NASA TECHNICAL NOTE

NASA TN D-7397

NASA TN D-7397

A SMOOTHING ALGORITHM USING CUBIC SPLINE FUNCTIONS

by Robert E. Smith, Jr., Joseph M. Price,
and Lona M. Howser

Langley Research Center
Hampton, Va. 23665

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • FEBRUARY 1974

1. Report No.
NASA TN D-7397

2. Government Accession No.

3. Recipient's Catalog No.

4. Title and Subtitle
A SMOOTHING ALGORITHM USING CUBIC SPLINE FUNCTIONS

5. Report Date
February 1974

6. Performing Organization Code

7. Author(s)
Robert E. Smith, Jr., Joseph M. Price, and Lona M. Howser

8. Performing Organization Report No.
L-8945

9. Performing Organization Name and Address
NASA Langley Research Center
Hampton, Va. 23665

10. Work Unit No.
501-06-01-11

11. Contract or Grant No.

12. Sponsoring Agency Name and Address
National Aeronautics and Space Administration
Washington, D.C. 20546

13. Type of Report and Period Covered
Technical Note

14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

Two algorithms are presented for smoothing arbitrary sets of data. They are the explicit variable algorithm and the parametric variable algorithm. The former would be used where large gradients are not encountered because of the smaller amount of calculation required. The latter would be used if the data being smoothed were double valued or experienced large gradients. Both algorithms use a least-squares technique to obtain a cubic spline fit to the data. The advantage of the spline fit is that the first and second derivatives are continuous. This method is best used in an interactive graphics environment so that the junction values for the spline curve can be manipulated to improve the fit.

17. Key Words (Suggested by Author(s))
Spline functions
Data smoothing
Function approximation
Least-squares theory
Interactive computer graphics

18. Distribution Statement
Unclassified – Unlimited

STAR Category 19

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified	84	$3.75

*For sale by the National Technical Information Service, Springfield, Virginia 22151

A SMOOTHING ALGORITHM USING CUBIC SPLINE FUNCTIONS

By Robert E. Smith, Jr., Joseph M. Price, and Lona M. Howser
Langley Research Center

SUMMARY

A technique for smoothing arbitrary sets of two-dimensional data is described. Both explicit and parametric cubic spline functions are used in a least-squares algorithm to obtain a least-squares cubic polynomial spline fit for smoothing data. The primary advantage of the least-squares spline technique is that a set of data can be represented by a continuous function with continuous first and second derivatives. The technique has been programed in FORTRAN IV for the Control Data series 6000 computer systems. Interactive computer graphics using a CDC 250 series CRT console and the Control Data series 6000 computer graphics system at the Langley Research Center is incorporated into the program to allow a user to interact with the program to aid in obtaining a satisfactory solution. A description of the computer program and the procedures for using it are included. Examples of applications are presented.

INTRODUCTION

Least-squares polynomial curve fitting is a standard smoothing technique for functionally representing a data set $S = \left\{(x_i,y_i)\right\}_{i=1}^{n}$. The process consists of finding the polynomial coefficients which minimize the sum of the squares of the differences between the polynomial curve and the values of the dependent variable in the data set. In order to obtain a better representation of a data set, polynomials can be fitted to a sequence of subsets of the data. The disadvantage of this type of polynomial fit – even though it may provide a closer representation of the data – is that there is no guarantee of continuity in either the polynomial representation or its derivatives over the entire data set. In contrast to this, applying the least-squares criteria to low-order spline polynomials guarantees continuity in the functional representation and certain of its derivatives while producing a good representation of the data set.

This concept is discussed by de Boor in references 1 and 2. He notes that the success of using spline functions in smoothing data lies in the proper choice of the "joints." In reference 2 he presents an algorithm to choose the number and position of the joints. His algorithm is limited as pointed out in reference 2 and is subject to failure on certain sets of data. The algorithm and computer program as described in this report emphasizes the user interaction with the algorithm to obtain a satisfactory solution. It should be noted, however, certain aspects of de Boor's algorithm (ref. 2) would work well in an interactive environment to aid the user selection of joints. The basic difference between de Boor's fixed joint algorithm (ref. 1) and Smith's algorithm (ref. 3) is the spline function representation and use of Lagrange multipliers. All three algorithms would produce similar results for data where globally large gradients are not encountered. In this report, spline smoothing is extended to data sets which may wander arbitrarily through a two-dimensional coordinate space and/or form a closed curve. This is done by introducing a parametric variable representation of the cubic spline function. Only cubic spline functions are dealt with in this report since they are naturally suitable to many engineering applications and offer a reasonable compromise between a simple and complex mathematical model.

SYMBOLS

$\left.\begin{matrix} A_j(x), B_j(x), \\ C_j(x), D_j(x) \end{matrix}\right\}$	coefficient functions for cubic polynomials
c_1, c_2	constants of integration
E	set of functions of coefficients
F	matrix of coefficients
$f(x)$	polynomial function in x
G	matrix of functions of coefficients
$\bar{g}(t)$	polynomial function in t
$g(x)$	function describing equality of first derivative of adjoining cubics at jth joint
$\bar{h}(t)$	polynomial function in t
i	index on elements in data sets
j	index of joints connecting cubic polynomials
m	number of joints

n	number of input data points
P	degree of polynomial spline function
R	vector of residuals
r	residual
S	ordered data set in two dimensions
$S_{\Delta}(x)$	polynomial spline function of x over mesh Δ
$S_{\Delta_x}(t)$	polynomial spline function of t over mesh Δ_x
$S_{\Delta_y}(t)$	polynomial spline function of t over mesh Δ_y
t	independent variable for parametric technique
t_i	parametric variable function of x_i and y_i
$\bar{t}_j$	abscissa of jth element in set of joints for parametric independent variable
$\hat{t}_j$	abscissa of jth element in set of joints for parametric dependent variable
W	diagonal matrix of weights
w	weight
X	vector of ordinates of junction points and their second derivatives
x	independent variable for explicit technique or dependent variable for parametric technique
x_i	ith abscissa of data set S
$\bar{x}_j$	abscissa of jth element in set of joints
Y	vector of ordinates of data points

y	dependent variable for explicit technique
y_i	ith ordinate in data set S
$\bar{y}_j$	ordinate of jth element in set of joints
Δ	set of interior mesh abscissas
Δ_x, Δ_y	sets of mesh abscissas for function of t
Λ	vector of Lagrange multipliers
λ_j	Lagrange multiplier associated with jth spline
σ	standard deviation for explicit variable solution
σ_x, σ_y	standard deviations for implicit variable solution
φ	function to be minimized
$\overline{\varphi}$	constrained function to be minimized

A prime indicates first derivative with respect to the independent variable; a double prime indicates second derivative with respect to the independent variable.

DERIVATION

The theory of least squares is applied to cubic polynomial spline functions $S_\Delta(x)$ to derive the least-squares cubic polynomial fit for data smoothing. The least-squares development in a vector notation is reviewed in appendix A and the development of polynomial spline functions is discussed in detail in references 4 and 5. This report presents a brief development of the cubic polynomial spline function and the application of least squares to cubic splines. The explicit cubic polynomial splines are derived first, followed by a description of parametric splines and their relationship to explicit variables.

Cubic Polynomial Splines

A polynomial spline function is defined over an interval $x_1 \leqq x \leqq x_n$ which is subdivided into a mesh

$$\Delta : x_1 = \bar{x}_1 < \bar{x}_2 < . . . \bar{x}_j . . . < \bar{x}_m = x_n$$

Associated with this mesh is a set $\{\bar{y}_j\}_{j=1}^{m}$. A polynomial spline function $S_\Delta(x)$ is a set of $m - 1$ polynomials of degree P connecting the points $\{(\bar{x}_j, \bar{y}_j)\}_{j=1}^{m}$ (called joints) such that the adjoining polynomials and the first $P - 1$ derivatives are continuous at the joints.

A cubic spline function $S_\Delta(x)$ is a set of cubic polynomials of degree three connecting the mesh points $\{(\bar{x}_j, \bar{y}_j)\}$ such that at the mesh points adjoining cubics and their first and second derivatives are continuous. This implies that $S_\Delta(x)$, $S'_\Delta(x)$, and $S''_\Delta(x)$ are continuous in the interval $\bar{x}_1 \leqq x \leqq \bar{x}_m$. The derivation of a cubic spline fit to the set $\{(\bar{x}_j, \bar{y}_j)\}_{j=1}^{m}$ can be performed by first examining a single cubic in the interval $\bar{x}_j \leqq x \leqq \bar{x}_{j+1}$ and enforcing the continuity conditions with the adjoining cubic in the interval $\bar{x}_{j-1} \leqq x \leqq \bar{x}_j$. The following is a detailed explanation of the procedure.

Let $(\bar{x}_j, \bar{y}_j)$ and $(\bar{x}_{j+1}, \bar{y}_{j+1})$ be two joints (see fig. 1(a)) between which it is desirable to construct a cubic polynomial $y(x)$. Since $y(x)$ is a cubic polynomial, $y''(x)$ (second derivative of $y(x)$) is a linear polynomial. The point slope representation (fig. 1(b)) for the second derivative is:

$$y''(x) = \frac{y''(\bar{x}_{j+1}) - y''(\bar{x}_j)}{\bar{x}_{j+1} - \bar{x}_j}(x - \bar{x}_j) + y''(\bar{x}_j) \tag{1}$$

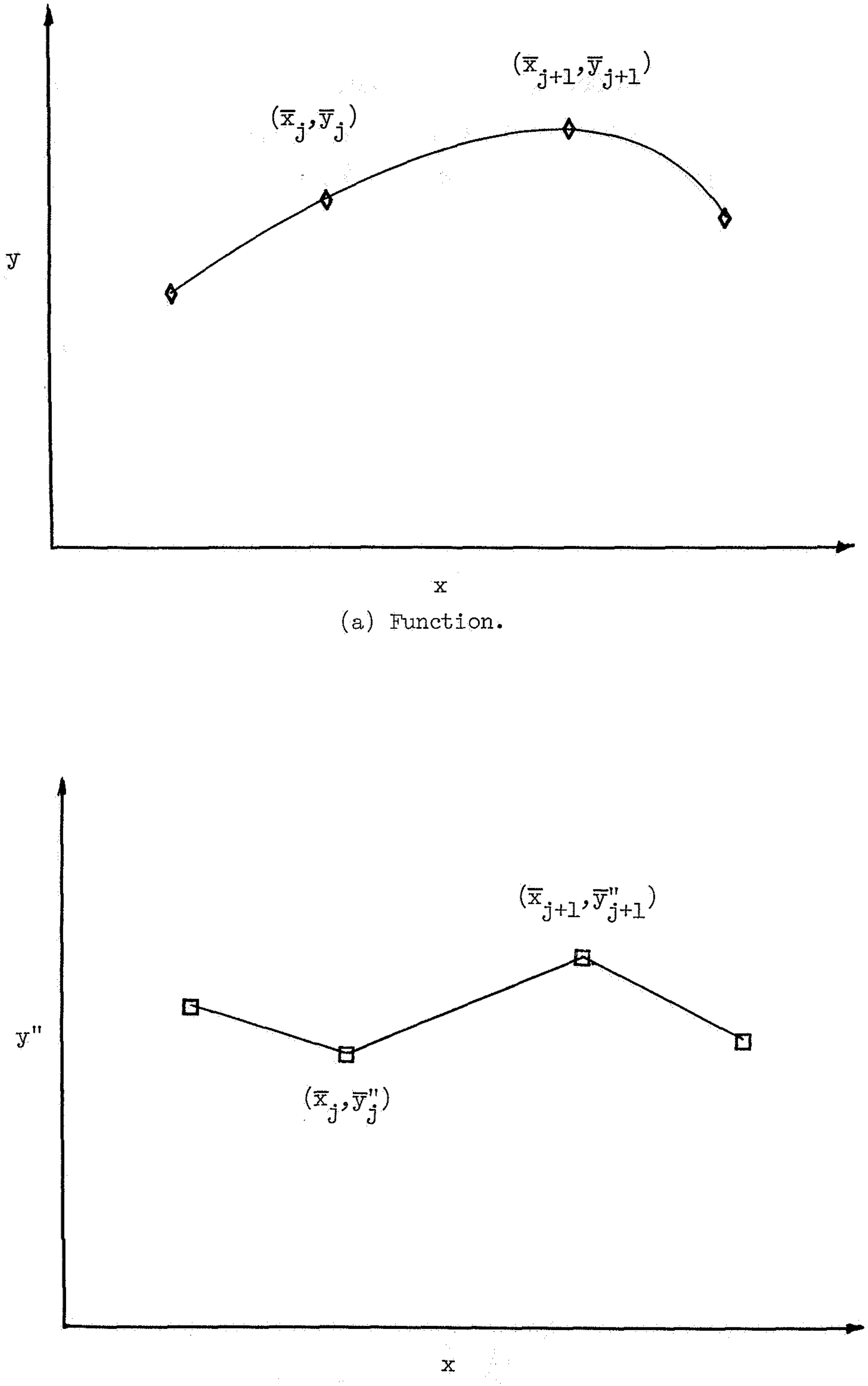
Letting $h_j = \bar{x}_{j+1} - \bar{x}_j$, $\bar{y}''_j = y''(\bar{x}_j)$, and $\bar{y}''_{j+1} = y''(\bar{x}_{j+1})$, equation (1) can be rewritten

$$y''(x) = \bar{y}''_{j+1} \frac{(x - \bar{x}_j)}{h_j} + \bar{y}''_j \frac{(\bar{x}_{j+1} - x)}{h_j} \tag{2}$$

Integrating equation (2) twice, the following equations are obtained:

$$y'(x) = \bar{y}''_{j+1} \frac{(x - \bar{x}_j)^2}{2h_j} - \bar{y}''_j \frac{(\bar{x}_{j+1} - x)^2}{2h_j} + c_1 \tag{3}$$

$$y(x) = \bar{y}''_{j+1} \frac{(x - \bar{x}_j)^3}{6h_j} + \bar{y}''_j \frac{(\bar{x}_{j+1} - x)^3}{6h_j} + c_1 x + c_2 \tag{4}$$


$(\overline{x}_{j+1},\overline{y}_{j+1})$
$(\overline{x}_j,\overline{y}_j)$
y
x
$(\overline{x}_{j+1},\overline{y}''_{j+1})$
y"
$(\overline{x}_j,\overline{y}''_j)$
x

(a) Function.

(b) Second derivative.

Figure 1.- Adjoining cubic polynomials.

Evaluating equation (4) at the jth and (j + 1)th joints yields equations for c_1 and c_2 as follows:

$$c_1 = \frac{\bar{y}_{j+1} - \bar{y}_j}{h_j} - \left(\frac{y''_{j+1} - y''_j}{6}\right)h_j$$

$$c_2 = y_j - \frac{y''_j h_j^2}{6} + \frac{(y''_{j+1} - y''_j)h_j\bar{x}_j}{6} - \frac{(\bar{y}_{j+1} - \bar{y}_j)\bar{x}_j}{h_j}$$

Collecting terms as coefficients of $\bar{y}_j$, $\bar{y}''_j$, $\bar{y}_{j+1}$, and $\bar{y}''_{j+1}$ results in:

$$y'(x) = A'_j(x)\bar{y}_j + B'_j(x)\bar{y}''_j + C'_j(x)\bar{y}_{j+1} + D'_j(x)\bar{y}''_{j+1} \tag{5}$$

$$y(x) = A_j(x)\bar{y}_j + B_j(x)\bar{y}''_j + C_j(x)\bar{y}_{j+1} + D_j(x)\bar{y}''_{j+1} \tag{6}$$

where

$$A_j(x) = \frac{h_j - (x - x_j)}{h_j}$$

$$B_j(x) = \frac{1}{6h_j}\left[\left(\bar{x}_{j+1} - x\right)^3 + h_j^2\left(x - \bar{x}_j\right) - h_j^3\right]$$

$$C_j(x) = \frac{\left(x - \bar{x}_j\right)}{h_j}$$

$$D_j(x) = \frac{1}{6h_j}\left[\left(x - \bar{x}_j\right)^3 - h_j^2\left(x - \bar{x}_j\right)\right]$$

$$A'_j(x) = -\frac{1}{h_j}$$

$$B'_j(x) = \frac{1}{6h_j}\left[h_j^2 - 3\left(\bar{x}_{j+1} - x\right)^2\right]$$

$$C'_j(x) = \frac{1}{h_j}$$

$$D'_j(x) = \frac{1}{6h_j}\left[3\left(x - \bar{x}_j\right)^2 - h_j^2\right]$$

For each cubic there is one equation of the form of equation (6) and for m mesh points there will be $m - 1$ cubics. Having written the cubics in the form of equation (6), adjoining cubics at the interior joints $\{\bar{x}_2, \bar{x}_3, . . ., \bar{x}_{m-1}\}$ are equal and the second derivatives at the interior joints are equal. That is,

$$y(\bar{x}_j -) = y(\bar{x}_j +) \qquad (j = 2, . . ., m - 1)$$

$$y''(\bar{x}_j -) = y''(\bar{x}_j +) \qquad (j = 2, . . ., m - 1)$$

Enforcing the condition at the joints

$$y'(\bar{x}_j -) = y'(\bar{x}_j +)$$

yields the continuity of the spline function $S_\Delta(x)$. Using equation (5) with $j - 1$ and j as indices establishes the above condition in the form

$$\begin{aligned} g(\bar{x}_j) = {} & A'_{j-1}(\bar{x}_j)\bar{y}_{j-1} + \left[C'_{j-1}(\bar{x}_j) - A'_j(\bar{x}_j)\right]\bar{y}_j + B'_{j-1}(\bar{x}_j)y''_{j-1} \\ & + \left[D'_{j-1}(\bar{x}_j) - B'_j(\bar{x}_j)\right]y''_j - C'_j(\bar{x}_j)\bar{y}_{j+1} - D'_j(\bar{x}_j)y''_{j+1} \\ = {} & 0 \end{aligned} \qquad (7)$$

Since there are $m - 2$ interior points, there are $m - 2$ condition equations of the form of equation (7) that must be satisfied in order to have a cubic spline polynomial fit $S_\Delta(x)$.

