	-0.0156811	-0.1216	-0.0695	0.0008	0.0034
17	-0.187935E 01	28.0174	14.2536	0.0108149	-0.0084539	-0.0695	-0.0321	-0.0000	0.0010
18	-0.196996E 01	26.9134	20.2372	0.0040473	-0.0020243	-0.0312	-0.0000	0.0009	0.0001
19	-0.199996E 01	26.5655	26.5655	-0.0020231	0.0040476	-0.0000	0.0312	-0.0000	0.0001
20	-0.196997E 01	26.9134	32.8922	-0.0084549	0.0108151	0.0321	0.0695	0.0009	0.0010
21	-0.187935E 01	28.0174	39.1048	-0.0156808	0.0188205	0.0695	0.1216	-0.0000	0.0034
22	-0.173247E 01	29.9940	45.2550	-0.0250908	0.0296709	0.1224	0.2044	0.0008	0.0091
23	-0.153220E 01	33.1308	51.6419	-0.0378809	0.0448281	0.2044	0.3418	-0.0000	0.0231
24	-0.128609E 01	37.8667	58.7568	-0.0567134	0.0677821	0.3423	0.5901	0.0005	0.0596
25	-0.100033E 01	44.9906	67.4556	-0.0847162	0.1013267	0.5900	1.0170	-0.0000	0.1540
26	-0.684234E 00	55.6187	78.9214	-0.1235040	0.1447909	1.0171	1.6387	0.0001	0.3675
27	-0.346320E-00	70.8980	-85.6391	-0.1624618	0.1740647	1.6378	2.0043	-0.0009	0.6267
28	0.	-90.0000	-67.0030	0.	0.	0.	0.	0.	0.

TOTAL ENERGY# 3.7336

Table B-4

CURVE PARAMETERS
(Sine Curve - Method 2)

SIΛE CLRVE- ΜETHCC 3 CATE- 08/26/62 TIΜE- 08 HRS 55 ΜIΛ 4C SEC PAGE- 3

NLM	SLCPE	ΛCRMAL	ALPHA	TAN ANGLE A	TAN ANGLE B	CLRV A	CLRV B	DELTA CLRV	EΛERGY
1	C.2C1444E C1	-26.4CC5	-26.4CC5	C.C049C28	-C.CC49C28	-0.C252	-C.0252	C.	C.CCC2
2	C.19658CE 01	-26.9624	89.7191	0.0075996	-C.C1C5711	-C.C244	-C.0714	C.CCC8	C.CCC9
3	C.188C45E C1	-28.CC34	89.C318	0.0159248	-C.C189581	-C.0714	-C.1218	-C.CCCC	C.CC34
4	C.173192E C1	-3C.0C19	87.6346	0.C249531	-C.C296266	-0.1211	-C.2C47	C.CCC7	C.CC91
5	C.153234E C1	-33.1283	85.3758	0.0379251	-C.0448701	-C.2C47	-C.3421	-C.CCCC	C.C231
6	C.128598E 01	-37.8693	81.7963	0.C566714	-C.C677822	-C.3416	-C.5904	C.CC04	C.C595
7	C.1CCC33E C1	-44.99C6	76.1664	C.0847162	-C.1013267	-C.59CC	-1.0170	C.CCC4	C.154C
8	C.684234E CC	-55.6188	67.4C8C	0.1235C4C	-C.1447909	-1.0171	-1.6387	-C.CCC1	C.3675
9	C.346322E-CC	-7C.898C	54.4348	C.1624618	-C.1740647	-1.6378	-2.CC43	C.CCC9	C.6267
1C	C.146113E-C7	-9C.0CCC	38.1460	0.1740647	-C.1624633	-2.CC43	-1.6379	C.CCCC	C.6267
11	-C.346322E-CC	7C.8979	22.4429	0.1447894	-C.1235C31	-1.6387	-1.0171	-C.CCC8	C.3675
12	-C.684232E CC	55.6188	11.2293	0.1013276	-C.C847166	-1.0171	-C.59CC	C.0CC1	C.1540
13	-C.1CCC33E C1	44.99C6	5.3999	0.C677816	-C.C567134	-C.59C1	-C.3423	-C.CCCC	C.C596
14	-C.128609E C1	37.8669	3.838C	0.0448283	-C.C37881C	-C.3418	-C.2C44	C.CCC5	C.C231
15	-C.153322E 01	33.1308	5.38C6	0.02967C7	-C.C250905	-0.2C44	-C.1224	-C.CCCC	C.CC91
16	-C.173247E C1	29.994C	9.C886	0.01882C7	-0.0156811	-C.1216	-C.C695	C.CCC8	C.CC34
17	-C.187935E C1	28.0174	14.2536	0.01C8149	-C.0084539	-C.C695	-C.C321	-C.CCCC	C.CC10
18	-C.196996E C1	26.9134	2C.2372	0.0040473	-C.CC20243	-C.C312	-C.CCCC	C.CCC9	C.CC01
19	-C.199996E C1	26.5655	26.5655	-0.0020231	C.C040476	-0.CCCC	C.0312	-C.CCCC	C.CC01
20	-C.196997E C1	26.9134	32.8922	-0.CC84549	C.C1C8151	C.0321	C.C695	C.CCC9	C.CC10
21	-C.187935E C1	28.0174	39.1C48	-0.0156808	C.C1882C5	0.C695	C.1216	-C.CCCC	C.CC34
22	-C.173247E C1	29.994C	45.2550	-C.0250908	C.C296964	C.1223	C.2C47	C.CCC7	C.CC91
23	-C.153211E C1	33.1323	51.6434	-0.C378554	C.C447309	0.2C47	C.3407	-C.CCCC	C.C230
24	-C.128634E 01	37.8614	58.7512	-0.05681C7	C.0681854	C.34C7	C.5953	C.CCCC	C.C601
25	-C.999526E CC	45.0136	67.4786	-0.0843119	C.C999105	0.5953	C.9965	C.CCCC	C.1507
26	-C.686294E CC	55.5384	78.841C	-0.1249276	C.1496C64	C.997C	1.7161	C.CCC5	C.3865
27	-C.341C5CE-C0	71.1681	-85.3691	-0.1576277	C.1576277	1.7152	1.7152	-C.CCC9	C.5497
28	-C.159980E-01	89.0835	-67.9195	0.	C.	0.	C.	C.	C.

TCTAL ENERGY# 3.6728

SINE CLRVE- ΜΕΤΗCD 3 CATE- 08/26/62 TIME- 08 HRS 55 MIN 40 SEC PAGE- 4

	CURVATURE
1	-C.C252
2	-C.C244
3	-C.C714
4	-C.1211
5	-C.2047
6	-C.3416
7	-C.59CC
8	-1.C171
9	-1.6378
1C	-2.C043
11	-1.6387
12	-1.C171
13	-C.59C1
14	-C.3418
15	-C.2044
16	-C.1216
17	-C.C695
18	-C.C312
19	-C.CDCC
2C	C.C321
21	C.C695
22	C.1223
23	C.2047
24	C.34C7
25	C.5953
26	C.997C
27	1.7152
28	1.7152

CURVATURE

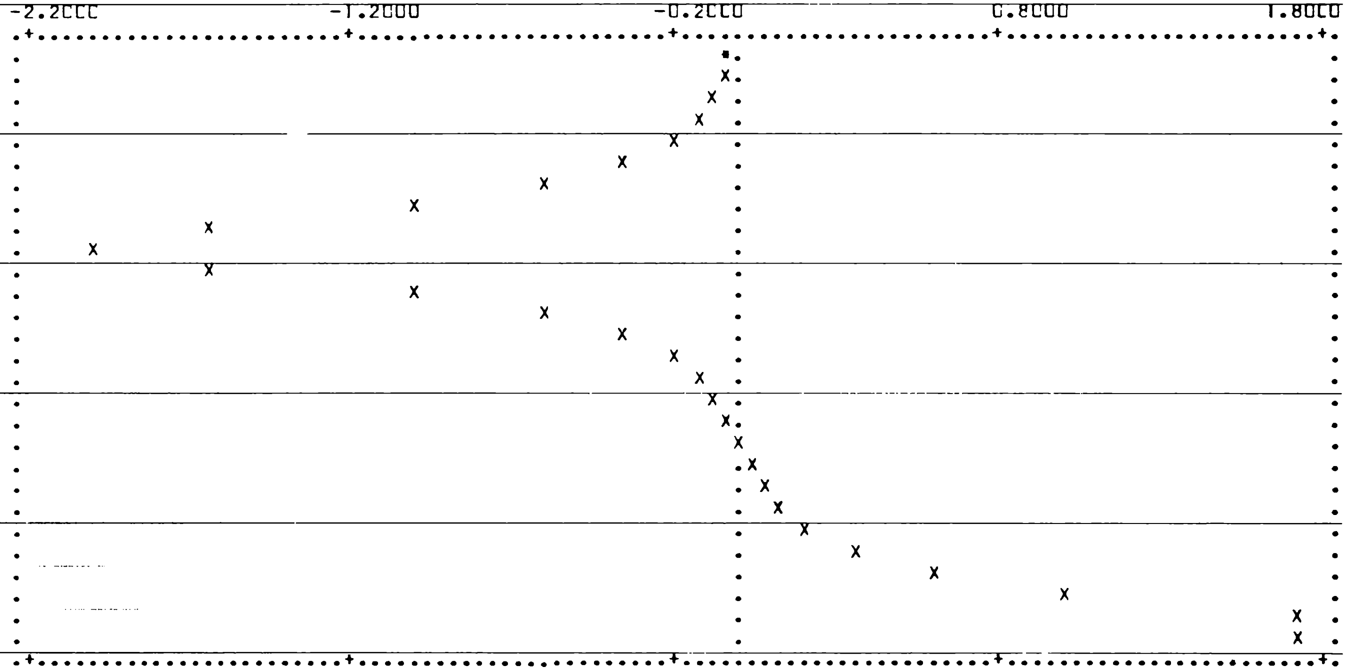

Figure B-2. CURVATURE PLOT. (Sine Curve – Method 3)

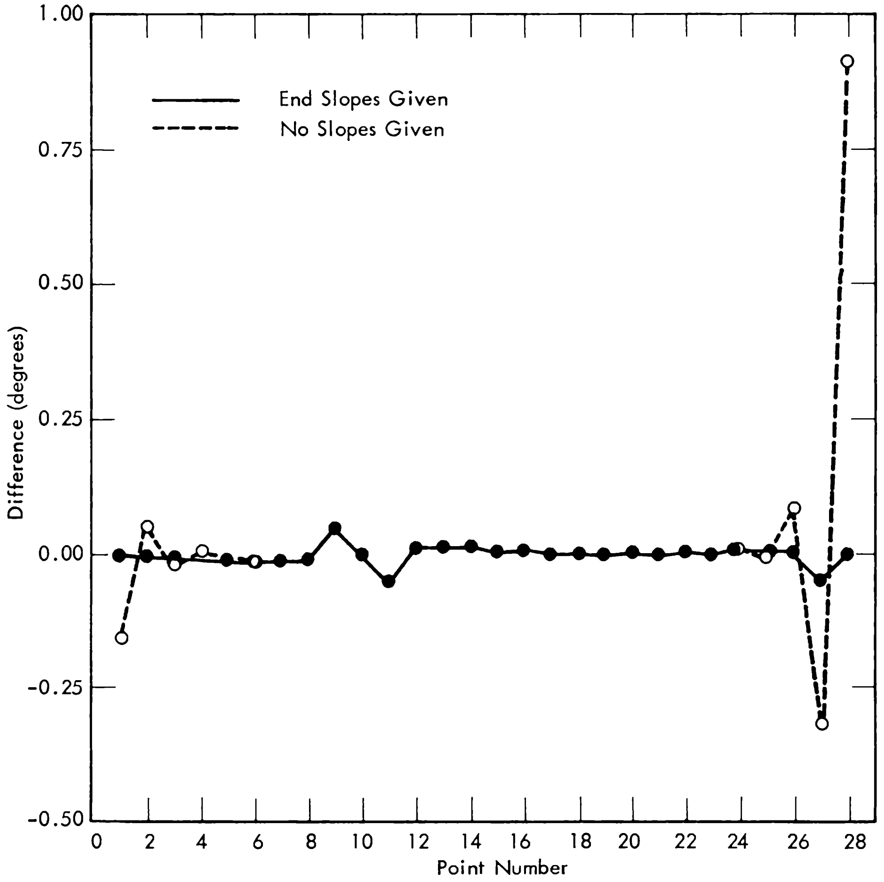

Figure B-3. DIFFERENCE BETWEEN FITTED SLOPES AND ACTUAL SINE CURVE SLOPES.

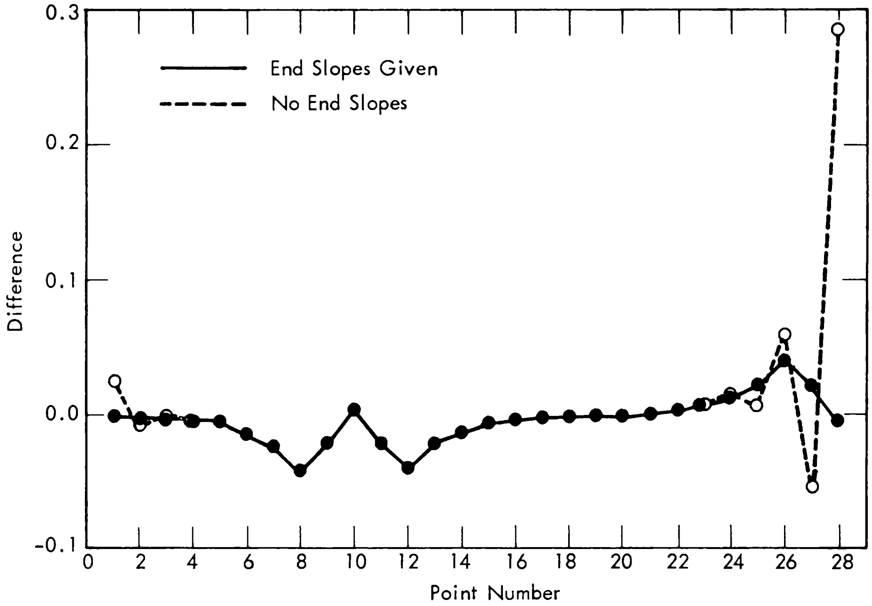

Figure B-4. DIFFERENCE BETWEEN CURVATURE OF FITTED CURVE AND CURVATURE OF ACTUAL SINE CURVE.