Least-Squares Spline Solution

The technique of least-squares polynomial curve fitting for smoothing noisy or irregular data represented by the data set $S = \{(x_i,y_i)\}_{i=1}^{n}$ is standard. It consists of choosing a polynomial function and finding the polynomial coefficients which minimize the sum of the squares of the differences between the function values at the abscissas $\{x_i\}_{i=1}^{n}$ of the data set and the ordinates $\{y_i\}_{i=1}^{n}$ of the data set. In the above, a set of cubic polynomials of the form of equation (6) with coefficients $\{(\bar{y}_j,\bar{y}_j'')\}_{j=1}^{m}$ is defined on the interval $[x_1,x_n]$ where the interval is divided into the mesh

$$\Delta : x_1 = \bar{x}_1 < \bar{x}_2 < . . . \bar{x}_j . . . < \bar{x}_m = x_n$$

In addition, at each $\{\bar{x}_j\}_{j=2}^{m-1}$ the condition described by equation (7) must be satisfied. The resulting set of polynomials is a cubic spline function.

The following is a vector development of a least-squares solution procedure for determining the set $\{\bar{y}_j,\bar{y}_j''\}_{j=1}^{m}$. Given a data set $\{(x_i,y_i)\}_{i=1}^{n}$ in the interval $x_1 = \bar{x}_1 \leqq x \leqq \bar{x}_m = x_n$, the least-squares solution using equation (6) determines the set $\{(\bar{y}_j,\bar{y}_j'')\}_{j=1}^{m}$ which minimizes

$$\varphi = \sum_{i=1}^{n} r_i^2 = \sum_{i=1}^{n} \left[y_i - y(x_i)\right]^2 \tag{8}$$

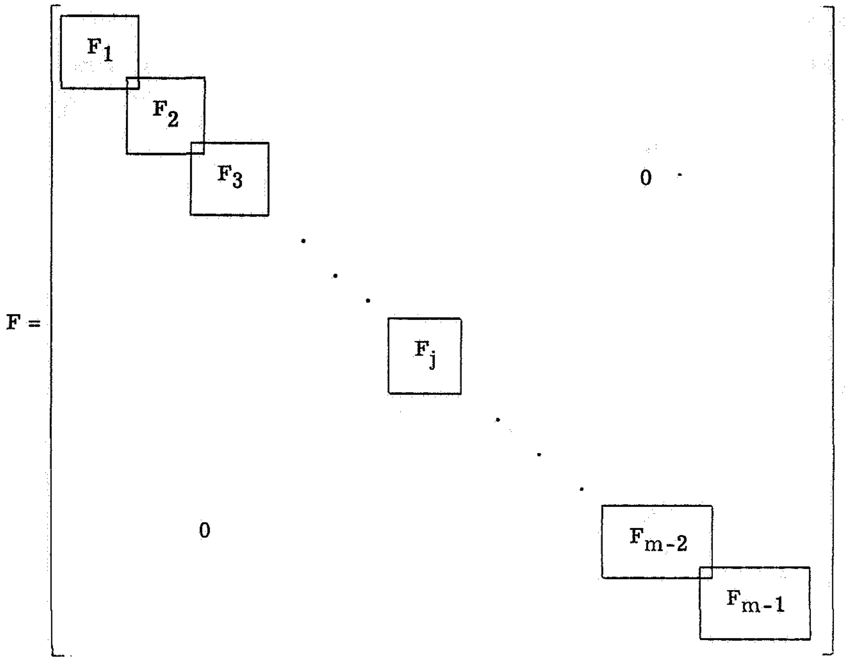
Using the method of Lagrange multipliers and adjoining the conditions of equation (7), the constrained least-squares solution minimizes

$$\overline{\varphi} = \sum_{i=1}^{n} \left[y_i - y(x_i)\right]^2 + \sum_{j=2}^{m-1} \lambda_j g(\bar{x}_j) \tag{9}$$

The derivation of the condition that minimizes $\overline{\varphi}$ is found in appendix A. This condition is expressed in matrix notation where the sets $\{y_i\}_{i=1}^{n}$, $\{(\bar{y}_j,\bar{y}_j'')\}_{j=1}^{m}$, and $\{(y_i - y(x_i))\}_{i=1}^{n}$ are ordered into the column vectors:

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ \cdot \\ \cdot \\ \cdot \\ y_i \\ \cdot \\ \cdot \\ \cdot \\ y_n \end{bmatrix} \qquad X = \begin{bmatrix} \bar{y}_1 \\ \bar{y}_1'' \\ \bar{y}_2 \\ \bar{y}_2'' \\ \cdot \\ \cdot \\ \cdot \\ \bar{y}_m \\ \bar{y}_m'' \end{bmatrix} \qquad R = \begin{bmatrix} y_1 - y(x_1) \\ y_2 - y(x_2) \\ \cdot \\ \cdot \\ \cdot \\ y_n - y(x_n) \end{bmatrix}$$

The sets $\{y(x_i)\}_{i=1}^{n}$ and $\{g(\bar{x}_j)\}_{j=2}^{m-1}$ found by evaluating equations (6) and (7) at $\{x_i\}_{i=1}^{n}$ and $\{\bar{x}_j\}_{j=2}^{m-1}$ are expressed as FX and GX, respectively, where



and

$$F_j = \begin{bmatrix} A_j(x_i) & B_j(x_i) & C_j(x_i) & D_j(x_i) \\ A_j(x_{i+1}) & B_j(x_{i+1}) & C_j(x_{i+1}) & D_j(x_{i+1}) \\ \vdots & \vdots & \vdots & \vdots \\ A_j(x_{s_j}) & B_j(x_{s_j}) & C_j(x_{s_j}) & D_j(x_{s_j}) \end{bmatrix}$$

F_j is a rectangular matrix with one row for each point in the interval $\bar{x}_j \leqq x \leqq \bar{x}_{j+1}$, s_j is the number of points in the interval, and F is the composite of all the F_j's. There is coupling between F_{j-1}, F_j, and F_{j+1} as indicated by the overlapping rectangles in F. Also,

$$G = \begin{bmatrix} E_{2,1} & E_{2,2} & E_{2,3} & E_{2,4} & E_{2,5} & E_{2,6} & & & 0 \\ & & E_{3,1} & E_{3,2} & E_{3,3} & E_{3,4} & E_{3,5} & E_{3,6} & \\ & & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & & E_{j,1} & E_{j,2} & E_{j,3} & E_{j,4} & E_{j,5} & E_{j,6} \\ & & & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ 0 & & & E_{m-1,1} & E_{m-1,2} & E_{m-1,3} & E_{m-1,4} & E_{m-1,5} & E_{m-1,6} \end{bmatrix}$$

and

$$E_{j,1} = A'_{j-1}(\bar{x}_j)$$

$$E_{j,2} = B'_{j-1}(\bar{x}_j)$$

$$E_{j,3} = \left[C'_{j-1}(\bar{x}_j) - A'_j(\bar{x}_j)\right]$$

$$E_{j,4} = \left[D'_{j-1}(\bar{x}_j) - B'_j(\bar{x}_j)\right]$$

$$E_{j,5} = -C'_j(\bar{x}_j)$$

$$E_{j,6} = -D'_j(\bar{x}_j) \qquad (j = 2, \ldots, m - 1)$$

An additional vector is needed to denote the Lagrange multipliers:

$$\Lambda = \begin{bmatrix} \lambda_2 \\ \lambda_3 \\ \cdot \\ \cdot \\ \cdot \\ \lambda_{m-1} \end{bmatrix}$$

Now equation (9) can be rewritten

$$\overline{\varphi} = \sum_{i=1}^{n} r_i^2 + \sum_{j=2}^{m-1} \lambda_j g(\bar{x}_j) = R^T R + \Lambda^T G X$$

or, since $R = Y - FX$,

$$\overline{\varphi} = \left[Y - FX\right]^T \left[Y - FX\right] + \Lambda^T G X$$

In appendix A the necessary and sufficient condition of φ to be a minimum is the existence of a unique solution for X in

$$\begin{bmatrix} X \\ \Lambda \end{bmatrix} = \begin{bmatrix} F^T F & G^T \\ G & 0 \end{bmatrix}^{-1} \begin{bmatrix} F^T Y \\ 0 \end{bmatrix} \qquad (10)$$

Often, when working with a set of data, it is desirable to place greater emphasis on certain points over others. This is achieved by associating a set of weights $\{w_i\}_{i=1}^{n}$ with the set $\{(x_i,y_i)\}_{i=1}^{n}$ such that each data point (x_i,y_i) has a weight w_i and minimizing

$$\overline{\varphi} = \sum_{i=1}^{n} (r_i w_i)^2 + \sum_{j=2}^{m-1} \lambda_j g(\bar{x}_j)$$

results in the following weighted least-squares solution

$$\begin{bmatrix} X \\ \Lambda \end{bmatrix} = \begin{bmatrix} F^T W F & G^T \\ G & 0 \end{bmatrix}^{-1} \begin{bmatrix} F^T W Y \\ 0 \end{bmatrix} \tag{11}$$

where

$$W = \begin{bmatrix} w_1^2 & & & & & & & 0 \\ & w_2^2 & & & & & & \\ & & \cdot & & & & & \\ & & & \cdot & & & & \\ & & & & \cdot & & & \\ & & & & w_i^2 & & & \\ & & & & & \cdot & & \\ & & & & & & \cdot & \\ & & & & & & \cdot & \\ 0 & & & & & & & w_n^2 \end{bmatrix}$$

The primary mathematical tool in the form of equation (11) is now available. In the next section a step-by-step procedure to apply equation (11) to obtain satisfactory smooth fits to the set $\{(x_i,y_i)\}_{i=1}^{n}$ is presented.

A Computational Algorithm for Explicit Variables $y = f(x)$

Given the sets $\{(x_i,y_i)\}_{i=1}^{n}$, $\bar{x}_1 = x_1$ and $\bar{x}_m = x_n$, and $\{w_i\}_{i=1}^{n}$:

(1) Choose $\{\bar{x}_j\}_{j=2}^{m-1}$.

(2) Determine the matrices F and G by computing the coefficients $A_j(x_i)$, $B_j(x_i)$, $C_j(x_i)$, $D_j(x_i)$, and $E_{j,k}$ for $k = 1, \ldots, 6$.

(3) Form the matrices $\begin{bmatrix} F^TWF & G^T \\ G & 0 \end{bmatrix}$ and $\begin{bmatrix} F^TWY \\ 0 \end{bmatrix}$.

(4) Find the inverse of $\begin{bmatrix} F^TWF & G^T \\ G & 0 \end{bmatrix}$ and solve $\begin{bmatrix} X \\ \Lambda \end{bmatrix} = \begin{bmatrix} F^TWF & G^T \\ G & 0 \end{bmatrix}^{-1} \begin{bmatrix} F^TWY \\ 0 \end{bmatrix}$.

(5) Compute the standard deviation

$$\sigma = \left[\sum_{i=1}^{m} \frac{(r_i w_i)^2}{m - 2m}\right]^{1/2} = \left\{\sum_{i=1}^{m} \frac{\left[y_i - S_\Delta(x_i)\right]^2}{m - 2m}\right\}^{1/2}$$

(6) Plot $\{(x_i,y_i)\}_{i=1}^{n}$ and the spline function $S_\Delta(x)$.

(7) If the resulting fit of $S_\Delta(x)$ to the data $\{(x_i,y_i)\}_{i=1}^{n}$ is not satisfactory (i.e., σ is not sufficiently small or $S_\Delta(x)$ is not satisfactory from an engineering point of view), choose a new set $\{\bar{x}_j\}_{j=2}^{m-1}$ and restart at (2). The number of joints m is not necessarily the same as in the previous attempt; however, the endpoints $\bar{x}_1$ and $\bar{x}_m$ must still be equal, respectively, to x_1 and x_n. Repeat until one of the following conclusions is reached:

(a) The fit is satisfactory and $S_\Delta(x)$ can be used to represent $\{(x_i,y_i)\}_{i=1}^{n}$.

(b) The fit is unsatisfactory and the technique is either abandoned or the parametric variable technique as described in the following section is applied.

This approach is highly feasible in an interactive computer graphic environment. Rapid successive solutions can be attained and plotted for use in making a decision on the acceptability of a solution. Appendix B describes a computer program (D3670) written for the Control Data series 6000 computer systems using the CDC 250 series CRT display system and the LRC 6000 series graphics system. The technique can be applied in a non-interactive mode using off-line plotting equipment.

Parametric Variables

Using explicit variables (x_i,y_i) where $y_i = f(x_i)$ and finding the spline fit $y = S_\Delta(x)$ as described in the previous sections required that on the mesh Δ, $\bar{x}_{j+1} > \bar{x}_j$. Consequently the described computer algorithm will not smooth arbitrary sets of data $\{(x_i,y_i)\}_{i=1}^{n}$ such as that shown in figure 2. If, however, a monotonic parametric

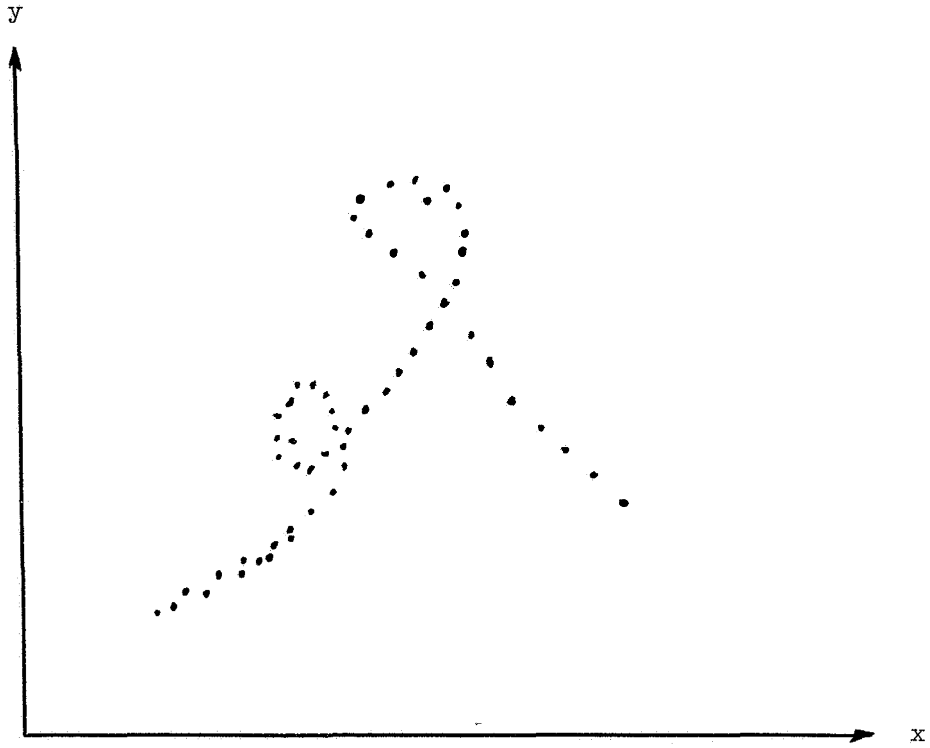


Figure 2.- Sample of data suitable for parametric fit.

variable t is introduced to the data set such that the set becomes $\{(t_i,x_i),(t_i,y_i)\}_{i=1}^{n}$ where $x_i = \bar{g}(t_i)$ and $y_i = \bar{h}(t_i)$, and choosing two meshes

$$\Delta_x : t_1 = \bar{t}_1 < \bar{t}_2 < . . . < \bar{t}_{m_1} = t_n$$

$$\Delta_y : t_1 = \hat{t}_1 < \hat{t}_2 < . . . < \hat{t}_{m_2} = t_n$$

the smoothing technique can be applied for arbitrary sets of data. A parametric variable which will satisfy the condition $t_{j+1} > t_j$ while $x_{j+1} \leqq x_j$ is

$$t_{i+1} = \left[(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2\right]^{1/2} + t_i$$

$$t_1 = 0$$

The accumulative chord length t will be monotonically increasing with respect to x and y provided $(x_{i+1},y_{i+1}) \neq (x_i,y_i)$. With the t's determined, there are two sets $\{(t_i,x_i)\}_{i=1}^{n}$ and $\{(t_i,y_i)\}_{i=1}^{n}$ upon which the algorithm described above can be applied. Applying this computational algorithm to the two sets yields the cubic spline functions

$S_{\Delta_x}(t)$ and $S_{\Delta_y}(t)$. For the purpose of interpolation, y can be found as a function of x by finding the inverse relation $t = \overline{S_{\Delta_x}(x)}$ and computing $y = S_{\Delta_y}\left[\overline{S_{\Delta_x}(x)}\right]$. The practical approach is to find t such that $x - S_{\Delta_x}(t) = 0$ in the interval $\left[t_j, t_{j+1}\right]$ and substituting this t into $y = S_{\Delta_y}(t)$. Since the cubic splines are being fitted with respect to a parametric variable, the first and second derivatives of y as a function of x are:

$$\frac{dy}{dx} = \frac{dy/dt}{dx/dt} = \frac{S'_{\Delta_y}(t)}{S'_{\Delta_x}(t)}$$

$$\frac{d^2y}{dx^2} = \frac{\frac{dx}{dt}\frac{d^2y}{dt^2} - \frac{d^2x}{dt^2}\frac{dy}{dt}}{(dx/dt)^3} = \frac{S'_{\Delta_x}(t)S''_{\Delta_y}(t) - S''_{\Delta_x}(t)S'_{\Delta_y}(t)}{\left[S'_{\Delta_x}(t)\right]^3}$$

For cases with data forming a closed curve, it is also desirable to have the spline curves continuous and equal in value at the endpoints. This is done by adding two additional constraints in equation (9), thus introducing two additional Lagrange multipliers in the least-squares solution. Enforcing the condition at the endpoints

$$y'\left(\bar{x}_1 +\right) = y'\left(\bar{x}_m -\right)$$

yields the continuity of the first derivative. Using equation (5) with 1 and m as indices establishes the above condition in the form

$$A'_1(\bar{x}_1)\bar{y}_1 + B'_1(\bar{x}_1)y''_1 + C'_1(\bar{x}_1)\bar{y}_2 + D'_1(\bar{x}_1)y''_2$$
$$- A'_{m-1}(\bar{x}_m)\bar{y}_{m-1} - B'_{m-1}(\bar{x}_m)y''_{m-1} - C'_{m-1}(\bar{x}_m)\bar{y}_m - D'_{m-1}(\bar{x}_m)y''_m = 0$$

Enforcing equality at the endpoints requires the additional constraint

$$\bar{y}_1 - \bar{y}_m = 0$$

A Computational Algorithm for Arbitrary Sets of Data $\left\{(x_i, y_i)\right\}_{i=1}^{n}$

Given the sets $\left\{(x_i, y_i)\right\}_{i=1}^{n}$ and $\left\{w_i\right\}_{i=1}^{n}$:

(1) Compute the set $\left\{t_i\right\}_{i=1}^{n}$ where $t_1 = 0$ and

$$t_{i+1} = \left[(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2\right]^{1/2} + t_i$$

(2) Form the sets $\{(t_i,x_i)\}_{i=1}^{n}$ and $\{(t_i,y_i)\}_{i=1}^{n}$.

(3) Choose a set $\Delta_x = \{\bar{t}_j\}_{j=2}^{m-1}$ where $\bar{t}_1 = t_1 = 0$ and $\bar{t}_{m_1} = t_n$.

(4) Form the matrices $\begin{bmatrix} F^TWF & G^T \\ G & 0 \end{bmatrix}_t$ and $\begin{bmatrix} F^TWY \\ 0 \end{bmatrix}_x$.

(5) Find the inverse of $\begin{bmatrix} F^TWF & G^T \\ G & 0 \end{bmatrix}_t$ and solve

$$\begin{bmatrix} X \\ \Lambda \end{bmatrix}_x = \begin{bmatrix} F^TWF & G^T \\ G & 0 \end{bmatrix}_t^{-1} \begin{bmatrix} F^TWF \\ 0 \end{bmatrix}_x$$

(6) Compute standard derivation in the (t,x) space

$$\sigma_x = \sum_{i=1}^{n} \left\{ \frac{\left[x_i - S_{\Delta_x}(t_i)\right]^2 w_i^2}{n - 2m_1} \right\}^{1/2}$$

(7) Plot $\{(t_i,x_i)\}_{i=1}^{n}$ and $S_{\Delta_x}(t)$.

(8) If the resulting fit of $S_{\Delta_x}(t)$ to the data $\{(t_i,x_i)\}_{i=1}^{n}$ is not satisfactory, choose a new set $\{\bar{t}_j\}_{j=2}^{m_1-1}$ and restart at (4). Otherwise, continue.

(9) Choose a set $\Delta_y = \{\hat{t}_j\}_{j=2}^{m_2-1}$ where $\hat{t}_1 = t_1$ and $\hat{t}_{m_2} = t_n$.

(10) Form the matrices $\begin{bmatrix} F^TWF & G^T \\ G & 0 \end{bmatrix}_t$ and $\begin{bmatrix} F^TWY \\ 0 \end{bmatrix}_y$.

(11) Find the inverse of $\begin{bmatrix} F^TWF & G \\ G & 0 \end{bmatrix}_t$ and solve

$$\begin{bmatrix} X \\ \Lambda \end{bmatrix}_y = \begin{bmatrix} F^TWF & G^T \\ G & 0 \end{bmatrix}_t^{-1} \begin{bmatrix} F^TWF \\ 0 \end{bmatrix}_y$$

(12) Compute standard derivation in the (t,y) space

$$\sigma_y = \left\{ \sum_{i=1}^{n} \frac{\left[y_i - S_{\Delta y}(t_i)\right]^2 w_i^2}{n - 2m_2} \right\}^{1/2}$$

(13) Plot $\left\{(t_i,y_i)\right\}_{i=1}^{n}$ and $S_{\Delta y}(t)$.

(14) If the resulting fit of $S_{\Delta y}(t)$ to the data $\left\{(t_i,y_i)\right\}_{i=1}^{n}$ is not satisfactory, choose a new set $\left\{\hat{t}_j\right\}_{j=2}^{m_2-1}$ and restart at (10). Otherwise, continue.

(15) Plot $\left\{(x_i,y_i)\right\}_{i=1}^{n}$ and $S_{\Delta y}\left(S_{\Delta x}(t)\right)$.

(16) If the resulting fit of $S_{\Delta y}\left(S_{\Delta x}(t)\right)$ to $\left\{(x_i,y_i)\right\}_{i=1}^{n}$ is not satisfactory, restart at (3). Repeat until one of the following conclusions is reached:

(a) The fit is satisfactory so that $S_{\Delta x}(t)$ and $S_{\Delta y}(t)$ can be used to represent $\left\{(x_i,y_i)\right\}_{i=1}^{n}$.

(b) The fit is unsatisfactory and the technique is abandoned.

Here again the technique is highly feasible in an interactive computer graphics environment. In appendix B, the described computer program has the option of using parametric variables in the interactive and off-line modes. In appendix C there is a description of example cases using both explicit and implicit variables.

CONCLUDING REMARKS

Two algorithms have been developed for smoothing sets of data in two-dimensional Cartesian coordinates. Both algorithms exploit the cubic spline function which is continuous and has continuous first and second derivatives. A least-squares technique is applied to the cubic spline function to obtain smooth representations of the data sets. The two algorithms, the explicit variable algorithm and the parametric variable algorithm, are distinguished according to how the independent and dependent variables are defined relative to each other. The explicit variable algorithm requires less computation and is, therefore, recommended whenever high gradients are not encountered. The parametric variable algorithm requires more computation and can be applied to arbitrary sets of data. Both algorithms require that the user specify a set of mesh points in the independent variable direction. Since this mesh usually must be modified in order to achieve satisfactory solutions, the algorithms are best applied in an interactive computer graphics environment.

Langley Research Center,
National Aeronautics and Space Administration,
Hampton, Va., November 1, 1973.

APPENDIX A

CONSTRAINED WEIGHTED LINEAR LEAST-SQUARES ESTIMATION

Given the data set $\left\{x_i, y_i, w_i\right\}_{i=1}^{n}$ and the function

$$y = y\left(x, \bar{y}_1, \bar{y}_1'', \ldots, \bar{y}_m, \bar{y}_m''\right) \qquad (2m < n)$$

a weighted least-squares solution is the set of estimates

$$\left\{\alpha_{2k-1} = \bar{y}_k, \alpha_{2k} = \bar{y}_k''\right\}_{k=1}^{m}$$

which minimizes the sum of the squares of the weighted differences between the corresponding elements of the sets $\left\{y_i\right\}_{i=1}^{n}$ and $\left\{y(x_i, \alpha_1, \alpha_2, \ldots, \alpha_l)\right\}_{i=1}^{n}$ (see ref. 6); that is,

$$\begin{aligned} \min_{\left\{\alpha_k\right\}_{k=1}^{l}} \Phi &= \min_{\left\{\alpha_k\right\}_{k=1}^{l}} \sum_{i=1}^{n} \left(w_i r_i\right)^2 \\ &= \min_{\left\{\alpha_k\right\}_{k=1}^{l}} \sum_{i=1}^{n} \left\{w_i \left[y_i - y(x_i, \alpha_1, \alpha_2, \ldots, \alpha_l)\right]\right\}^2 \qquad (l = 2m) \end{aligned} \tag{A1}$$

If there exists a set of functions

$$\left\{g_p(\alpha_1, \alpha_2, \ldots, \alpha_l)\right\}_{p=1}^{q} \qquad (l + q < n)$$

such that

$$\psi = \left\{g_p(\alpha_1, \alpha_2, \ldots, \alpha_l) = 0\right\}_{p=1}^{q}$$

then ψ defines a set of constraints which the function $y(x, \alpha_1, \alpha_2, \ldots, \alpha_l)$ must satisfy. The constrained least-squares problem consists of determining the set $\left\{\alpha_k\right\}_{k=1}^{l}$ which minimizes equation (A1) subject to the conditions described by ψ. This is accomplished by associating a set of Lagrange multipliers $\left\{\lambda_p\right\}_{p=1}^{q}$ with the set ψ and determining

$$\min_{\substack{\{\alpha_k\}_{k=1}^{l} \\ \{\lambda_p\}_{p=1}^{q}}} \Phi = \min_{\substack{\{\alpha_k\}_{k=1}^{l} \\ \{\lambda_p\}_{p=1}^{q}}} \left(\sum_{i=1}^{n} \left\{ w_i \left[y_i - y(x_i, \alpha_1, \alpha_2, . . ., \alpha_l) \right] \right\}^2 + \sum_{p=1}^{q} \lambda_p g_p \right) \tag{A2}$$

An additional mathematical assumption is that

$$y(x, \alpha_1, \alpha_2, . . ., \alpha_l)$$

and

$$\left\{ g_p(\alpha_1, \alpha_2, . . ., \alpha_l) \right\}_{p=1}^{q}$$

are linear functions with respect to the elements of the set $\{\alpha_k\}_{k=1}^{l}$ and, therefore, can be written in the form

$$y(x, \alpha_1, \alpha_2, . . ., \alpha_l) = \sum_{k=1}^{l} a_k(x)\alpha_k$$

$$g_p(\alpha_1, \alpha_2, . . ., \alpha_l) = \sum_{k=1}^{l} b_{pk}\alpha_k$$

With these definitions, the minimization problem for constrained weighted linear least-squares estimation can be rewritten in matrix notation as

$$\begin{aligned} \min_{X,\Lambda} \Phi &= \min_{X,\Lambda} \left[R^T W R + \Lambda^T B X \right] \\ &= \min_{X,\Lambda} \left[\left(Y^T - X^T A^T \right) W (Y - AX) + 2\Lambda^T B X \right] \end{aligned} \tag{A3}$$

where

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ \cdot \\ \cdot \\ \cdot \\ y_n \end{bmatrix} \qquad R = \begin{bmatrix} r_1 \\ r_2 \\ \cdot \\ \cdot \\ \cdot \\ r_n \end{bmatrix} \qquad A = \begin{bmatrix} a_1(x_1) & a_2(x_1) & . . . & a_l(x_1) \\ a_1(x_2) & a_2(x_2) & . . . & a_l(x_2) \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ a_1(x_n) & a_2(x_n) & . . . & a_l(x_n) \end{bmatrix}$$

$$X = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \cdot \\ \cdot \\ \cdot \\ \alpha_l \end{bmatrix} \qquad B = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1l} \\ b_{21} & b_{22} & \cdots & b_{2l} \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ b_{q1} & b_{q2} & \cdots & b_{ql} \end{bmatrix} \qquad \Lambda = \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \cdot \\ \cdot \\ \cdot \\ \lambda_q \end{bmatrix}$$

$$W = \begin{bmatrix} w_1^2 & & & & & & & 0 \\ & w_2^2 & & & & & & \\ & & \cdot & & & & & \\ & & & \cdot & & & & \\ & & & & w_i^2 & & & \\ & & & & & \cdot & & \\ & & & & & & \cdot & \\ 0 & & & & & & & w_n^2 \end{bmatrix}$$

To determine $\min\limits_{X,\Lambda} \Phi$, the first variation $\delta\Phi$ with respect to X and Λ must vanish; that is,

$$\delta\Phi = \left[-2\left(Y^TWA - X^TA^TWA\right) + \Lambda^TB\right]\delta X + 2\delta\Lambda^TBX = 0$$

Since $\delta X \neq 0$ and $\delta\Lambda \neq 0$,

$$-A^TWY + A^TWAX + B^T\Lambda = 0 \tag{A4a}$$

$$BX = 0 \tag{A4b}$$

This can be written in a single matrix equation as

$$\begin{bmatrix} A^TWA & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} X \\ \Lambda \end{bmatrix} = \begin{bmatrix} A^TWY \\ 0 \end{bmatrix} \tag{A5}$$

with the solution

$$\begin{bmatrix} X \\ \Lambda \end{bmatrix} = \begin{bmatrix} A^TWA & B^T \\ B & 0 \end{bmatrix}^{-1} \begin{bmatrix} A^TWY \\ 0 \end{bmatrix} \tag{A6}$$

A sufficient condition that X and Λ are unique $\left(\text{i.e., } \begin{bmatrix} A^TWA & B^T \\ B & 0 \end{bmatrix} \text{ can be inverted}\right)$ is that A^TWA is invertible and $B\left[A^TWA\right]^{-1} B^T$ is also invertible. This can be shown by solving equation (A4a) for X; that is,

$$X = \left[A^TWA\right]^{-1} A^TWY - \left[A^TWA\right]^{-1} B^T\Lambda \tag{A7}$$

and then multiplying this equation by B

$$BX = B\left[A^TWA\right]^{-1} A^TWY - B\left[A^TWA\right]^{-1} B^T\Lambda$$

From equation (A4b)

$$BX = 0$$

This implies

$$B\left[A^TWA\right]^{-1} A^TWY - B\left[A^TWA\right]^{-1} B^T\Lambda = 0$$

or

$$\Lambda = \left[B\left[A^TWA\right]^{-1} B^T\right]^{-1} B\left[A^TWA\right]^{-1} A^TWY \tag{A8}$$

The vector Λ has a unique solution if A^TWA is invertible and if $B\left[A^TWA\right]^{-1} B^T$ is invertible. If these conditions are satisfied for Λ, then X has a unique solution by substituting Λ into equation (A7).

In addition, a sufficient condition that solution of equations (A4) yields a unique global minimum of equation (A2) subject to the constraints $BX = 0$ is that $\delta X^TA^TWA\delta X > 0$ for all $\delta X > 0$ with $B\delta X = 0$ (ref. 7).

APPENDIX B

PROGRAM DESCRIPTION

General Discussion

The program, written in FORTRAN IV for the CDC 6000 series computers, consists of a main program and five subroutines. CalComp plotting routines and CDC 250 routines are used for CRT display. The program optionally can run in either the batch mode or on line with control from a CRT console.

The main program directs the solution procedure, either from preset instructions in the batch mode or according to commands from the CRT console in the on-line mode. In the on-line mode, instructions to the user are displayed to indicate what options are available at the various points in the program.

The following subroutines are called from the main program or its subprograms.