POINT ADJUSTMENT (Example 2)

To evaluate the effectiveness of the point adjustment routine, a test case was devised that was based on a circle. Forty-six points were selected from a circle of 5.0045 radius as theta varied from 0 to 90 degrees. A table of random digits was then used to select the thousandths place of the input data; that is, the radius of the points being fitted varied from 5.0000 to 5.0090 (see Figure B-5). The first and last points were left on the original circle. The contour was then run on the cubic spline program with a maximum allowable movement of 0.0045.

The coordinates of the data points and the related point parameters are shown in Table B-5. The curve parameters and the curvature plot for the given data are shown in Table B-6 and Figure B-6, respectively. The same three kinds of information about the curve after 25 iterations in the point moving routine are shown in Tables B-7 and B-8 and Figure B-7. Figure B-5 and Table B-7 show that the curvature plot is considerably smoother after the points have been adjusted and that no point was moved as much as allowed. Figure B-8 is a plot of the strain energy of the spline after each iteration. The energy of the original circle is

$$\int_0^s K^2 \, ds = \int_0^s \left(\frac{1}{5.0045}\right)^2 ds = 0.31387677 .$$

Notice how rapidly the routine knocks out the high-frequency curvature kinks to reduce the energy. After only ten passes the strain energy has already been reduced from 1.2396 to 0.3238.

Since the point-defined curve was derived by applying a random fluctuation to a circular arc, it contains a wide range of frequency components. This is evident in Figure B-6. Only the high-frequency components are removed by the smoothing routine and therefore the energy will approach a value equal to the energy of the circle plus the energy contained in the low-frequency components.

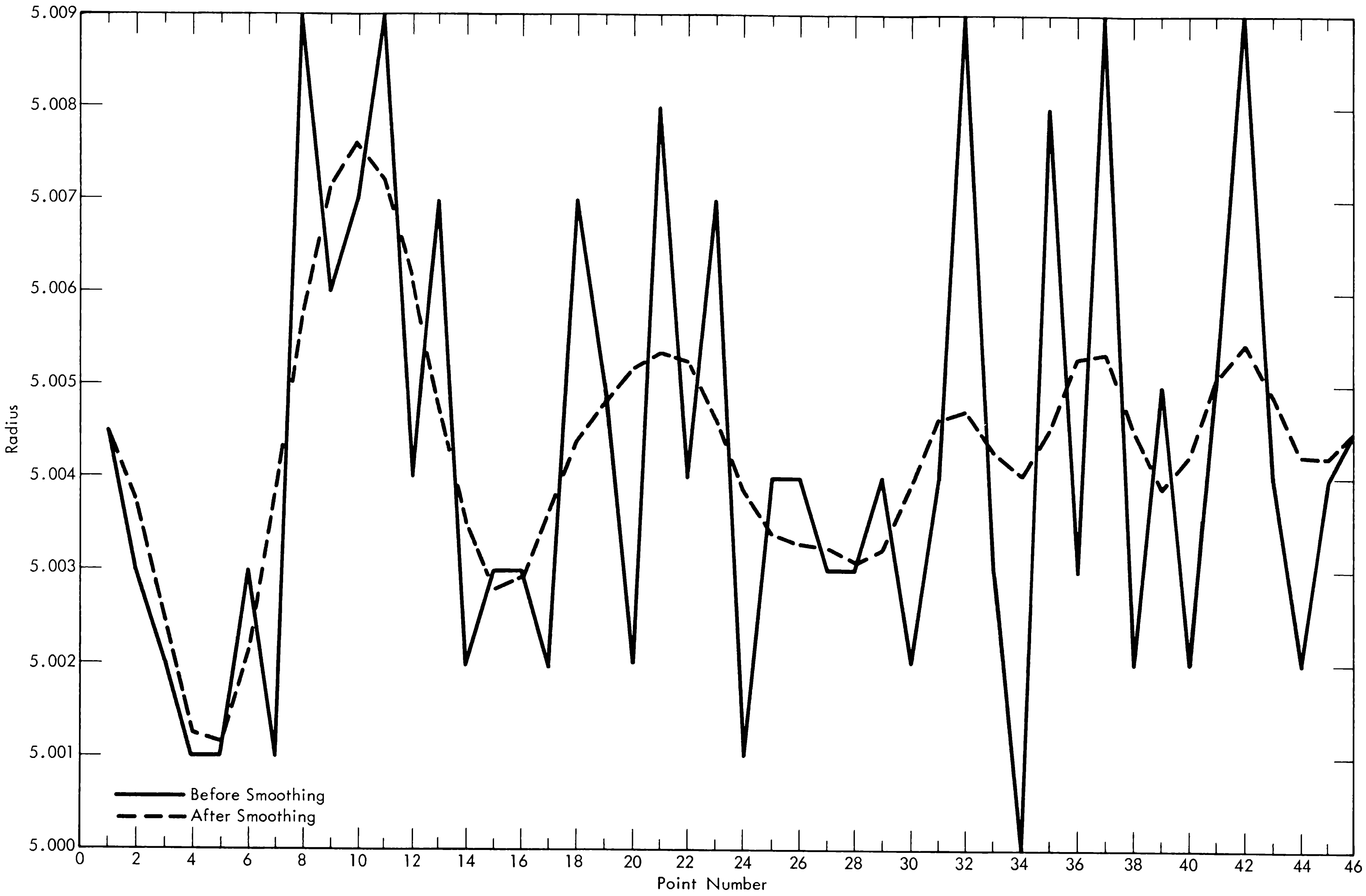

Figure B-5. RADIUS VALUES OF TEST CURVE.