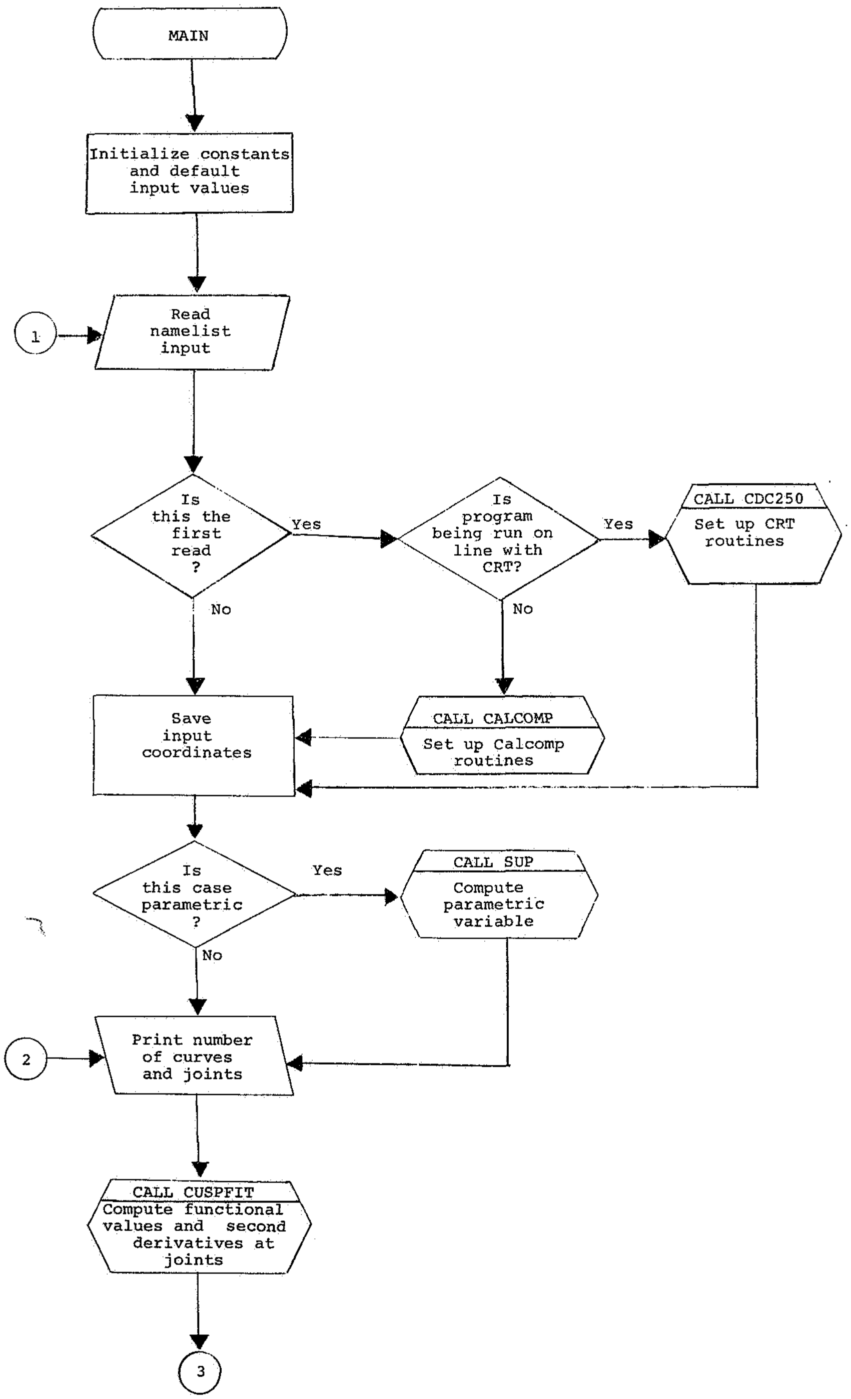
Subroutine	Function
PLPT	scales and plots computed values with the CalComp POINT routine
CUSPFIT	determines a cubic equation approximating the input data and computes the first and second derivative
SUP	computes the parametric variable t and sets up the arrays for subroutine CUSPFIT in cases using the parametric option of the program
MINMAX	finds the minimum and maximum values of the data to be plotted
SCALEBW	scales the values for plotting in the parametric cases
SIMEQ	solves the matrix equation AX = B where A is a square coefficient matrix and B is a matrix of constant vectors
FTLUP	performs a second-order interpolation to find intermediate values from a tabular array

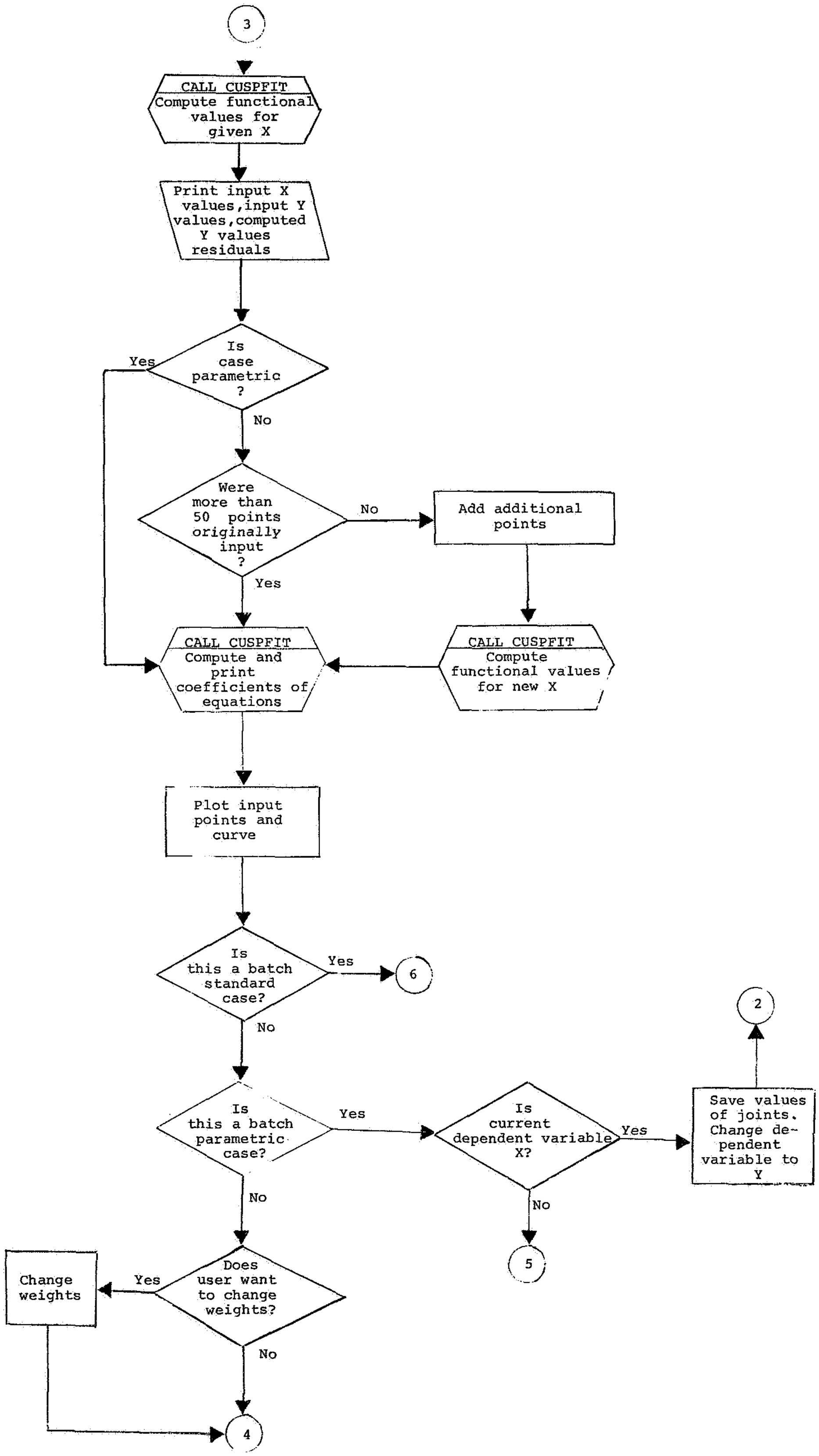
Interactive Graphics and Plotting Routines

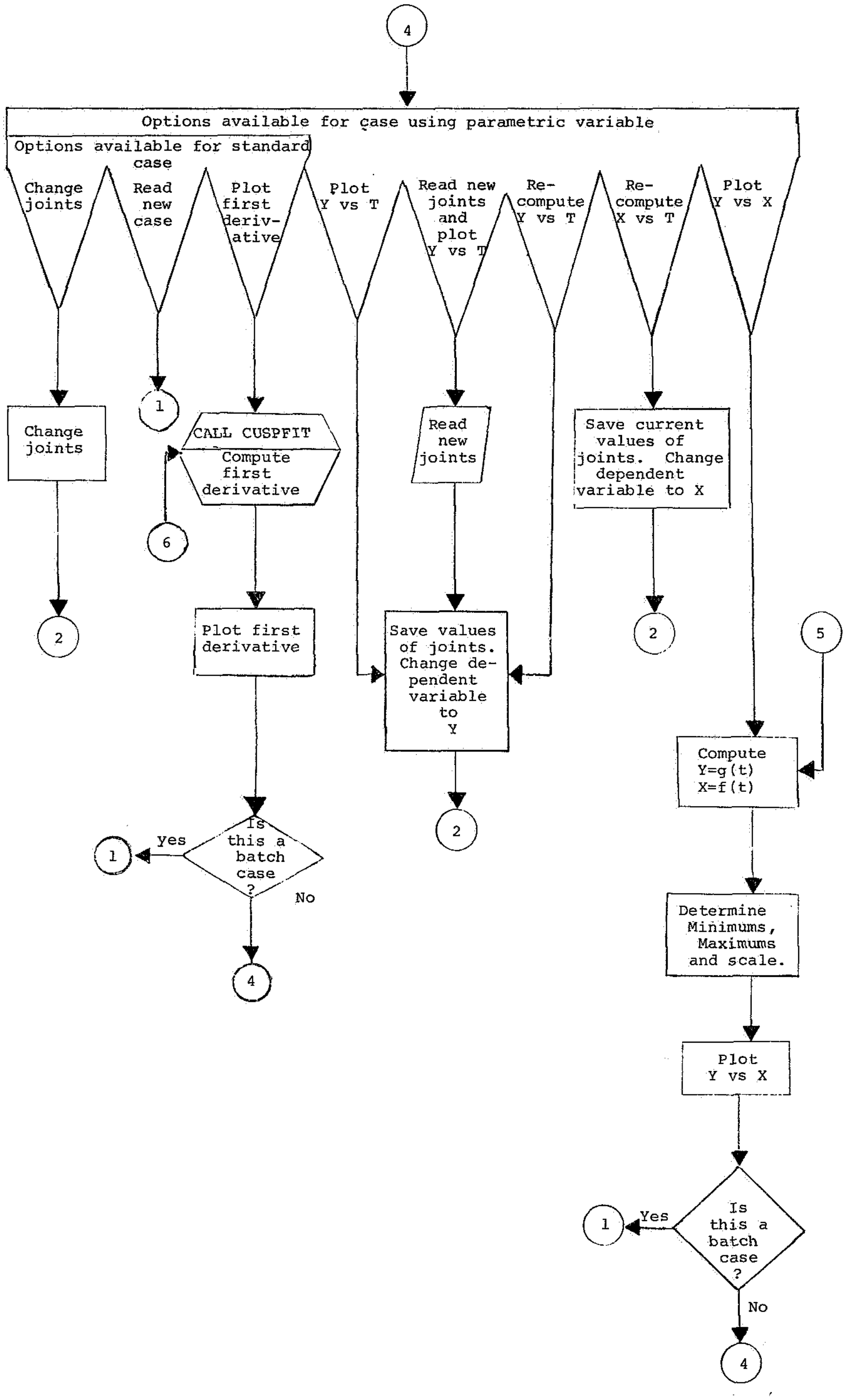
The CRT routines CDC 250, SCREEN, PARAMS, MESSAGE, and NEXT are from the LRC interactive graphics software package and the plotting routines CALCOMP, LEROY, ASCALE, AXES, LINPLT, LINE, NOTATE, NUMBER, CALPLT, and INFOPLT are from the CalComp software package. Plotter output is routed to a tape during job execution and after job completion is plotted on a CalComp digital incremental plotter. The arrays X, XC, Y, YY, YJOIN, COMPY, and R are used for plotting.

Descriptions, Flow Charts, and Listings of the Main Program and Subprograms

Main program.- The main program directs the solution procedure from preset instructions in the batch mode, or from CRT console commands in the on-line mode. The flow diagram of the main program is as follows:


MAIN
Initialize constants and default input values
1
Read namelist input
Is this the first read ?
Yes
No
Is program being run on line with CRT?
Yes
No
CALL CDC250
Set up CRT routines
CALL CALCOMP
Set up Calcomp routines
Save input coordinates
Is this case parametric ?
Yes
No
CALL SUP
Compute parametric variable
2
Print number of curves and joints
CALL CUSPFIT
Compute functional values and second derivatives at joints
3


3
CALL CUSPFIT
Compute functional values for given X
Print input X values, input Y values, computed Y values residuals
Is case parametric ?
Yes
No
Were more than 50 points originally input ?
No
Add additional points
Yes
CALL CUSPFIT
Compute and print coefficients of equations
CALL CUSPFIT
Compute functional values for new X
Plot input points and curve
Is this a batch standard case?
Yes
6
No
2
Is this a batch parametric case?
Yes
Is current dependent variable X?
Yes
Save values of joints. Change dependent variable to Y
No
5
No
Does user want to change weights?
Yes
Change weights
No
4


4
Options available for case using parametric variable
Options available for standard case
Change joints
Read new case
Plot first deriv-ative
Plot Y vs T
Read new joints and plot Y vs T
Re-compute Y vs T
Re-compute X vs T
Plot Y vs X
Change joints
1
CALL CUSPFIT
Compute first derivative
6
Read new joints
Save current values of joints. Change dependent variable to X
2
Plot first derivative
Save values of joints. Change de-pendent variable to Y
2
5
Compute Y=g(t) X=f(t)
2
Is this a batch case ?
yes
1
No
4
Determine Minimums, Maximums and scale.
Plot Y vs X
Is this a batch case ?
Yes
1
No
4

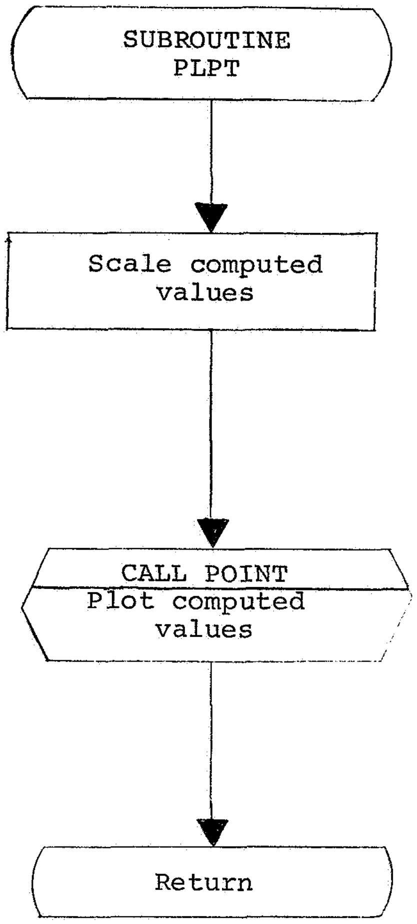
The program listing for the main program is as follows:

```
        PROGRAM MAIN(INPUT=201,OUTPUT=201,TAPE5=INPUT,TAPE6=OUTPUT)

*       X AND YY ARE COORDINATES OF THE INPUT DATA POINTS
*       R-LEFT ENDPOINT,JUNCTION POINTS,RIGHT ENDPOINT
*       NKR-NUMBER OF CURVES
*       NMAT- NUMBER OF POINTS
*       IFLAG =0  DATA POINT COORDINATES ARE READ WITH FORMAT
*       IFLAG =1  DATA POINT COORDINATES ARE READ WITH NAMELIST
*       W - ARRAY OF WEIGHTS OF DATA POINTS
*       KO = 1    STANDARD CRT
*       KO = 2    STANDARD BATCH
*       KO = 3    PARAMETRIC CRT
*       KO = 4    PARAMETRIC BATCH
*       KLOSE = 0    ARBITRARY CURVE
*       KLOSE = 1    CLOSED CURVE (MUST BE PARAMETRIC)

        REAL MAXRES
        DIMENSION TSAVE(310)
        DIMENSION CS(54,4), Z(310), T(310), RS(30), Q(30)
        DIMENSION XSAVE(310), YSAVE(310)
        COMMON XRAY(3),YRAY(3)
        COMMON X(310),Y(310),A(304,4),R(30),C(84,85),L(84),N(28),YY(310)
        COMMON COMPY(310),W(310),YJOIN(30),XC(310)
        NAMELIST /NAM1/ X,YY,NKR,R,NMAT,IFLAG,W,KO,KLOSE

*       INITIALIZE PLOT ROUTINE AND SET ORIGIN

        DATA NKR/1/,IFLAG/1/,KO/1/,NMAX/304/,KLOSE/0/
        IBEN=0
   1    CONTINUE
        R(1)=0.
        DO 2 I=1,310
   2    W(I)=1.
        IL=1
        WT=1.0
        XPG=10.
        YPG=10.
        XDV=10.
        NC=84
        XTIC=1.
        YDV=10.
        YTIC=1.
        IF (IBEN.EQ.0) GO TO 4
        DO 3 I=1,NMAT
   3    X(I)=XSAVE(I)
   4    CONTINUE
        READ (5,NAM1)
        IF (IBEN.EQ.1) GO TO 7
        IBEN=1
        GO TO (6,5,6,5), KO
   5    CALL CALCOMP
        CALL LEROY
        GO TO 7
   6    CALL CDC250

*       SETS PARAMETERS IN PLOT ROUTINE TO OUTPUT OR CHANGE AT CRT

        CALL SCREEN (1.,1.,.9)
        CALL PARAMS

*       THIS CLEARS THE PARAMETER TABLE

        CALL PARAMS (2LAN,AN,3LSTD,STD)
        CALL PARAMS (2LR1,R(1),2LR2,R(2),2LR3,R(3))
        CALL PARAMS (2LR4,R(4),2LR5,R(5),2LR6,R(6))
        CALL PARAMS (2LR7,R(7),2LR8,R(8),2LR9,R(9))
        CALL PARAMS (3LR10,R(10),3LR11,R(11))
```

```
      CALL PARAMS (3LR12,R(12),3LR13,R(13),3LR14,R(14))
      CALL PARAMS (3LR15,R(15),3LR16,R(16),3LR17,R(17))
      CALL PARAMS (3LR18,R(18),3LR19,R(19),3LR20,R(20))
      CALL PARAMS (3LR21,R(21))
      CALL PARAMS (2LA1,A1,2LWT,WT,1LI,I)
      CALL MESAGE (1,39HPROGRAM D3290 --FINDS BEST FIT FOR DATA,39)
      CALL MESAGE (1,32HUSE BEST FIT TO SOLVE PARAMETERS,32)
      CALL MESAGE (1,40HHIT KEY 46 TO READ DATA TO START PROGRAM,40)
      CALL MESAGE (1,22HHIT KEY 45 TO STOP JOB,22)
      CALL NEXT (NKEY)
      IF (NKEY.NE.46) GO TO 48
7     IF (KO.LT.3) KODE=4
      IF (KO.EQ.2) NKEY=49
      IF (KO.EQ.4) NKEY=51
      IF (IFLAG.EQ.0) GO TO 9
      DO 8 I=1,NMAT
      YSAVE(I)=YY(I)
8     XSAVE(I)=X(I)
9     IF (KO.GT.2) KODE=0
      AN=NKR
      NKR1=NKR+1
      IF (ENDFILE 5) 47,10
10    IF (IFLAG.NE.0) GO TO 16
      NMAT=0
11    NMAT=NMAT+1
      READ (5,75) X(NMAT),Y(NMAT)
      IF (X(NMAT).NE.111111.) GO TO 11
      GO TO (13,13,12,12), KO
12    CONTINUE
      CALL SUP (X,Y,T,Z,XMAXT,NMAT)
      XM=9H T VALUES
      YM=9H X VALUES
      GO TO 14
13    XM=9H X VALUES
      YM=9H Y VALUES
14    CONTINUE
      NMAT=NMAT-1
      DO 15 I=1,NMAT
15    YY(I)=Y(I)
      GO TO 21
16    GO TO (19,19,17,17), KO
17    CALL SUP (X,YY,T,Z,XMAXT,NMAT)
      XM=9H T VALUES
      YM=9H X VALUES
      DO 18 I=1,NMAT
18    TSAVE(I)=X(I)
19    DO 20 I=1,NMAT
20    Y(I)=YY(I)
      IF (KO.GE.3) GO TO 21
      XM=9H X VALUES
      YM=9H Y VALUES
21    NKR1=NKR+1
      IF (KO.GE.3) GO TO 23
      XO=X(1)
      DO 22 I=1,NMAT
22    X(I)=X(I)-XO
      R(1)=X(1)
23    CONTINUE
      IF (KODE.EQ.0) WRITE (6,74)
      IF (KODE.EQ.1) WRITE (6,81)
      IF (KODE.EQ.4) WRITE (6,88)
      R(NKR1)=X(NMAT)
      WRITE (6,76) NKR,NMAT,(R(I),I=1,NKR1)
      LL=NKR*2+2
      LR=LL+NKR
      LA=LR-1
      LO=LA
      CALL CUSPFIT (1,C,NKR,L,X,Y,R,NMAX,NC,N,A,W,KODE,CS,KLOSE)
      ISTUNT=N(NKR+1)
      CALL CUSPFIT (2,C,NKR,L,X,Y,R,NMAX,NC,N,A,W,KODE,CS,KLOSE)
      RES=0
      IF (KODE.EQ.0) WRITE (6,78)
      IF (KODE.EQ.1) WRITE (6,77)
      IF (KODE.EQ.4) WRITE (6,87)
      DO 24 I=1,ISTUNT
      COMPY(I)=Y(I)
      RESID=YY(I)-Y(I)
      IF (W(I).EQ.0.) RESID=0.
      WRITE (6,79) I,X(I),YY(I),COMPY(I),RESID
      RES=RESID**2*W(I)
   24 CONTINUE
      RES=RES/(ISTUNT-2*NKR)
      STD=SQRT(RES)
      PRINT 80, STD
      NO=ISTUNT
      IF (KO.GT.2) GO TO 28

*     PUT IN MORE POINTS IF ORIGINALLY LESS THAN 50 FOR STANDARD VERSION

      IF (NO.GT.50) GO TO 27
      NM1=NO-1
      NP=120/NM1
      NPM1=NP-1
      IC=1
      XC(1)=X(1)
      DO 26 II=1,NM1
      FAC=(X(II+1)-X(II))/NPM1
      DO 25 I=1,NPM1
      IC=IC+1
      XC(IC)=XC(IC-1)+FAC
   25 CONTINUE
   26 CONTINUE
      XC(IC)=R(NKR1)
      CALL CUSPFIT (2,C,NKR,L,XC,COMPY,R,NMAX,NC,N,A,W,KODE,CS,KLOSE)
   27 CONTINUE
   28 CALL CUSPFIT (5,C,NKR,L,X,Y,R,NMAX,NC,N,A,W,KODE,CS,KLOSE)
      K=1
      FNKR=NKR

*     COMPUTE MINIMUMS AND MAXIMUMS

      CALL ASCALE (X,XPG,NO,K,10.)
      CALL ASCALE (YY,YPG,NO,K,10.)

*     DRAW X AXIS

      NP1=NO+1
      NP2=NO+2
      CALL AXES (0.,0.,0.,XPG,X(NP1),X(NP2),XTIC,XDV,XM,.15,-9)

*     DRAW Y AXIS

      CALL AXES (0.,0.,90.,YPG,YY(NP1),YY(NP2),YTIC,YDV,YM,.15,9)

*     PLOT CURVE

      CALL PLPT (X,YY,NO)
      NP=IC
      IF (KO.LT.3.AND.NO.LT.50) GO TO 30
      NP=NO
      DO 29 I=1,NO
   29 XC(I)=X(I)
   30 NN1=NP+1
      NN2=NP+2
      XC(NN1)=X(NP1)
      XC(NN2)=X(NP2)
      COMPY(NN1)=YY(NP1)
      COMPY(NN2)=YY(NP2)
      CALL LINPLT (XC,COMPY,NP,K,0,0,0,0)
      NK11=NKR1+1
      NK12=NKR1+2
      R(NK11)=X(NP1)
      R(NK12)=X(NP2)
      YJOIN(NK11)=YY(NP1)
      YJOIN(NK12)=YY(NP2)
      YJOIN(1)=COMPY(1)
      DO 31 I=2,NKR1
   31 CALL FTLUP (R(I),YJOIN(I),1,NP,XC,COMPY)
      CALL LINE (R,YJOIN,NKR1,K,-1,5,.5)
      CALL NOTATE (1.,9.5,.14,19HNUMBER OF CURVES = ,0.,19)
      CALL NUMBER (4.,9.5,.14,FNKR,0.,-1)
      CALL NOTATE (5.,9.5,.14,32HENDPOINTS AND JUNCTION POINTS = ,0.,32)
      XN=0.
      YNN=9.0
      DO 32 I=1,NKR1
      XN=XN+.7
      IF (I.NE.14) GO TO 32
      YNN=8.5
      XN=.7
   32 CALL NUMBER (XN,YNN,.11,R(I),0.,4)

*     ESTABLISH A NEW REFERENCE POINT FOR THE NEXT GRAPH

      CALL CALPLT (0.,0.,-3)
      GO TO (33,34,33,34), KO
   33 CONTINUE

*     THIS WILL STOP PROGRAM

   34 CONTINUE
      CALL CALPLT (12.,0.,-3)
      IF (KO.EQ.2) GO TO 49
      IF (IL.EQ.2) CALL BALLPT
      IL=1
      IF (KO.EQ.4) GO TO 41
      CALL MESAGE (1,44HK45-CHANGE WEIGHTS     K46-SET ALL WEIGHTS=1,44)
      CALL MESAGE (1,23HHIT ANY KEY TO CONTINUE,23)
      CALL NEXT (NKEY)
      IF (NKEY.EQ.45) GO TO 36
      IF (NKEY.NE.46) GO TO 39
      DO 35 I=1,NO
   35 W(I)=1.0
      GO TO 39

*     LOOK AT EACH POINT AND WEIGHT THEM IF YOU WANT TO

   36 CALL PLPT (X,YY,NO)
      XRAY(2)=X(NP1)
      XRAY(3)=X(NP2)
      YRAY(2)=YY(NP1)
      YRAY(3)=YY(NP2)
      A11=0.
      A1=0.
      DO 37 II=1,NO
      A11=A11+A1
      I=II+A11
      A1=0.
      XRAY(1)=X(I)
      YRAY(1)=YY(I)
      CALL LINE (XRAY,YRAY,1,K,-1,4,.09)
      CALL CALPLT (0.,0.,-3)
      CALL MESAGE (1,48HANY K CONTINUE   K45-CHANGE WT   K46-STOP SEARCH
     1,48)
      CALL MESAGE (1,35HWT = NEW WEIGHT    A1 = SKIP POINTS,35)
      CALL NEXT (NKEY)
      IF (NKEY.EQ.46) GO TO 38
      IF (NKEY.NE.45) GO TO 37
```

```
*     CHANGE WEIGHT OF POINT

      W(I)=WT
   37 CONTINUE
   38 CALL CALPLT (12.,0.,-3)
   39 CONTINUE
      CALL MESAGE (1,42HK45-TYPE NEW R    K46-NEW CASE    K47-STOP,42)
      CALL MESAGE (1,30HK-49 TO PLOT FIRST DERIVATIVES,30)
      IF (KO.NE.3) GO TO 40
      CALL MESAGE (1,19HK-44 TO PLOT Y VS T,19)
      CALL MESAGE (1,19HK-50 TO PLOT Y VS X,19)
      CALL MESAGE (1,34HK-51 TO READ NEW R AND PLOT Y VS T,34)
      CALL MESAGE (1,25HK-52 TO RE-COMPUTE X VS T,25)
      CALL MESAGE (1,25HK-53 TO RE-COMPUTE Y VS T,25)

*     THIS STOPS PROGRAM

   40 CALL NEXT (NKEY)
   41 NKR=AN
      IF (KODE.NE.1) GO TO 43
      N1=NKR1
      NK=NKR
      DO 42 I=1,NKR1
   42 RS(I)=R(I)
   43 CONTINUE
      X(ISTUNT+1)=R(NKR+1)+1.
      IF (NKEY.EQ.45) GO TO 19
      IF (NKEY.EQ.44) GO TO 50
      IF (NKEY.EQ.50) GO TO 57
      IF (NKEY.EQ.51) GO TO 50
      IF (NKEY.EQ.46) GO TO 1
      IF (NKEY.EQ.52) GO TO 44
      IF (NKEY.EQ.53) GO TO 50
      IF (NKEY.EQ.49) GO TO 49
      IF (NKEY.EQ.47) STOP
      GO TO 39

*         RE-COMPUTE X VS. T

   44 DO 45 I=1,NMAT
      X(I)=TSAVE(I)
      YY(I)=XSAVE(I)
      T(I)=TSAVE(I)
   45 Z(I)=YSAVE(I)
      XM=9H T VALUES
      YM=9H X VALUES
      IF (KO.GT.2) KODE=0
      DO 46 I=1,NKRS
   46 R(I)=Q(I)
      NK=NKR
      N1=NKR1
      NKR=NKS
      NKR1=NKRS
      GO TO 19
   47 CALL CALPLT (0,0,999)
   48 STOP 1

*     PLOT FIRST DERIVATIVES

   49 CALL CUSPFIT (3,C,NKR,L,XC,COMPY,R,NMAX,NC,N,A,W,KODE,CS,KLOSE)
      CALL ASCALE (XC,XPG,NP,K,10.)
      CALL ASCALE (COMPY,YPG,NP,K,10)
      CALL AXES (0.,0.,0.,XPG,XC(NN1),XC(NN2),XTIC,XDV,XM,.15,-9)
      CALL AXES (0.,0.,90.,YPG,COMPY(NN1),COMPY(NN2),YTIC,YDV,YM,.15,9)
      CALL LINPLT (XC,COMPY,NP,K,0,0,0,0)
      CALL CUSPFIT (3,C,NKR,L,X,COMPY,R,NMAX,NC,N,A,W,KODE,CS,KLOSE)
      WRITE (6,72)
      WRITE (6,73) (X(I),YY(I),COMPY(I),I=1,NO)
      CALL CALPLT (12.,0.,-3)                                           A 312  34400000
      IF (KO.EQ.2) GO TO 1                                              A 313  34500000
      GO TO 39                                                          A 314  34600000
   50 DO 51 I=1,NMAT                                                    A 315  34700000
                                                                               34800000
*         COMPUTE Y VS.T                                                A 316  34900000
                                                                               35000000
   51 YY(I)=Z(I)                                                        A 317  35100000
      YM=9H Y VALUES                                                    A 318  35200000
      XM=9H T VALUES                                                    A 319  35300000
      IF (KO.GT.2) KODE=1                                               A 320  35400000
      IF (NKEY.EQ.53) GO TO 54                                          A 321  35500000
      DO 52 I=1,NKR1                                                    A 322  35600000
   52 Q(I)=R(I)                                                         A 323  35700000
      NKRS=NKR1                                                         A 324  35800000
      NKS=NKR                                                           A 325  35900000
      IF (NKEY.EQ.51) READ (5,NAM1)                                     A 326  36000000
      IF (ENDFILE 5) 47,53                                              A 327  36100000
   53 AN=NKR                                                            A 328  36200000
      IF (KO.EQ.4) NKEY=50                                              A 329  36300000
      IF (NKEY.NE.53) GO TO 19                                          A 330  36400000
   54 DO 55 I=1,N1                                                      A 331  36500000
   55 R(I)=RS(I)                                                        A 332  36600000
      DO 56 I=1,NMAT                                                    A 333  36700000
      X(I)=TSAVE(I)                                                     A 334  36800000
   56 YY(I)=YSAVE(I)                                                    A 335  36900000
      NKR=NK                                                            A 336  37000000
      NKR1=N1                                                           A 337  37100000
      GO TO 19                                                          A 338  37200000
                                                                               37300000
*         COMPUTE X VS.Y                                                A 339  37400000
                                                                               37500000
   57 T(1)=0.                                                           A 340  37600000
      XM=9H X VALUES                                                    A 341  37700000
      DT=XMAXT/300.                                                     A 342  37800000
      DO 58 I=1,N1                                                      A 343  37900000
   58 R(I)=RS(I)                                                        A 344  38000000
      NKR1=N1                                                           A 345  38100000
      NKR=NK                                                            A 346  38200000
      DO 61 I=2,301                                                     A 347  38300000
      T(I)=T(I-1)+DT                                                    A 348  38400000
      DO 59 J=1,NKR1                                                    A 349  38500000
      JJ=NKR1-J+1                                                       A 350  38600000
      JI=JJ                                                             A 351  38700000
      IF (T(I).GE.R(JJ)) GO TO 60                                       A 352  38800000
   59 CONTINUE                                                          A 353  38900000
   60 IF (JI.GT.NKR) JI=NKR                                             A 354  39000000
   61 Y(I)=((CS(JI+27,1)*T(I)+CS(JI+27,2))*T(I)+CS(JI+27,3))*T(I)+CS(JI+ A 355  39100000
     127,4)                                                             A 356  39200000
      DO 64 I=2,301                                                     A 357  39300000
      DO 62 J=1,NKRS                                                    A 358  39400000
      JJ=NKRS-J+1                                                       A 359  39500000
      JI=JJ                                                             A 360  39600000
      IF (T(I).GE.Q(JJ)) GO TO 63                                       A 361  39700000
   62 CONTINUE                                                          A 362  39800000
   63 IF (JI.GT.NKS) JI=NKS                                             A 363  39900000
      X(I)=((CS(JI,1)*T(I)+CS(JI,2))*T(I)+CS(JI,3))*T(I)+CS(JI,4)       A 364  40000000
   64 CONTINUE                                                          A 365  40100000
      X(1)=CS(1,4)                                                      A 366  40200000
      Y(1)=CS(28,4)                                                     A 367  40300000
      WRITE (6,82)                                                      A 368  40400000
      WRITE (6,83) (I,T(I),X(I),Y(I),I=1,301)                           A 369  40500000
      CALL MINMAX (X,301,AMIN,AMAX)                                     A 370  40600000
      CALL MINMAX (Y,301,BMIN,BMAX)                                     A 371  40700000
      IF (AMIN.GT.BMIN) AMIN=BMIN                                       A 372  40800000
      IF (AMAX.LT.BMAX) AMAX=BMAX                                       A 373  40900000
      CALL SCALEBW (AMIN,AMAX)                                          A 374  41000000
      XSAVE(NMAT+1)=AMIN                                                A 375  41100000
      YSAVE(NMAT+1)=AMIN                                                A 376  41200000
      SF=(AMAX-AMIN)/10.                                                A 377  41300000
      XSAVE(NMAT+2)=SF
      YSAVE(NMAT+2)=SF
      CALL PLPT (XSAVE,YSAVE,NMAT)
      CALL INFOPLT (1,301,X,1,Y,1,AMIN,AMAX,AMIN,AMAX,.5,9,XM,9,YM,0)
      IF (KO.EQ.4) NKEY=46

*            COMPUTE RESIDUALS

      DO 67 I=2,NMAT
      DO 65 J=1,NKR1
      JJ=NKR1-J+1
      JI=JJ
      IF (TSAVE(I).GE.R(JJ)) GO TO 66
   65 CONTINUE
   66 IF (JI.GT.NKR) JI=NKR
   67 Y(I)=((CS(JI+27,1)*TSAVE(I)+CS(JI+27,2))*TSAVE(I)+CS(JI+27,3))*TSA
     1VE(I)+CS(JI+27,4)
      DO 70 I=2,NMAT
      DO 68 J=1,NKRS
      JJ=NKRS-J+1
      JI=JJ
      IF (TSAVE(I).GE.Q(JJ)) GO TO 69
   68 CONTINUE
   69 IF (JI.GT.NKS) JI=NKS
      X(I)=((CS(JI,1)*TSAVE(I)+CS(JI,2))*TSAVE(I)+CS(JI,3))*TSAVE(I)+CS(
     1JI,4)
   70 CONTINUE
      X(1)=CS(1,4)
      Y(1)=CS(28,4)
      MAXRES=0.
      DO 71 I=1,NMAT
      Z(I)=ABS(X(I)-XSAVE(I))
      T(I)=ABS(Y(I)-YSAVE(I))
      IF (Z(I).GT.MAXRES) MAXRES=Z(I)
      IF (T(I).GT.MAXRES) MAXRES=T(I)
   71 CONTINUE
      WRITE (6,84)
      WRITE (6,85) (I,TSAVE(I),XSAVE(I),YSAVE(I),X(I),Y(I),Z(I),T(I),I=1
     1,NMAT)
      WRITE (6,86) MAXRES
      GO TO (39,1,39,1), KO

*

   72 FORMAT (//9X,1HX,14X,1HY,11X,4HYDOT)
   73 FORMAT (3E15.6)
   74 FORMAT (1H1,10X,,15H***************,/11X,15HDATA FOR X VS T/11X,15
     1H***************,//)
   75 FORMAT (40X,F7.3,3X,F7.3)
   76 FORMAT (1X,18HNUMBER OF CURVES =,I4,10X,18HNUMBER OF POINTS =,I4/3
     12H ENDPOINTS AND JUNCTION POINTS =/(6E20.7))HENDPOINTSANDJUNCTIONP
     2OINTS=/(6E20.7))
   77 FORMAT (1H0,12X,1HT,19X,1HY,15X,10HCOMPUTED Y,10X,9HRESIDUALS)
   78 FORMAT (1H0,12X,1HT,19X,1HX,15X,10HCOMPUTED X,10X,9HRESIDUALS)
   79 FORMAT (I4,E16.7,3E20.7)
   80 FORMAT (/8X,19HSTANDARD DEVIATION=,E12.5)
   81 FORMAT (1H1,10X,,15H***************,/11X,15HDATA FOR Y VS T/11X,15
     1H***************,//)
   82 FORMAT (//3X,1H ,18X,1HT,20X,1HX,20X,1HY,/)
   83 FORMAT (I5,3E20.8)
   84 FORMAT (5H1    ,8X,1HT,9X,1HX,9X,41HY COMPUTED X COMPUTED Y  RES X
     1      RES Y /)
   85 FORMAT (I5,7E10.2)
   86 FORMAT (//19H MAXIMUM RESIDUAL =,E20.7)
   87 FORMAT (1H0,12X,1HX,19X,1HY,15X,10HCOMPUTED Y,10X,9HRESIDUALS)
   88 FORMAT (1H1,10X,,15H***************,/11X,15HDATA FOR X VS Y/11X,15
     1H***************,//)
      END
```