Table B-5

POINT PARAMETERS FOR TEST CURVE BEFORE MOVEMENT

CIRCLE.RAD#5.0045.TEST PT MOVE   DATE- 08/23/62   TIME- 16 HRS 45 MIN 00 SEC   PAGE- 2

NUM	THETA	RADIUS	X-CORD	Y-CORD	SEG LENGTH	SEG ANGLE	EXT ANGLE	MOVED
1	0.	5.004500	5.004500	0.	0.174661	-88.5080	0.	0.
2	2.0000	5.003000	4.999952	0.174602	0.174614	-86.6719	1.8361	0.
3	4.0000	5.002000	4.989815	0.348922	0.174579	-84.6718	2.0001	0.
4	6.0000	5.001000	4.973604	0.522747	0.174559	-83.0000	1.6718	0.
5	8.0000	5.001000	4.952331	0.696005	0.174605	-81.6562	1.3438	0.
6	10.0000	5.003000	4.926993	0.868762	0.174605	-78.3438	3.3124	0.
7	12.0000	5.001000	4.891716	1.039766	0.174882	-79.6215	-1.2777	0.
8	14.0000	5.009000	4.860211	1.211787	0.174812	-74.0169	5.6046	0.
9	16.0000	5.006000	4.812076	1.379841	0.174754	-73.3278	0.6890	0.
10	18.0000	5.007000	4.761940	1.547248	0.174815	-71.6554	1.6724	0.
11	20.0000	5.009000	4.706920	1.713179	0.174822	-67.3613	4.2941	0.
12	22.0000	5.004000	4.639628	1.874531	0.174742	-67.9836	-0.6222	0.
13	24.0000	5.007000	4.574122	2.036530	0.174753	-63.3607	4.6228	0.
14	26.0000	5.002000	4.495768	2.192732	0.174614	-63.3280	0.0327	0.
15	28.0000	5.003000	4.417387	2.348766	0.174629	-61.0000	2.3281	0.
16	30.0000	5.003000	4.332725	2.501500	0.174614	-58.6719	2.3280	0.
17	32.0000	5.002000	4.241937	2.650656	0.174753	-58.6393	0.0326	0.
18	34.0000	5.007000	4.150991	2.799879	0.174745	-54.3443	4.2950	0.
19	36.0000	5.005000	4.049130	2.941865	0.174672	-52.0161	2.3283	0.
20	38.0000	5.002000	3.941630	3.079539	0.174802	-52.9667	-0.9507	0.
21	40.0000	5.008000	3.836351	3.219080	0.174779	-47.6888	5.2779	0.
22	42.0000	5.004000	3.718697	3.348329	0.174742	-47.9836	-0.2948	0.
23	44.0000	5.007000	3.601734	3.478154	0.174767	-43.0329	4.9507	0.
24	46.0000	5.001000	3.473986	3.597418	0.174637	-43.9841	-0.9513	0.
25	48.0000	5.004000	3.348329	3.718697	0.174664	-41.0000	2.9841	0.
26	50.0000	5.004000	3.216509	3.833286	0.174649	-38.6720	2.3280	0.
27	52.0000	5.003000	3.080154	3.942418	0.174629	-37.0000	1.6720	0.
28	54.0000	5.003000	2.940690	4.047512	0.174649	-35.3280	1.6720	0.
29	56.0000	5.004000	2.798201	4.148504	0.174640	-32.3439	2.9841	0.
30	58.0000	5.002000	2.650656	4.241937	0.174640	-31.6561	0.6879	0.
31	60.0000	5.004000	2.502000	4.333591	0.174822	-30.6387	1.0174	0.
32	62.0000	5.009000	2.351583	4.422684	0.174836	-25.0336	5.6050	0.
33	64.0000	5.003000	2.193171	4.496666	0.174602	-24.0157	1.0180	0.
34	66.0000	5.000000	2.033683	4.567727	0.174847	-25.6221	-1.6064	0.
35	68.0000	5.008000	1.876030	4.643337	0.174788	-19.3610	6.2610	0.
36	70.0000	5.003000	1.711127	4.701282	0.174836	-20.9663	-1.6053	0.
37	72.0000	5.009000	1.547866	4.763842	0.174856	-14.7060	6.2603	0.
38	74.0000	5.002000	1.378738	4.808231	0.174672	-15.9839	-1.2779	0.
39	76.0000	5.005000	1.210819	4.856330	0.174672	-12.0161	3.9679	0.
40	78.0000	5.002000	1.039974	4.892694	0.174672	-11.9840	0.0321	0.
41	80.0000	5.005000	0.869109	4.928963	0.174814	-10.3109	1.6731	0.
42	82.0000	5.009000	0.697118	4.960253	0.174822	-5.3613	4.9496	0.
43	84.0000	5.004000	0.523060	4.976587	0.174640	-4.3439	1.0174	0.
44	86.0000	5.002000	0.348922	4.989815	0.174640	-3.6560	0.6879	0.
45	88.0000	5.004000	0.174637	5.000952	0.174673	-1.1640	2.4920	0.
46	90.0000	5.004500	0.000000	5.004500	0.	0.	0.	0.

TOTAL LENGTH# 7.862020

Table B-6

CURVE PARAMETERS FOR TEST CURVE BEFORE MOVEMENT

CIRCLE,RAD#5.0045,TEST PT MOVE   DATE- 08/23/62   TIME- 16 HRS 45 MIN 00 SEC   PAGE- 3

NUM	SLOPE	NORMAL	ALPHA	TAN ANGLE A	TAN ANGLE B	CURV A	CURV B	DELTA CURV	ENERGY
1	-1.000000E 12	0.	-0.	-0.0260461	0.0184322	0.3850	0.1238	0.	0.0123
2	-0.224721E 02	2.5480	0.5480	-0.0136163	0.0163798	0.1243	0.2192	0.0005	0.0053
3	-0.134044E 02	4.2665	0.2665	-0.0185317	0.0179453	0.2189	0.1988	-0.0003	0.0076
4	-0.897711E 01	6.3562	0.3562	-0.0112363	0.0051208	0.1988	-0.0114	0.0000	0.0022
5	-0.781337E 01	7.2934	-0.7066	-0.0183352	0.0376895	-0.0117	0.6520	-0.0003	0.0244
6	-0.539434E 01	10.5022	0.5022	-0.0201427	-0.0167537	0.6530	-0.6143	0.0009	0.0234
7	-0.529420E 01	10.6963	-1.3037	0.0055473	0.0426469	-0.6146	1.0361	-0.0003	0.0476
8	-0.439423E 01	12.8205	-1.1795	-0.0552543	0.0195348	1.0361	-0.1851	0.0000	0.0538
9	-0.325009E 01	17.1023	1.1023	0.0075064	0.0011134	-0.1845	0.1114	0.0005	0.0015
10	-0.332556E 01	16.7360	-1.2640	-0.0280827	0.0464616	0.1109	0.7394	-0.0005	0.0375
11	-0.260445E 01	21.0047	1.0047	-0.0285257	-0.0076623	0.7394	-0.5016	0.0000	0.0249
12	-0.245147E 01	22.1997	0.1997	0.0031979	0.0374750	-0.5021	0.8926	-0.0005	0.0351
13	-0.222899E 01	24.1626	0.1626	-0.0432531	0.0082734	0.8928	-0.3056	0.0003	0.0361
14	-0.195305E 01	27.1133	1.1133	0.0077030	0.0112139	-0.3049	0.3450	0.0007	0.0062
15	-0.193627E 01	27.3144	-0.6856	-0.0294272	0.0287112	0.3448	0.3202	-0.0003	0.0194
16	-0.168791E 01	30.6446	0.6446	-0.0119292	-0.0041060	0.3202	-0.2307	0.0000	0.0048
17	-0.165819E 01	31.0928	-0.9072	-0.0046758	0.0295317	-0.2309	0.6216	-0.0003	0.0173
18	-0.153679E 01	33.0522	-0.9478	-0.0454699	0.0364733	0.6215	0.3139	-0.0002	0.0397
19	-0.129177E 01	37.7445	1.7445	-0.0041787	-0.0190540	0.3139	-0.4839	0.0000	0.0105
20	-0.133224E 01	36.8924	-1.1076	-0.0024592	0.0472315	-0.4841	1.0492	-0.0002	0.0484
21	-0.120291E 01	39.7374	-0.2626	-0.0449512	-0.0020376	1.0489	-0.5610	-0.0003	0.0484
22	-0.110306E 01	42.1945	0.1945	0.0031078	0.0428008	-0.5610	1.0125	0.0000	0.0451
23	-0.101877E 01	44.4672	0.4672	-0.0436599	-0.0013561	1.0119	-0.5307	-0.0006	0.0450
24	-0.936129E 00	46.8894	0.8894	0.0152482	0.0158573	-0.5307	0.5376	0.0000	0.0166
25	-0.934987E 00	46.9243	-1.0757	-0.0362428	0.0254733	0.5373	0.1682	-0.0004	0.0238
26	-0.825533E 00	50.4592	0.4592	-0.0151651	0.0155772	0.1689	0.1830	0.0007	0.0054
27	-0.775108E 00	52.2205	0.2205	-0.0136062	0.0111680	0.1837	0.1000	0.0007	0.0036
28	-0.736191E 00	53.6398	-0.3602	-0.0180166	0.0272999	0.1000	0.4185	-0.0000	0.0132
29	-0.668536E 00	56.2358	0.2358	-0.0247936	0.0129777	0.4189	0.0133	0.0004	0.0106
30	-0.615214E 00	58.3996	0.3996	0.0009713	-0.0030458	0.0126	-0.0586	-0.0007	0.0002
31	-0.620765E 00	58.1694	-1.8306	-0.0208058	0.0467426	-0.0587	0.8287	-0.0000	0.0376
32	-0.530869E 00	62.0375	0.0375	-0.0511620	0.0296143	0.8285	0.0922	-0.0003	0.0452
33	-0.431441E-00	66.6626	2.6626	0.0118393	-0.0318049	0.0931	-0.5921	0.0009	0.0177
34	-0.484223E-00	64.1626	-1.8374	-0.0037576	0.0593178	-0.5925	1.3071	-0.0004	0.0755
35	-0.408650E-00	67.7726	-0.2274	-0.0500693	-0.0145287	1.3072	-0.9051	0.0000	0.0787
36	-0.367797E-00	69.8066	-0.1934	0.0134915	0.0521990	-0.9055	1.3431	-0.0004	0.0825
37	-0.324500E-00	72.0217	0.0217	-0.0571732	-0.0037086	1.3437	-0.7388	0.0006	0.0797
38	-0.266426E-00	75.0815	1.0815	0.0185971	0.0273216	-0.7383	0.8377	0.0004	0.0366
39	-0.257108E-00	75.5811	-0.4189	-0.0419626	0.0105965	0.8374	-0.2378	-0.0003	0.0327
40	-0.201798E-00	78.5911	0.5911	0.0100364	0.0006965	-0.2378	0.1309	0.0000	0.0025
41	-0.211536E-00	78.0559	-1.9441	-0.0285117	0.0456346	0.1301	0.7158	-0.0007	0.0364
42	-0.135170E-00	82.3020	0.3020	-0.0408058	0.0188664	0.7160	-0.0351	0.0003	0.0286
43	-0.748481E-01	85.7195	1.7195	0.0011074	0.0008534	-0.0351	0.0322	-0.0000	0.0001
44	-0.751035E-01	85.7049	-0.2951	-0.0111532	0.0194687	0.0325	0.3180	0.0003	0.0065
45	-0.443726E-01	87.4593	-0.5407	-0.0240325	0.0203186	0.3174	0.1900	-0.0006	0.0115
46	-0.298023E-07	90.0000	-0.0000	0.	0.	0.	0.	0.	0.