Subroutine PLPT.- Subroutine PLPT scales and plots computed values with the CalComp POINT routine. The flow diagram for subroutine PLPT is as follows:



The program listing for subroutine PLPT is as follows:

```
      SUBROUTINE PLPT (X,YY,NO)
      DIMENSION X(1), YY(1)
      NP1=NO+1
      NP2=NO+2

*     SCALE AND PLOT COMPUTED VALUES WITH POINT ROUTINE

      XMV=X(NP1)
      XSF=X(NP2)
      YMV=YY(NP1)
      YSF=YY(NP2)
      DO 1 I=1,NO
      X1=(X(I)-XMV)/XSF
      Y1=(YY(I)-YMV)/YSF
      CALL POINT (X1,Y1)
      RETURN
      END
```

Subroutine CUSPFIT.- Subroutine CUSPFIT applies the least-squares technique described in this report to smooth data using cubic spline functions. The routine obtains (1) the value of the cubic spline function and the second derivative at the endpoints and junction points, (2) the functional values and values of the first and second derivative for a given x, and (3) the coefficients for each segment of the cubic spline function. In the flow diagram below, ICODE is a code which specifies the purpose of the current entry into the subroutine. ICODE is defined as follows:

ICODE = 1 Computes second derivatives and functional values at endpoints and junction points.

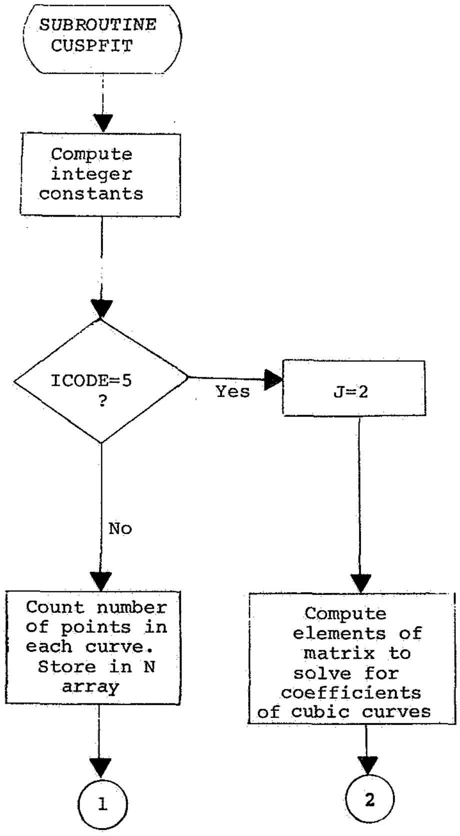
= 2 Computes functional values for given x.

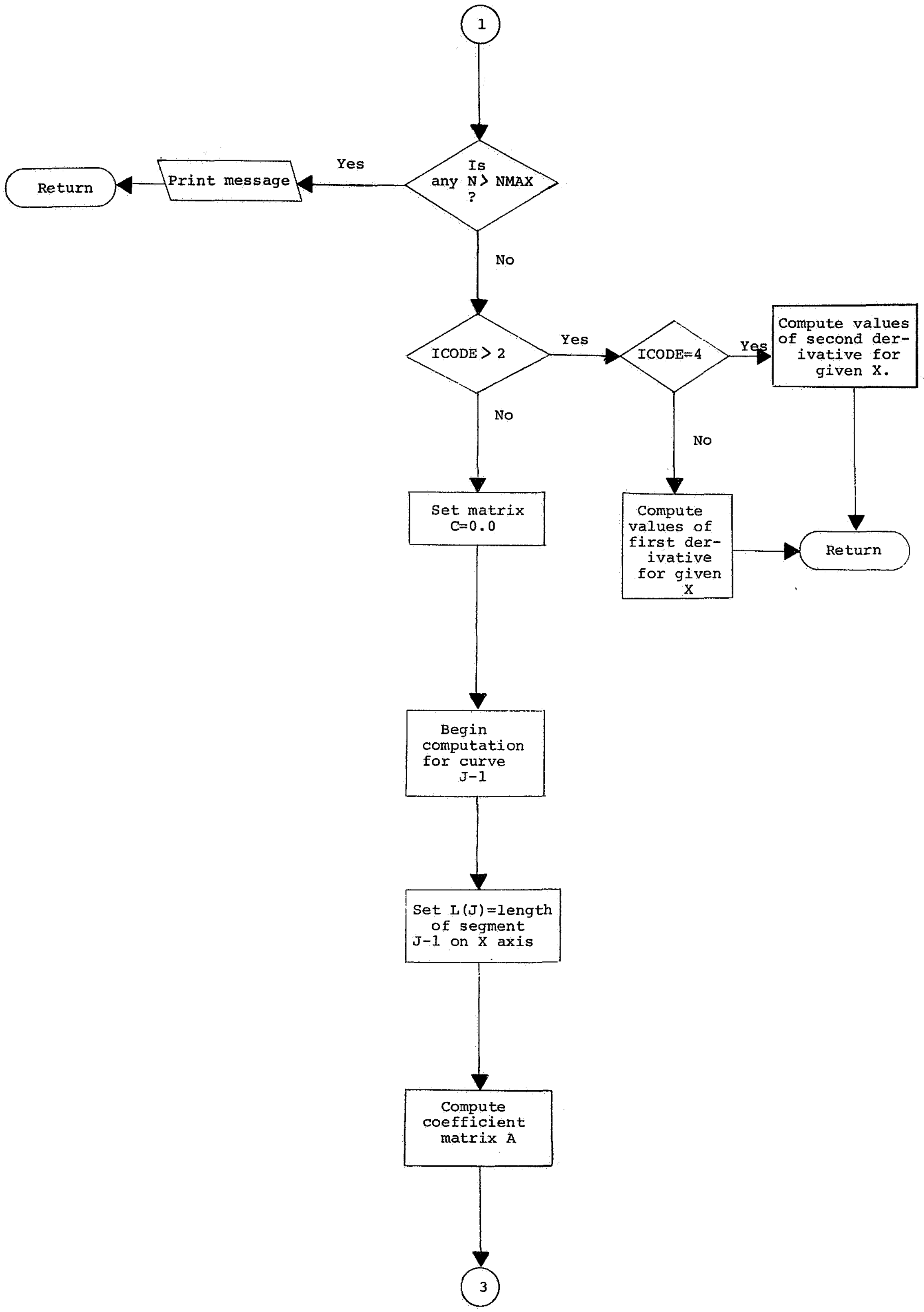
= 3 Computes the first derivative for given x.

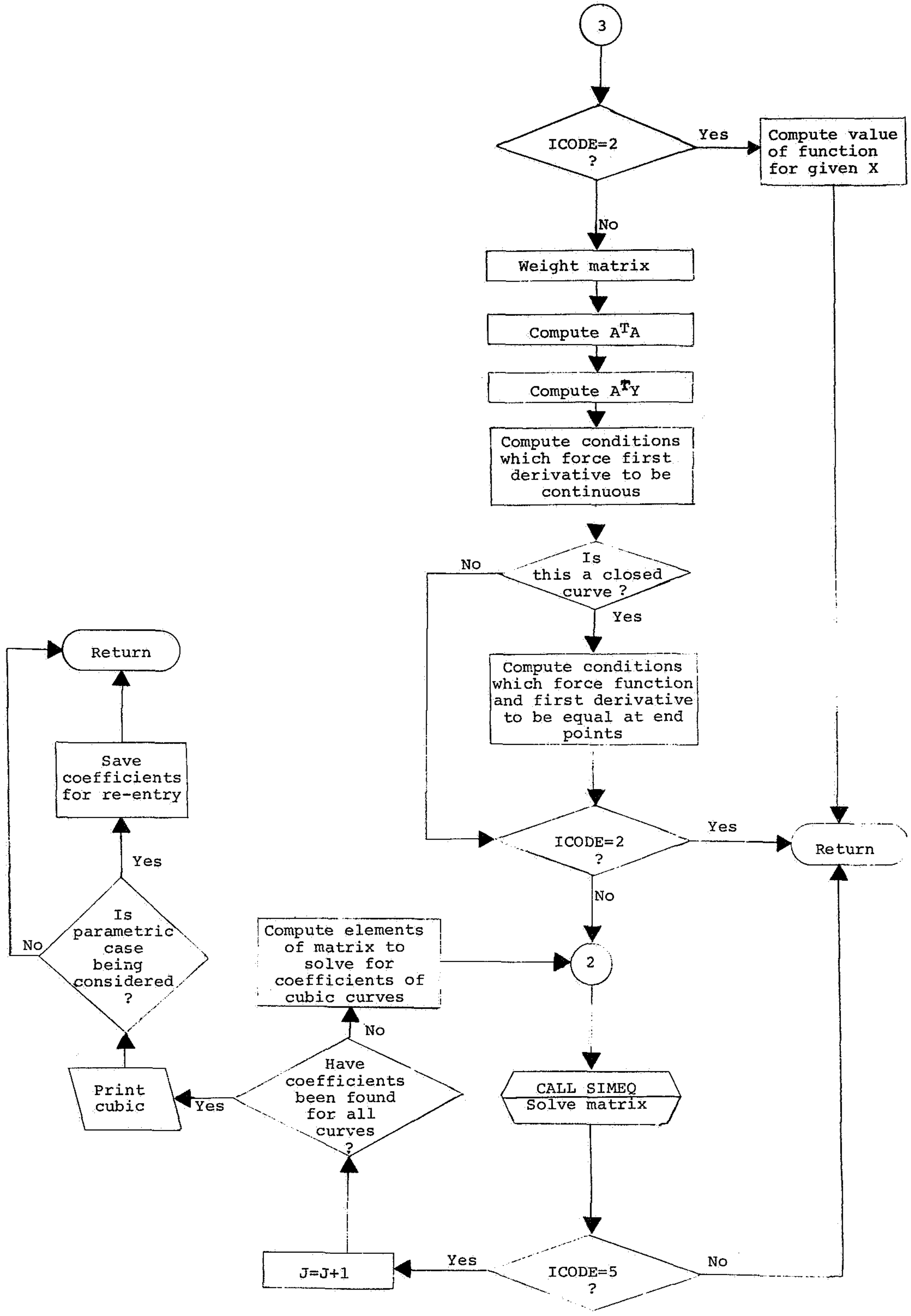
= 4 Computes the second derivative for given x.

= 5 Computes coefficients of cubic equations of the form $Y = Ax^3 + Bx^2 + Cx + D$

The flow diagram of subroutine CUSPFIT is as follows:


SUBROUTINE CUSPFIT
Compute integer constants
ICODE=5 ?
Yes
J=2
No
Count number of points in each curve. Store in N array
Compute elements of matrix to solve for coefficients of cubic curves
1
2


1
Is
any N > NMAX
?
Yes
Print message
Return
No
ICODE > 2
Yes
ICODE=4
Yes
Compute values of second derivative for given X.
No
No
Set matrix C=0.0
Compute values of first derivative for given X
Return
Begin computation for curve J-1
Set L(J)=length of segment J-1 on X axis
Compute coefficient matrix A
3

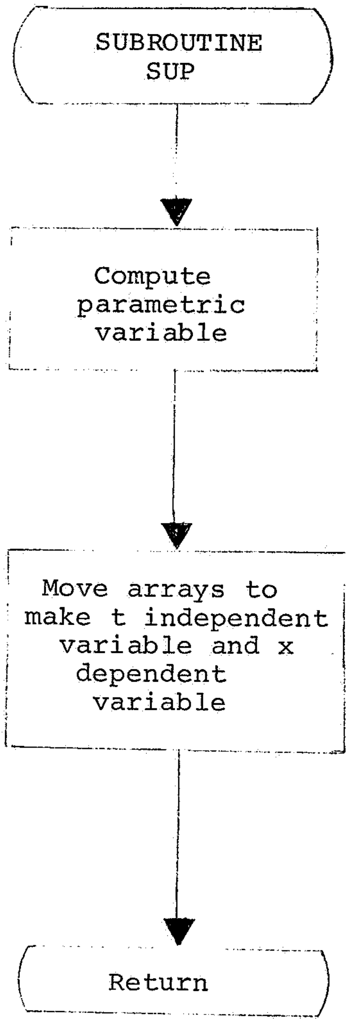

3
ICODE=2
?
Yes
Compute value of function for given X
No
Weight matrix
Compute A^TA
Compute A^TY
Compute conditions which force first derivative to be continuous
Is this a closed curve ?
No
Yes
Compute conditions which force function and first derivative to be equal at end points
ICODE=2
?
Yes
Return
No
2
CALL SIMEQ
Solve matrix
ICODE=5
?
No
Yes
J=J+1
Have coefficients been found for all curves
?
No
Compute elements of matrix to solve for coefficients of cubic curves
Yes
Print cubic
Is parametric case being considered
?
Yes
No
Save coefficients for re-entry
Return

The program listing for subroutine CUSPFIT is as follows:

```
      SUBROUTINE CUSPFIT (ICODE,C,NKR,L,X,Y,R,NMAX,NC,N,A,W,KODE,CS,KLOS
     1E)

*                          FIELD DEFINITIONS
*     ICODE=1   COMPUTES SECOND DERIVATIVE AND Y VALUES AT R VALUES
*     ICODE=2   COMPUTES THE VALUE OF THE FUNCTION FOR GIVEN X VALUES
*     ICODE=3   COMPUTES THE VALUE OF THE 1ST DERIVATIVE FOR VALUES OF X
*     ICODE=4   COMPUTES THE VALUE OF THE 2ND DERIVATIVE FOR VALUES OF X
*     ICODE=5   COMPUTES AN EQUATION OF THE FORM Y=AX**3+BX**2+CX+D
*     X(I) AND Y(I) ARE THE CO-ORDINATES OF THE DATA POINTS
*     R(J) IS THE VALUE OF A JUNCTION POINT OR AN END POINT
*     NMAX IS THE MAXIMUM NUMBER OF POINTS PER CURVE
*     LO=3*NUMBER OF CURVES+1
*     A IS THE COEFFICIENT MATRIX
*     C IS THE MATRIX WHICH REPRESENTS THE EQUATION
*         T    T         T
*      (A  A   B )(X)   (A Y)
*      (         )( ) = (    )
*      ( B     0 )(K)   ( 0 )
*     K IS THE LAGRANGIAN MULTIPLIER
*     L(1) ERROR CODE
*     L(J+1) IS THE LENGTH ALONG THE X-AXIS OF THE JTH SEGMENT
*     N(J) IS THE NUMBER OF POINTS PER CURVE  N(NUMBER OF CURVES + 1)=
*     THE TOTAL NUMBER OF POINTS

      DIMENSION CS(54,4)
      DIMENSION A(NMAX,1), C(NC,1), N(1), R(1), L(1), X(1), Y(1), IPIVOT
     1(1), W(1)
      REAL L
      JJ=0
      L(1)=0.
      LO=3*NKR+1
      IF (KLOSE.EQ.1) LO=LO+2
      LA=LO
      LR=LO+1
      LL=LR-NKR
      IF (KLOSE.EQ.1) LL=LL-2
      NKR1=NKR+1
      IF (ICODE.EQ.5) GO TO 30
      J=2
      ISTUNT=0
    1 ICOUNT=0
    2 ICOUNT=ICOUNT+1
      ISTUNT=ISTUNT+1
      IF (X(ISTUNT)-R(J)) 2,3,4
    3 N(J-1)=ICOUNT
      GO TO 5
    4 N(J-1)=ICOUNT-1
      ISTUNT=ISTUNT-1
    5 IF (N(J-1)-NMAX) 7,7,6
    6 JI=J-1
      L(1)=2.
      PRINT 40, N(JI),JI
      GO TO 39
    7 J=J+1
      IF (J-NKR1) 1,1,8
    8 N(NKR+1)=ISTUNT
      IF (ICODE-2) 9,11,24
    9 DO 10 J=1,LR
      DO 10 I=1,LO
      C(I,J)=0.0
   10 CONTINUE
```

```
*     COMPUTES COEFFICIENT MATRIX(A)

   11 IN=0
      IM=1
      II=0
      DO 20 J=2,NKR1
      IK=N(J-1)
      IF (IK.EQ.0) GO TO 19
      L(J)=R(J)-R(J-1)
      IN=IK+IN
      DO 12 I=IM,IN
      K=I-IM+1
      XI=X(I)-R(J)
      XIM1=X(I)-R(J-1)
      A(K,1)=(-XI**3/(6.*L(J))+L(J)*XIM1/6.-L(J)**2/6.)
      A(K,2)=(1.-XIM1/L(J))
      A(K,3)=(XIM1**3/(6.0*L(J))-L(J)*XIM1/6.)
      A(K,4)=XIM1/L(J)
   12 CONTINUE
      IF (ICODE.EQ.2) GO TO 22

*     WEIGHT POINTS    W * A * X = W * Y


      DO 13 I=IM,IN
      K=I-IM+1
      A(K,1)=A(K,1)*W(I)
      A(K,2)=A(K,2)*W(I)
      A(K,3)=A(K,3)*W(I)
      A(K,4)=A(K,4)*W(I)
   13 Y(I)=W(I)*Y(I)

*     COMPUTES A TRANSPOSE * A

      KK=0
      IL=1
      DO 16 K=1,4
      KK=K+II
      DO 15 M=IL,4
      MM=M+II
      DO 14 I=1,IK
      C(KK,MM)=A(I,K)*A(I,M)+C(KK,MM)
   14 CONTINUE
      C(MM,KK)=C(KK,MM)
   15 CONTINUE
      IL=IL+1
   16 CONTINUE

*     COMPUTES A TRANSPOSE * Y

      DO 17 M=1,4
      MM=M+II
      DO 17 I=IM,IN
      K=I-IM+1
      C(MM,LR)=A(K,M)*Y(I)+C(MM,LR)
   17 CONTINUE
      IF (J.EQ.2) GO TO 19

*     COMPUTES CONDITIONS FORCING THE FIRST DERIVATIVE TO BE CONTINUOUS

      J=J-1
      LLJ=LL+J-1
      K=II-2
      C(LLJ,1+K)=L(J)/6.
      C(LLJ,2+K)=-1./L(J)
      C(LLJ,3+K)=(L(J)+L(J+1))/3.
      C(LLJ,4+K)=(L(J)+L(J+1))/(L(J+1)*L(J))
      C(LLJ,5+K)=L(J+1)/6.
      C(LLJ,6+K)=-1./L(J+1)
      C(1+K,LLJ)=C(LLJ,1+K)
      C(2+K,LLJ)=C(LLJ,2+K)
      C(3+K,LLJ)=C(LLJ,3+K)
      C(4+K,LLJ)=C(LLJ,4+K)
      C(5+K,LLJ)=C(LLJ,5+K)
      C(6+K,LLJ)=C(LLJ,6+K)
      IF (KLOSE.EQ.0) GO TO 18
      IF (J.NE.NKR1-1) GO TO 18

*     FORCE EQUALITY OF THE FIRST DERIVATIVES AT THE ENDPOINTS

      LLJ=LLJ+1
      C(LLJ,3+K)=-L(J+1)/6.
      C(LLJ,4+K)=1./L(J+1)
      C(LLJ,5+K)=-L(J+1)/3.
      C(LLJ,6+K)=-1./L(J+1)
      C(LLJ,1)=-L(2)/3.
      C(LLJ,2)=-1./L(2)
      C(LLJ,3)=-L(2)/6.
      C(LLJ,4)=1./L(2)
      C(3+K,LLJ)=C(LLJ,3+K)
      C(4+K,LLJ)=C(LLJ,4+K)
      C(5+K,LLJ)=C(LLJ,5+K)
      C(6+K,LLJ)=C(LLJ,6+K)
      C(1,LLJ)=C(LLJ,1)
      C(2,LLJ)=C(LLJ,2)
      C(3,LLJ)=C(LLJ,3)
      C(4,LLJ)=C(LLJ,4)

*     FORCE EQUALITY OF THE FUNCTIONAL VALUES AT THE ENDPOINTS

      LLJ=LLJ+1
      C(LLJ,2)=1.
      C(2,LLJ)=1.
      C(LLJ,6+K)=-1.
      C(6+K,LLJ)=-1.
   18 J=J+1
   19 II=II+2
      IM=IN+1
   20 CONTINUE
      IF (ICODE.EQ.2) GO TO 39
   21 CALL SIMEQ (C,LA,C(1,LR),1,DET,L,NC,ISCALE)
      IF (ICODE.EQ.5) GO TO 32
      GO TO 39

*     COMPUTES VALUES OF THE FUNCTION

   22 DO 23 I=IM,IN
      K=I-IM+1
      Y(I)=C(1+II,LR)*A(K,1)+C(2+II,LR)*A(K,2)+C(3+II,LR)*A(K,3)+C(4+II,
     1LR)*A(K,4)
   23 CONTINUE
      GO TO 19
   24 IN=0
      IM=1
      IF (ICODE.EQ.4) GO TO 27

*     COMPUTES VALUES OF THE FIRST DERIVATIVE

      DO 26 J=2,NKR1
      L(J)=R(J)-R(J-1)
      IN=N(J-1)+IN
      SLJ=C(2*J-1,LR)
      SLJ1=C(2*J-3,LR)
      YLJ=C(2*J,LR)
      YLJ1=C(2*J-2,LR)
      DO 25 I=IM,IN
      Y(I)=SLJ1*(-(X(I)-R(J))**2/(2.*L(J))+L(J)/6.)+SLJ*((X(I)-R(J-1))**
     12/(2.*L(J))-L(J)/6.)+(YLJ-YLJ1)/L(J)
   25 CONTINUE
      IM=IN+1
   26  CONTINUE
       GO TO 39

*      COMPUTES VALUES OF THE SECOND DERIVATIVE

   27  DO 29 J=2,NKR1
       SLJ=C(2*J-1,LR)
       SLJ1=C(2*J-3,LR)
       YLJ=C(2*J,LR)
       YLJ1=C(2*J-2,LR)
       IN=N(J-1)+IN
       L(J)=R(J)-R(J-1)
       DO 28 I=IM,IN
       Y(I)=SLJ*((X(I)-R(J-1))/L(J))+SLJ1*((R(J)-X(I))/L(J))
   28  CONTINUE
       IM=IN+1
   29  CONTINUE
       GO TO 39

*      COMPUTES THE EQUATIONS OF THE FORM AX**3+BX**2+CX+D

   30  LR1=LR
       DO 37 J=2,NKR1
       DO 31 IL=1,2
       C(2*IL-1,2)=2.
       C(2*IL-1,3)=0.
       C(2*IL-1,4)=0.
       C(2*IL,4)=1.
       C(2*IL-1,1)=6.*R(J+IL-2)
       DO 31 I=1,3
       C(I,5)=C(2*J-4+I,LR)
       C(2*IL,I)=(R(J+IL-2))**(4-I)
   31  CONTINUE
       C(4,5)=C(2*J,LR)
       LA=4
       LR=5
       GO TO 21
   32  LR=LR1
       JI=J-1
       PRINT 41, JI,(C(I,5),I=1,4)

*           SAVE COEFFICIENTS

       IF (KODE.EQ.4) GO TO 36
       IF (KODE.NE.0) GO TO 34
       DO 33 I=1,4
   33  CS(JI,I)=C(I,5)
       GO TO 36
   34  DO 35 I=1,4
   35  CS(JI+27,I)=C(I,5)
   36  CONTINUF
       NR=LO-NKR1+J
       DO 37 I=1,4
       C(NR,I)=C(I,5)
   37  CONTINUF
       DO 38 J=1,NKR
       NR=LR-NKR1+J
       DO 38 I=1,4
       C(J,I)=C(NR,I)
   38  CONTINUF
   39  RETURN

*

   40  FORMAT (36H ERROR CONDITION--NMAX IS TOO SMALL./11H THERE ARE ,I5,
      117H POINTS ON CURVE ,I2,1H.)
   41  FORMAT (/7H CURVE ,I2/4X,F23.9,7H(X**3)+,F23.9,7H(X**2)+,F23.9,4H(
      1X)+,F23.9)
       END
```

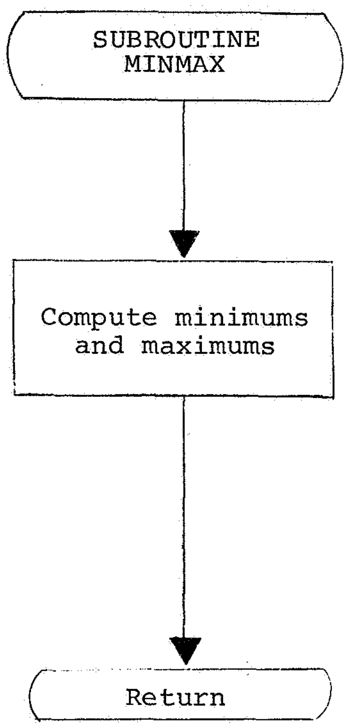
Subroutine SUP. - Subroutine SUP, used in cases selecting the parametric version, computes the parametric variable t and sets up the arrays to be used in subroutine CUSPFIT. The flow diagram for subroutine SUP is as follows:


SUBROUTINE SUP
Compute parametric variable
Move arrays to make t independent variable and x dependent variable
Return

The program listing for subroutine SUP is as follows:

```
      SUBROUTINE SUP (X,Y,T,Z,XMAXT,NMAT)                              D   1    77000000
                                                                                77100000
*            COMPUTE PARAMETRIC VARIABLE AND SET UP ARRAYS             D   2    77200000
                                                                                77300000
      DIMENSION X(310), Y(310), T(310), Z(310)                         D   3    77400000
      T(1)=XMAXT=0.                                                    D   4    77500000
      DO 1 I=2,NMAT                                                    D   5    77600000
    1 T(I)=(X(I)-X(I-1))**2+(Y(I)-Y(I-1))**2+T(I-1)                    D   6    77700000
      IF (T(I).GT.XMAXT) XMAXT=T(I)                                    D   7    77800000
      DO 2 I=1,NMAT                                                    D   8    77900000
      Z(I)=Y(I)                                                        D   9    78000000
      Y(I)=X(I)                                                        D  10    78100000
    2 X(I)=T(I)                                                        D  11    78200000
      RETURN                                                           D  12    78300000
      END                                                              D  13-   78400000
```

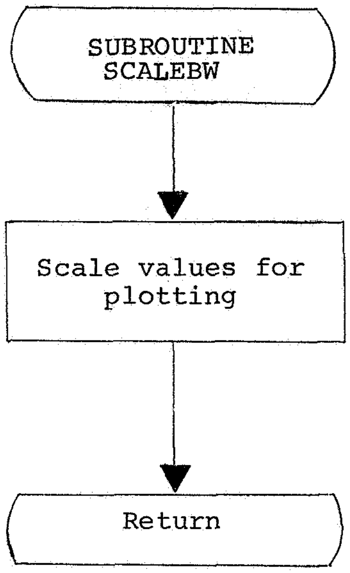
Subroutine MINMAX. - Subroutine MINMAX finds the minimum and maximum values of data to be plotted when using the parametric option of the program. The flow diagram for subroutine MINMAX is as follows:



The program listing for subroutine MINMAX is as follows:

```
      SUBROUTINE MINMAX (A,N,AMIN,AMAX)                              E  1   78500000
                                                                            78600000
*       TO FIND MINIMUM AND MAXIMUM VALUES IN                        E  2   78700000
*     C      A= AN ARRAY, USER MUST DIMENSION A TO FIT HIS ARRAY      E  3   78800000
*     C      N= NO OF VALUES IN THE ARRAY                            E  4   78900000
                                                                            79000000
      DIMENSION A(310)                                               E  5   79100000
      AMIN=1.0E20                                                    E  6   79200000
      AMAX=1.0E-20                                                   E  7   79300000
      DO 1 I=1,N                                                     E  8   79400000
      IF (A(I).LT.AMIN) AMIN=A(I)                                    E  9   79500000
      IF (A(I).GT.AMAX) AMAX=A(I)                                    E 10   79600000
    1 CONTINUF                                                       E 11   79700000
      RETURN                                                         E 12   79800000
      FND                                                            E 13-  79900000
```

Subroutine SCALEBW.- Subroutine SCALEBW scales the values for plotting in cases using the parametric option. The flow diagram for subroutine SCALEBW is as follows:



The program listing for subroutine SCALEBW is as follows:

```
      SUBROUTINE SCALEBW (YMIN,YMAX)

*     SCALE FOR PLOTTING

      DIMENSION FAC(3)
      FAC(1)=1.0
      FAC(2)=2.0
      FAC(3)=5.0
      FAK=1.0E-8
      YMINS=YMIN
      YMAXS=YMAX
      A=(YMAX-YMIN)/10.0
    1 CONTINUE
      DO 3 J=1,16
      FAK=10.0*FAK
      DO 2 I=1,3
      C=FAK*FAC(I)
      IF (A.GT.C) GO TO 2
      B=AMOD(YMIN,C)
      IF (YMIN.GT.0.0) YMIN=YMIN-B
      IF (YMIN.LT.0.0) YMIN=YMIN-(C+B)
      YMAX=YMIN+10.0*C
      IF (YMINS.LT.YMIN) GO TO 4
      IF (YMAXS.GT.YMAX) GO TO 4
      RETURN
    2 CONTINUE
    3 CONTINUE
      PRINT 5, A,YMIN,YMAX,C
      RETURN
    4 A=(YMAXS-YMINS)/5.0
      GO TO 1

*
    5 FORMAT (/40H    NEED TO ALTER SUBROUTINE SCALEBW    A=E15.8,2X,5HYMI
     1N=E15.8,2X,5HYMAX=E15.8,2X,5H$$$$$/3X,2HC=E15.8/)
      END
```

Langley Library Subroutine SIMEQ

Language: FORTRAN

Purpose: SIMEQ solves the matrix equation $AX = B$ where A is a square coefficient matrix and B is a matrix of constant vectors. The solution to a set of simultaneous equations and the determinant may be obtained. If the user wants the determinant only, use DETEV for savings in time and storage.

Use: CALL SIMEQ (A, N, B, M, DETERM, IPIVOT, NMAX, ISCALE)

A A two-dimensional array of the coefficients.

N The order of A; $1 \leqq N \leqq NMAX$.

B A two-dimensional array of the constant vectors B. On return to calling program, X is stored in B.

M The number of column vectors in B.

DETERM Gives the value of the determinant by the following formula:

$$\mathrm{DET}(A) = 10^{100}\ \mathrm{ISCALE}_{(\mathrm{DETERM})}$$

IPIVOT A one-dimensional array of temporary storage used by the routine.

NMAX The maximum order of A as stated in dimension statement of calling program.

ISCALE A scale factor computed by subroutine to keep results of computation within the floating-point word size of the computer.

Restrictions: Arrays A, B, and IPIVOT are dimensioned with variable dimensions in the subroutine. The maximum size of these arrays must be specified in a DIMENSION statement of the calling program as: A (NMAX, NMAX), B (NMAX, M), IPIVOT (NMAX). The original matrices, A and B, are destroyed. They must be saved by the user if there is further need for them. The determinant is set to zero for a singular matrix.

Method: Jordan's method is used through a succession of elementary transformations: $l_n, l_{n-1}, \ldots, l_1$. If these transformations are applied to a matrix B of constant vectors, the result is X where $AX = B$. Each transformation is selected so that the largest element is used in the pivotal position.

Accuracy: Total pivotal strategy is used to minimize the rounding errors; however, the accuracy of the final results depends upon how well-conditioned the original matrix is.

Reference: (a) Fox, L.: An Introduction to Numerical Linear Algebra. Oxford Univ. Press, c.1965.

Storage: 432_8 locations.

Subroutine date: August 1, 1968.

Langley Library Subroutine FTLUP

Language: FORTRAN

Purpose: Computes $y = F(x)$ from a table of values using first- or second-order interpolation. An option to give y a constant value for any x is also provided.

Use: CALL FTLUP(X, Y, M, N, VARI, VARD)

X The name of the independent variable x.

Y The name of the dependent variable $y = F(x)$.

M The order of interpolation (an integer)

$M = 0$ for y a constant. VARD(I) corresponds to VARI(I) for $I = 1, 2, . . ., N$. For $M = 0$ or $N \leqq 1$, $y = F(VARI(1))$ for any value of x. The program extrapolates.

$M = 1$ or 2. First or second order if VARI is strictly increasing (not equal).

$M = -1$ or -2. First or second order if VARI is strictly decreasing (not equal).