TOTAL ENERGY# 1.2416

CIRCLE,RAD#5.0045,TEST PT MOVE   DATE- 08/23/62   TIME- 16 HRS 45 MIN 00 SEC   PAGE- 4

-1.8000   -0.8000   0.2000   1.2000   2.2000

	CURVATURE
1	0.3850
2	0.1243
3	0.2189
4	0.1988
5	-0.0117
6	0.6530
7	-0.6146
8	1.0361
9	-0.1845
10	0.1109
11	0.7394
12	-0.5021
13	0.8928
14	-0.3049
15	0.3448
16	0.3202
17	-0.2309
18	0.6215
19	0.3139
20	-0.4841
21	1.0489
22	-0.5610
23	1.0119
24	-0.5307
25	0.5373
26	0.1689
27	0.1837
28	0.1000
29	0.4189
30	0.0126
31	-0.0587
32	0.6285
33	0.0931
34	-0.5925
35	1.3072
36	-0.9055
37	1.3437
38	-0.7383
39	0.8374
40	-0.2378
41	0.1301
42	0.7160
43	-0.0351
44	0.0325
45	0.3174
46	0.1900
	CURVATURE

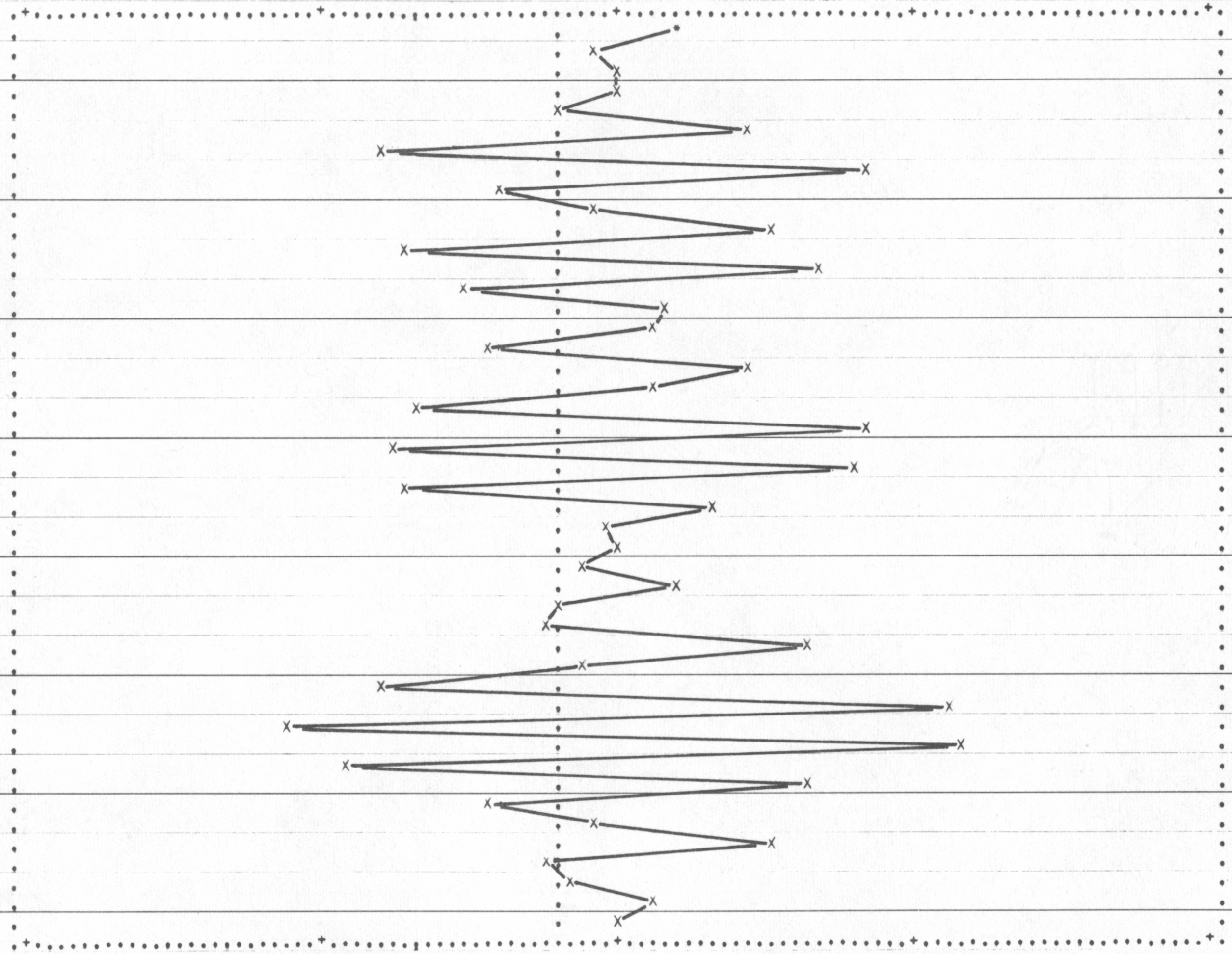

Figure B-6. CURVATURE OF TEST CURVE BEFORE MOVEMENT.

Table B-7

POINT PARAMETERS FOR TEST CURVE AFTER MOVEMENT

CIRCLE,RAD#5.0045,TEST PT MOVE   DATE- 08/23/62   TIME- 16 HRS 45 MIN 00 SEC   PAGE- 5

NUM	THETA	RADIUS	X-CORD	Y-CORD	SEG LENGTH	SEG ANGLE	EXT ANGLE	MOVED
1	0.	5.004500	5.004500	0.	0.174670	-88.7511	0.	0.
2	2.0000	5.003741	5.000693	0.174628	0.174636	-86.5302	2.2209	0.000741
3	4.0000	5.002309	4.990124	0.348943	0.174589	-84.6537	1.8765	0.000309
4	6.0000	5.001254	4.973856	0.522773	0.174566	-82.9696	1.6841	0.000254
5	8.0000	5.001161	4.952490	0.696027	0.174584	-81.3188	1.6508	0.000161
6	10.0000	5.002133	4.926139	0.868611	0.174637	-79.5683	1.7506	-0.000867
7	12.0000	5.003865	4.894519	1.040362	0.174702	-77.6161	1.9521	0.002865
8	14.0000	5.005744	4.857052	1.210999	0.174754	-75.4482	2.1679	-0.003256
9	16.0000	5.007111	4.813144	1.380147	0.174782	-73.1608	2.2874	0.001111
10	18.0000	5.007602	4.762512	1.547434	0.174783	-70.8748	2.2860	0.000602
11	20.0000	5.007220	4.705248	1.712570	0.174761	-68.6601	2.2146	-0.001780
12	22.0000	5.006183	4.641652	1.875349	0.174721	-66.5451	2.1150	0.002183
13	24.0000	5.004796	4.572108	2.035634	0.174674	-64.5718	1.9733	-0.002204
14	26.0000	5.003490	4.497107	2.193386	0.174635	-62.7683	1.8035	0.001490
15	28.0000	5.002784	4.417196	2.348665	0.174624	-61.0480	1.7203	-0.000216
16	30.0000	5.002930	4.332665	2.501465	0.174641	-59.2378	1.8102	-0.000070
17	32.0000	5.003655	4.243340	2.651533	0.174666	-57.2433	1.9944	0.001655
18	34.0000	5.004397	4.146833	2.798423	0.174686	-55.1534	2.0899	-0.002603
19	36.0000	5.004865	4.049021	2.941786	0.174699	-53.0978	2.0557	-0.000135
20	38.0000	5.005163	3.944122	3.081486	0.174706	-51.0641	2.0337	0.003163
21	40.0000	5.005358	3.834327	3.217382	0.174709	-48.9586	2.1055	-0.002642
22	42.0000	5.005232	3.719612	3.349154	0.174697	-46.8040	2.1545	0.001232
23	44.0000	5.004634	3.600033	3.476511	0.174674	-44.7496	2.0544	-0.002366
24	46.0000	5.003871	3.475981	3.599483	0.174652	-42.8463	1.9033	0.002871
25	48.0000	5.003402	3.347929	3.718252	0.174641	-40.9637	1.8825	-0.000598
26	50.0000	5.003292	3.216054	3.832744	0.174638	-38.9786	1.9852	-0.000708
27	52.0000	5.003226	3.080294	3.942596	0.174634	-36.9569	2.0217	0.000226
28	54.0000	5.003095	2.940745	4.047589	0.174635	-35.0482	1.9087	0.000095
29	56.0000	5.003242	2.797777	4.147875	0.174650	-33.2158	1.8325	-0.000758
30	58.0000	5.003900	2.651663	4.243548	0.174674	-31.2387	1.9771	0.001900
31	60.0000	5.004628	2.502314	4.334135	0.174688	-29.0379	2.2008	0.000628
32	62.0000	5.004743	2.349585	4.418926	0.174682	-26.8432	2.1947	-0.004257
33	64.0000	5.004265	2.193725	4.497804	0.174669	-24.9154	1.9278	0.001265
34	66.0000	5.004007	2.035313	4.571388	0.174674	-23.1684	1.7470	0.004007
35	68.0000	5.004520	1.874726	4.640110	0.174697	-21.2483	1.9201	-0.003480
36	70.0000	5.005278	1.711906	4.703422	0.174709	-19.0154	2.2329	0.002278
37	72.0000	5.005325	1.546730	4.760347	0.174698	-16.7485	2.2669	-0.003675
38	74.0000	5.004557	1.379443	4.810689	0.174673	-14.7920	1.9565	0.002558
39	76.0000	5.003923	1.210559	4.855285	0.174666	-13.0990	1.6930	-0.001077
40	78.0000	5.004225	1.040437	4.894871	0.174689	-11.2835	1.8155	0.002225
41	80.0000	5.005089	0.869125	4.929051	0.174708	-9.1170	2.1665	0.000089
42	82.0000	5.005446	0.696623	4.956734	0.174706	-6.8259	2.2911	-0.003554
43	84.0000	5.004915	0.523156	4.977498	0.174685	-4.7850	2.0409	0.000916
44	86.0000	5.004260	0.349079	4.992070	0.174673	-2.9947	1.7903	0.002260
45	88.0000	5.004244	0.174645	5.001195	0.174677	-1.0841	1.9106	0.000244
46	90.0000	5.004500	0.000000	5.004500	0.	0.	0.	0.

TOTAL LENGTH# 7.860414

Table B-8

CURVE PARAMETERS FOR TEST CURVE AFTER MOVEMENT

CIRCLE,RAD#5.0045,TEST PT MOVE DATE- 08/23/62 TIME- 16 HRS 45 MIN 00 SEC PAGE- 6

NUM	SLOPE	NORMAL	ALPHA	TAN ANGLE A	TAN ANGLE B	CURV A	CURV B	DELTA CURV	ENERGY
1	-1.000000E 12	0.	-0.	-0.0218010	0.0205326	0.2640	0.2204	0.	0.0103
2	-0.236113E 02	2.4252	0.4252	-0.0182337	0.0171848	0.2207	0.1847	0.0003	0.0072
3	-0.128371E 02	4.4543	0.4543	-0.0155699	0.0150095	0.1847	0.1655	-0.0000	0.0054
4	-0.919585E 01	6.2062	0.2062	-0.0143851	0.0142977	0.1658	0.1628	0.0003	0.0047
5	-0.725355E 01	7.8495	-0.1505	-0.0145157	0.0148160	0.1628	0.1731	0.0000	0.0049
6	-0.595662E 01	9.5300	-0.4700	-0.0157398	0.0163582	0.1731	0.1943	-0.0000	0.0059
7	-0.497338E 01	11.3689	-0.6311	-0.0177161	0.0183903	0.1950	0.2181	0.0007	0.0075
8	-0.418543E 01	13.4374	-0.5626	-0.0194515	0.0198079	0.2184	0.2306	0.0003	0.0088
9	-0.356083E 01	15.6865	-0.3135	-0.0201205	0.0200732	0.2306	0.2290	-0.0000	0.0092
10	-0.307967E 01	17.9892	-0.0108	-0.0198310	0.0196121	0.2293	0.2218	0.0003	0.0089
11	-0.271080E 01	20.2488	0.2488	-0.0190454	0.0187865	0.2208	0.2119	-0.0010	0.0082
12	-0.242424E 01	22.4161	0.4161	-0.0181320	0.0177153	0.2122	0.1979	0.0003	0.0074
13	-0.219737E 01	24.4698	0.4698	-0.0167280	0.0161631	0.1979	0.1785	-0.0000	0.0062
14	-0.201854E 01	26.3542	0.3542	-0.0153171	0.0150140	0.1788	0.1684	0.0003	0.0053
15	-0.187347E 01	28.0919	0.0919	-0.0150130	0.0153246	0.1683	0.1790	-0.0001	0.0053
16	-0.174398E 01	29.8300	-0.1700	-0.0162727	0.0169071	0.1790	0.2008	-0.0000	0.0063
17	-0.161719E 01	31.7308	-0.2692	-0.0179060	0.0181997	0.2016	0.2117	0.0008	0.0075
18	-0.149382E 01	33.7993	-0.2007	-0.0182796	0.0180919	0.2113	0.2049	-0.0003	0.0076
19	-0.138231E 01	35.8830	-0.1170	-0.0177903	0.0176755	0.2049	0.2009	0.0000	0.0072
20	-0.128387E 01	37.9149	-0.0851	-0.0178228	0.0181123	0.2006	0.2106	-0.0003	0.0074
21	-0.119287E 01	39.9736	-0.0264	-0.0186397	0.0188594	0.2108	0.2183	0.0002	0.0080
22	-0.110587E 01	42.1219	0.1219	-0.0187408	0.0183920	0.2186	0.2064	0.0003	0.0079
23	-0.102654E 01	44.2496	0.2496	-0.0174686	0.0169052	0.2064	0.1870	-0.0000	0.0068
24	-0.958332E 00	46.2189	0.2189	-0.0163174	0.0162677	0.1874	0.1856	0.0003	0.0061
25	-0.897698E 00	48.0857	0.0857	-0.0165916	0.0170220	0.1850	0.1998	-0.0006	0.0065
26	-0.838759E 00	50.0115	0.0115	-0.0176292	0.0177840	0.2000	0.2053	0.0003	0.0072
27	-0.780154E 00	52.0403	0.0403	-0.0175046	0.0170706	0.2053	0.1904	0.0000	0.0068
28	-0.725980E 00	54.0211	0.0211	-0.0162448	0.0159183	0.1897	0.1765	-0.0007	0.0059
29	-0.677974E 00	55.8637	-0.1363	-0.0160608	0.0165705	0.1782	0.1954	-0.0003	0.0061
30	-0.631354E 00	57.7336	-0.2664	-0.0179391	0.0188005	0.1954	0.2250	0.0000	0.0077
31	-0.581116E 00	59.8383	-0.1617	-0.0196157	0.0196361	0.2242	0.2249	-0.0008	0.0088
32	-0.529763E 00	62.0870	0.0870	-0.0180735	0.0176921	0.2249	0.1912	0.0000	0.0076
33	-0.484057E-00	64.1704	0.1704	-0.0159575	0.0152811	0.1904	0.1672	-0.0008	0.0056
34	-0.446064E-00	65.9601	-0.0399	-0.0152120	0.0158438	0.1669	0.1886	-0.0003	0.0055
35	-0.409329E-00	67.7393	-0.2607	-0.0176708	0.0188381	0.1889	0.2289	0.0003	0.0077
36	-0.367316E-00	69.8309	-0.1691	-0.0201390	0.0202698	0.2289	0.2334	-0.0000	0.0093
37	-0.322106E-00	72.1458	0.1458	-0.0193000	0.0181461	0.2340	0.1944	0.0006	0.0080
38	-0.281255E-00	74.2911	0.2911	-0.0160040	0.0150207	0.1944	0.1607	-0.0000	0.0055
39	-0.248057E-00	76.0686	0.0686	-0.0145301	0.0149913	0.1610	0.1769	0.0004	0.0050
40	-0.216940E-00	77.7599	-0.2401	-0.0166980	0.0179977	0.1762	0.2208	-0.0007	0.0069
41	-0.180872E-00	79.7476	-0.2524	-0.0198188	0.0202561	0.2217	0.2367	0.0009	0.0092
42	-0.139766E-00	82.0434	0.0434	-0.0197359	0.0187868	0.2367	0.2041	-0.0001	0.0085
43	-0.100689E-00	84.2503	0.2503	-0.0168380	0.0158859	0.2036	0.1709	-0.0005	0.0061
44	-0.677327E-01	86.1251	0.1251	-0.0153638	0.0157951	0.1709	0.1857	0.0000	0.0056
45	-0.364891E-01	87.9103	-0.0898	-0.0175539	0.0189231	0.1852	0.2322	-0.0005	0.0076
46	-0.298023E-07	90.0000	-0.0000	0.	0.	0.	0.	0.	0.

TOTAL ENERGY# 0.3171

CIRCLE,RAD=5.0,45,TEST PT MOVE   DATE- 08/23/62   TIME- 16 HRS 45 MIN 00 SEC   PAGE- 7

	CURVATURE	-1.800 · -0.8000 · 0.2000 · 1.2000 · 2.2000
1	0.2640	
2	0.2207	
3	0.1847	
4	0.1658	
5	0.1628	
6	0.1731	
7	0.1950	
8	0.2184	
9	0.2306	
10	0.2293	
11	0.2208	
12	0.2122	
13	0.1979	
14	0.1788	
15	0.1683	
16	0.1790	
17	0.2016	
18	0.2113	
19	0.2049	
20	0.2006	
21	0.2108	
22	0.2186	
23	0.2064	
24	0.1874	
25	0.1850	
26	0.2000	
27	0.2053	
28	0.1897	
29	0.1782	
30	0.1954	
31	0.2242	
32	0.2249	
33	0.1904	
34	0.1669	
35	0.1889	
36	0.2289	
37	0.2340	
38	0.1944	
39	0.1610	
40	0.1762	
41	0.2217	
42	0.2367	
43	0.2036	
44	0.1709	
45	0.1852	
46	0.2322	
	CURVATURE	

Figure B-7. CURVATURE PLOT OF TEST CURVE AFTER MOVEMENT.

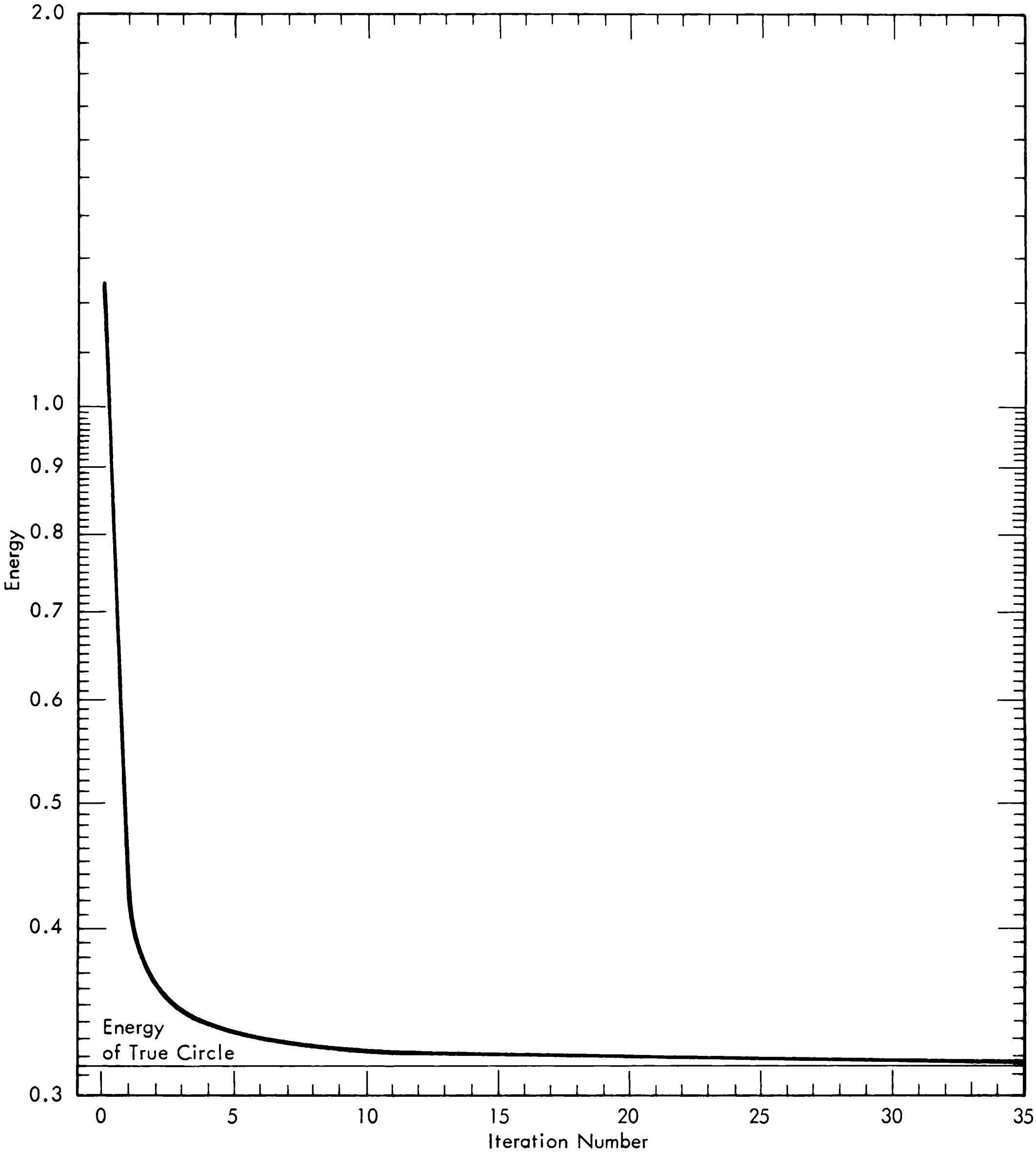

Figure B-8. STRAIN ENERGY OF CUBIC SPLINE AFTER EACH ITERATION OF POINT MOVING ROUTINE.

INTERPOLATION AND OFFSET (Example 3)

The sine curve used in Example 1 was run through the interpolation and offset routine with TOLER = 0.001, OFFCON = 0.25, and OFFVAR = 0.0. The point and curve parameters are the same as in Tables B-1 and B-3. The results of the interpolation routine are represented by the first two columns of Table B-9, and the results of the offset routine are shown in the other two columns of the same table.

Table B-9

SAMPLE OUTPUT PAGE FROM INTERPOLATION AND OFFSET ROUTINES

SINE CURVE- INTER AND OFFSET DATE- 12/11/6K TIME- 21 HRS 26 MIN 00 SEC PAGE- 5

SEG	PT	CURVE POINTS X	CURVE POINTS Y	OFFSET POINTS X	OFFSET POINTS Y
1	1	0.	0.	-0.223493	0.112031
2	1	0.174533	0.347300	-0.048144	0.460954
3	1	0.349066	0.684040	0.128632	0.802022
4	1	0.523599	1.000000	0.306594	1.124191
4	2	0.608915	1.143982	0.395579	1.274363
5	1	0.698132	1.285580	0.488826	1.422359
5	2	0.782847	1.410636	0.578769	1.555132
6	1	0.872664	1.532080	0.675354	1.685727
6	2	0.956819	1.634785	0.768374	1.799251
7	1	1.047197	1.732060	0.870398	1.909060
7	2	1.131038	1.809779	0.969609	2.001027
8	1	1.221730	1.879380	1.080485	2.086119
8	2	1.305970	1.930353	1.190892	2.152925
9	1	1.396263	1.969620	1.314243	2.206568
9	2	1.482252	1.992152	1.439229	2.239319
10	1	1.570796	2.000000	1.570796	2.250980
10	2	1.659340	1.992152	1.702363	2.239319
11	1	1.745329	1.969620	1.827349	2.206568
11	2	1.835622	1.930353	1.950700	2.152925
12	1	1.919862	1.879380	2.061106	2.086119
12	2	2.010554	1.809779	2.171983	2.001027
13	1	2.094395	1.732060	2.271194	1.909061
13	2	2.184774	1.634786	2.373219	1.799252
14	1	2.268928	1.532080	2.466240	1.685724
14	2	2.358743	1.410634	2.562820	1.555130
15	1	2.443461	1.285580	2.652763	1.422364
15	2	2.532682	1.143985	2.746019	1.274365
16	1	2.617993	1.000000	2.835000	1.124187
17	1	2.792526	0.684040	3.012960	0.802022
18	1	2.967059	0.347300	3.189736	0.460954
19	1	3.141593	-0.	3.364972	0.112258
20	1	3.316126	-0.347300	3.538804	-0.233646
21	1	3.490658	-0.684040	3.711092	-0.566058
22	1	3.665191	-1.000000	3.882197	-0.875813
22	2	3.750503	-1.143985	3.963839	-1.013604
23	1	3.839724	-1.285580	4.049026	-1.148796
23	2	3.924442	-1.410634	4.128519	-1.266138
24	1	4.014257	-1.532080	4.211569	-1.378436
24	2	4.098411	-1.634786	4.286856	-1.470320
25	1	4.188790	-1.732060	4.365588	-1.555059
25	2	4.272630	-1.809779	4.434059	-1.618531
26	1	4.363323	-1.879380	4.504567	-1.672641
26	2	4.447563	-1.930353	4.562641	-1.707781
27	1	4.537856	-1.969620	4.619876	-1.732672
27	2	4.623845	-1.992152	4.666867	-1.744985
28	1	4.712389	-2.000000	4.723447	-1.750000

UNCLASSIFIED

UNCLASSIFIED