N The number of points in the table (an integer).

VARI The name of a one-dimensional array which contains the N values of the independent variable.

VARD The name of a one-dimensional array which contains the N values of the dependent variable.

Restrictions: All the numbers must be floating point. The values of the independent variable x in the table must be strictly increasing or strictly decreasing. The following arrays must be dimensioned by the calling program as indicated: VARI(N), VARD(N).

Accuracy: A function of the order of interpolation used.

References: (a) Nielsen, Kaj L.: Methods in Numerical Analysis. The Macmillan Co., c.1956, pp. 87-91.
(b) Milne, William Edmund: Numerical Calculus. Princeton Univ. Press, c.1949, pp. 69-73.

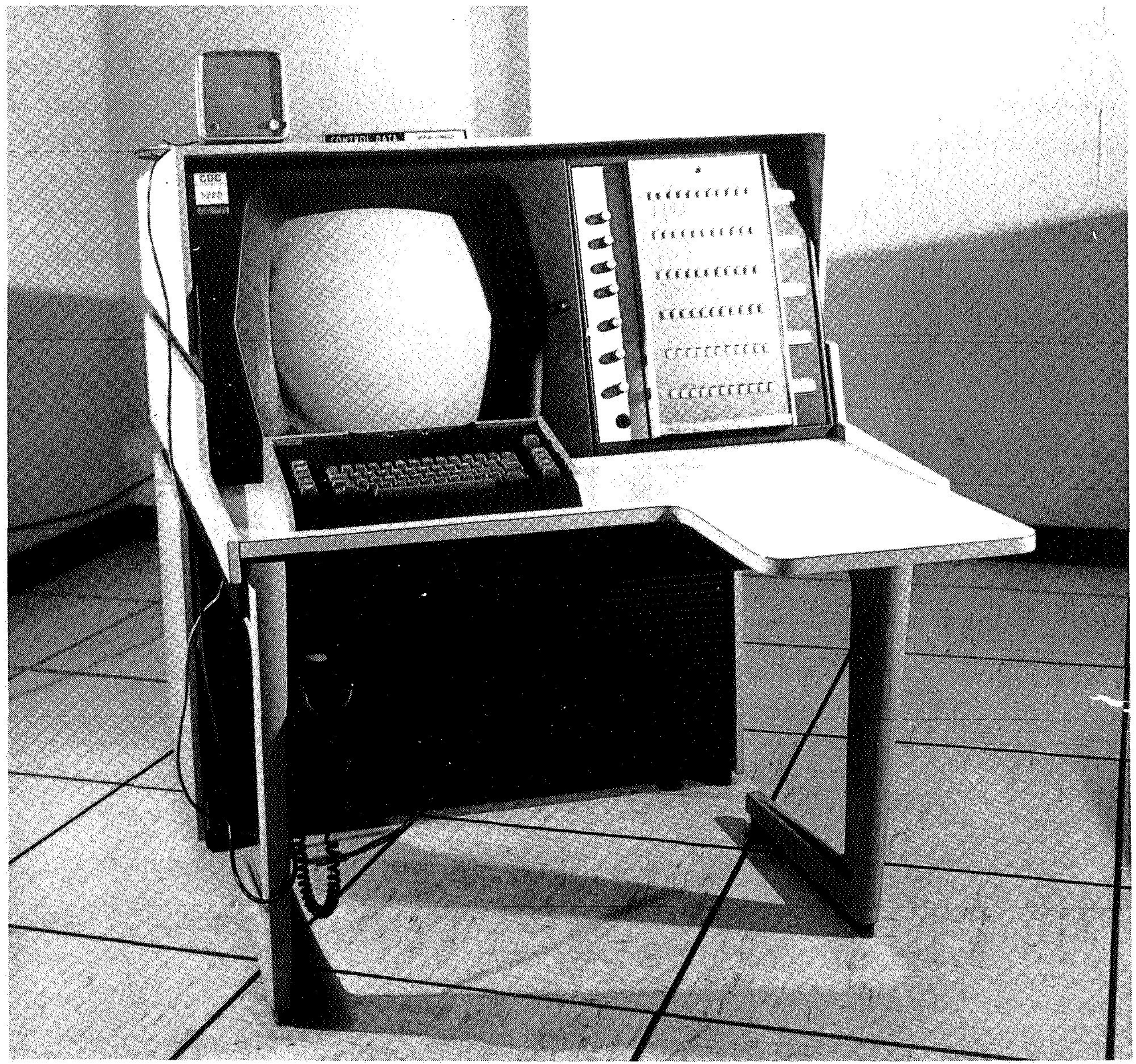
Storage: 430_8 locations.

Error condition: If the VARI values are not in order, the subroutine will print TABLE BELOW OUT OF ORDER FOR FTLUP AT POSITION xxx TABLE IS STORED IN LOCATION xxxxxx (absolute). It then prints the contents of VARI and VARD, and STOPS the program.

Subroutine date: September 12, 1969.

Usage

The program D3670 is run on the Control Data series 6000 computer under the scope 3.0 operating system. The storage required for a batch run is approximately 70000_8 locations and for an on-line CRT run is approximately 75000_8 locations. Cases implementing the on-line CRT capability use the CDC 250 CRT console. Instructions are written into the program and displayed on the screen to inform the on-line CRT user of options available to him at various points in the program. The user conveys his selections through the use of the function keyboard and the typewriter keyboard at the CRT console (see fig. 3). If the curve fit obtained is not satisfactory, an on-line CRT user can



L-72-4136

Figure 3.- CDC 250 series CRT console.

change the values of the joints $\left(\text{set}\ \left\{\bar{x}_j\right\}_{j=2}^{m-1}\right)$ and possibly the number of curves (m - 1) in the spline approximation and go back to recompute the spline function. When running a parametric CRT case, the user can go back at any time to recompute $S_{\Delta_x}(t)$ or $S_{\Delta_y}(t)$ so that these approximations might be improved. This can be desirable after displaying the curve on an x-y coordinate system with points computed as t is incremented through its range.

When a case is run as a batch job, the options mentioned above are not available so that the job is completed with the values originally input.

Input Description

Input is standard CDC FORTRAN NAMELIST. There is an option available allowing the coordinates of the input points to be read in a format so that cards punched by another program could be directly input into this program. If a batch parametric case is being run, the first input values for the joints are taken as the joints for the spline function $S_{\Delta_x}(t)$. A second set of input is read so that the joints may be changed for computing $S_{\Delta_y}(t)$. If a CRT parametric case is being run, the second read could be bypassed if the same joints used in the computation for $S_{\Delta_x}(t)$ are desired for the computation of $S_{\Delta_y}(t)$. This is done according to instructions from the CRT console.

To simplify the necessary input in a standard nonparametric case, the value of the abscissa of the first point to be smoothed is subtracted from all other abscissas before the spline functions are computed. This enables the user to input joints beginning with zero and stepping up to the value of the abscissa of the last point minus the value of the abscissa of the first point.

The NAMELIST input data, listed under $NAM1 are given as follows:

X array of abscissas of points to be smoothed

YY array of ordinates of points to be smoothed

NKR number of cubics to be used to fit the spline function
(DEFAULT = 1)

R array of (NKR + 1) values for the endpoints and joints between curves in the spline function
(DEFAULT: R(1) = 0.0, R(NKR + 1) = X(NMAT) - X(1))

NMAT number of points to be smoothed

IFLAG = 0 if coordinates of points are input with a format
= 1 if coordinates are input with NAMELIST
(DEFAULT = 1)

W array of NMAT values of weights for input points
(DEFAULT = 1.0, i = 1, 2, . . ., NMAT)

KØ = 1 for standard CRT version
= 2 for standard batch version
= 3 for parametric CRT version
= 4 for parametric batch version
(DEFAULT = 1)

KLØSE = 0 for arbitrary curve
= 1 for closed curve parametric case
(DEFAULT = 0)

Output Description

Output is in the form of plotted curves and printed data. For a standard case, the input data points and the computed spline function are plotted on an x-y grid. Small vertical bars are plotted to indicate the junction points in the spline function. For an on-line case, these joints may then be manipulated to obtain a satisfactory curve fit. The printed data for a standard case consist of a listing of the number of points, the number of curves, the endpoints and junction points, and a table giving the input y values, the computed spline and the residuals at the input x values. This is followed by the residual standard deviation and a listing of the coefficients of the computed cubic curves making up the spline function.

For a parametric case, similar data is plotted on a t-x grid and printed with this input and computed x as a function of t. This is followed by a plot on a t-y grid and printed output of y as a function of t. The input points are then plotted on an x-y grid with the curve determined by incrementing through values of t and plotting $S_{\Delta_y}\left(S_{\Delta_x}(t)\right)$. A listing of the value of t for the input x and y is also printed with x, y, $S_{\Delta_x}(t)$, $S_{\Delta_y}\left(S_{\Delta_x}(t)\right)$, and the residuals.

APPENDIX C

EXAMPLE APPLICATIONS OF ALGORITHMS AND COMPUTER PROGRAM

This appendix describes three example applications of the use of the algorithms and corresponding computer program. These examples demonstrate the explicit techniques and the parametric technique with and without the closed-curve option. The data sets of the applications are chosen for their demonstrative character rather than their relativeness to particular engineering problems. The input data are shown as they would be written for submittal to the computer, and the output data are shown as they would appear on the CRT and output listing.

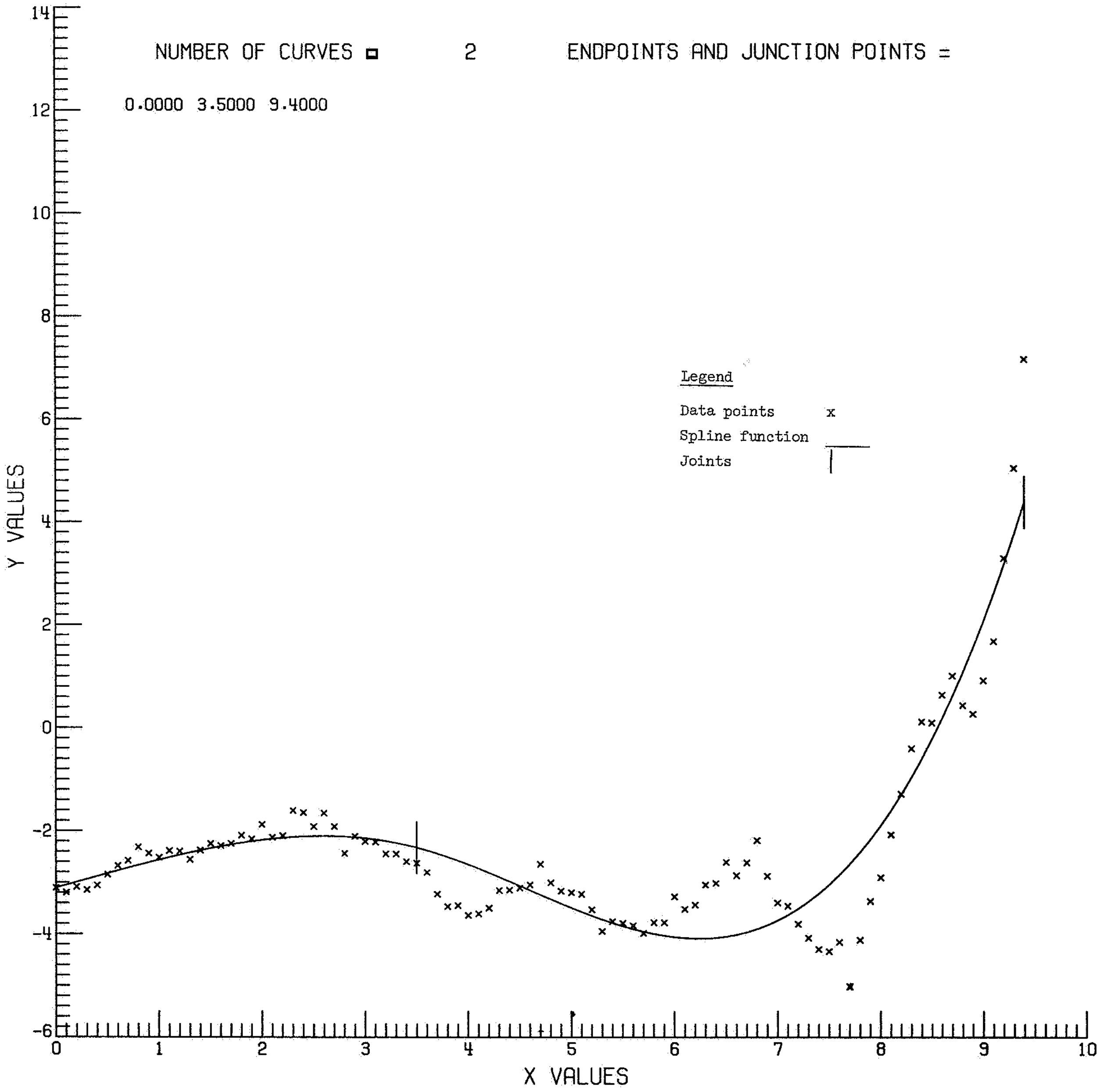
Example Applications

Case 1 Explicit Algorithm

Case 1 is a standard case, the input is as follows:

```
$NAM1

X        =  0.0,   0.1E+00,   0.2E+00,   0.3E+00,   0.4E+0C,   0.5E+00,   0.6E+00,
            0.7E+00,   0.8E+0C,   0.9E+0C,   0.1E+01,   0.11E+01,   0.12E+01,
            0.13E+01,   0.14E+01,   0.15E+01,   0.16E+01,   0.17E+01,   0.18E+01,
            0.19E+01,   0.2E+01,   0.21E+01,   0.22E+01,   0.23E+01,   0.24E+01,
            0.25E+01,   0.26E+01,   0.27E+01,   0.28E+01,   0.29E+01,   0.3E+01,
            0.31E+01,   0.32E+01,   0.33E+01,   0.34E+01,   0.35E+01,   0.36E+01,
            0.37E+01,   0.38E+01,   0.39E+01,   0.4E+01,   0.41E+01,   0.42E+01,
            0.43E+01,   0.44E+01,   0.45E+01,   0.46E+01,   0.47E+01,   0.48E+01,
            0.49E+01,   0.5E+01,   0.51E+01,   0.52E+01,   0.53E+01,   0.54E+01,
            0.55E+01,   0.56E+01,   0.57E+C1,   0.58E+01,   0.59E+01,   0.6E+01,
            0.61E+01,   0.62E+01,   0.63E+C1,   0.64E+01,   0.65E+01,   0.66E+01,
            0.67E+01,   0.68E+01,   0.69E+01,   0.7E+01,   0.71E+01,   0.72E+01,
            0.73E+01,   0.74E+01,   0.75E+01,   0.76E+01,   0.77E+01,   0.78E+01,
            0.79E+01,   0.8E+01,   0.81E+01,   0.82E+01,   0.83E+01,   0.84E+01,
            0.85E+01,   0.86E+01,   0.87E+C1,   0.88E+01,   0.89E+01,   0.9E+01,
            0.91E+01,   0.92E+01,   0.93E+01,   0.94E+01,

YY       = -0.311E+01, -0.32E+01, -0.309E+01, -0.315E+01, -0.306E+01,
           -0.285E+01, -0.268E+01, -0.258E+01, -0.232E+01, -0.244E+01,
           -0.252E+01, -0.239E+01, -0.24E+01, -0.256E+01, -0.238E+01,
           -0.225E+01, -0.229E+01, -0.225E+01, -0.209E+01, -0.216E+01,
           -0.188E+01, -0.213E+01, -0.21E+01, -0.161E+01, -0.165E+01,
           -0.191994E+01, -0.166E+01, -0.192E+01, -0.244E+01, -0.211E+01,
           -0.221E+01, -0.222E+01, -0.245E+01, -0.245E+01, -0.26E+01,
           -0.263E+01, -0.281E+01, -0.323E+01, -0.347E+01, -0.345E+01,
           -0.364E+01, -0.361E+01, -0.35E+01, -0.316E+01, -0.315E+01,
           -0.311E+01, -0.305E+01, -0.265E+01, -0.301E+01, -0.317E+01,
           -0.32E+01, -0.323E+01, -0.353E+01, -0.395E+01, -0.376E+01,
           -0.379E+01, -0.384E+01, -0.399E+01, -0.378E+01, -0.378E+01,
           -0.328E+01, -0.352E+01, -0.344E+01, -0.305E+01, -0.302E+01,
           -0.261E+01, -0.287E+01, -0.262E+01, -0.219E+01, -0.288E+01,
           -0.34E+01, -0.346E+01, -0.381E+01, -0.408E+01, -0.43E+01, -0.434E+01,
           -0.416E+01, -0.502E+01, -0.412E+01, -0.337E+01, -0.291E+01,
           -0.208E+01, -0.129E+01, -0.41E+0C,  0.11E+00,  0.9E-01,  0.63E+00,
            0.1E+01,  0.43E+00,  0.26E+00,  0.91E+00,  0.167E+01,  0.328E+01,
            0.5C3E+01,  0.715E+01,
NKR      =  2,

R        =  0.0,   0.35E+01,   0.94E+01,

NMAT     =  95,

IFLAG    =  1,

W        =  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,
            0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,
            0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,
            0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,
            0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,
            0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,
            0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,
            0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,
            0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,
            0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,
            0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,
            0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,
            0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,
            0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,

KC       =  2,

KLOSE    =  0,

$END
```

In this case, it is desirable to find a curve fit which will smooth the input data. The output is as follows:

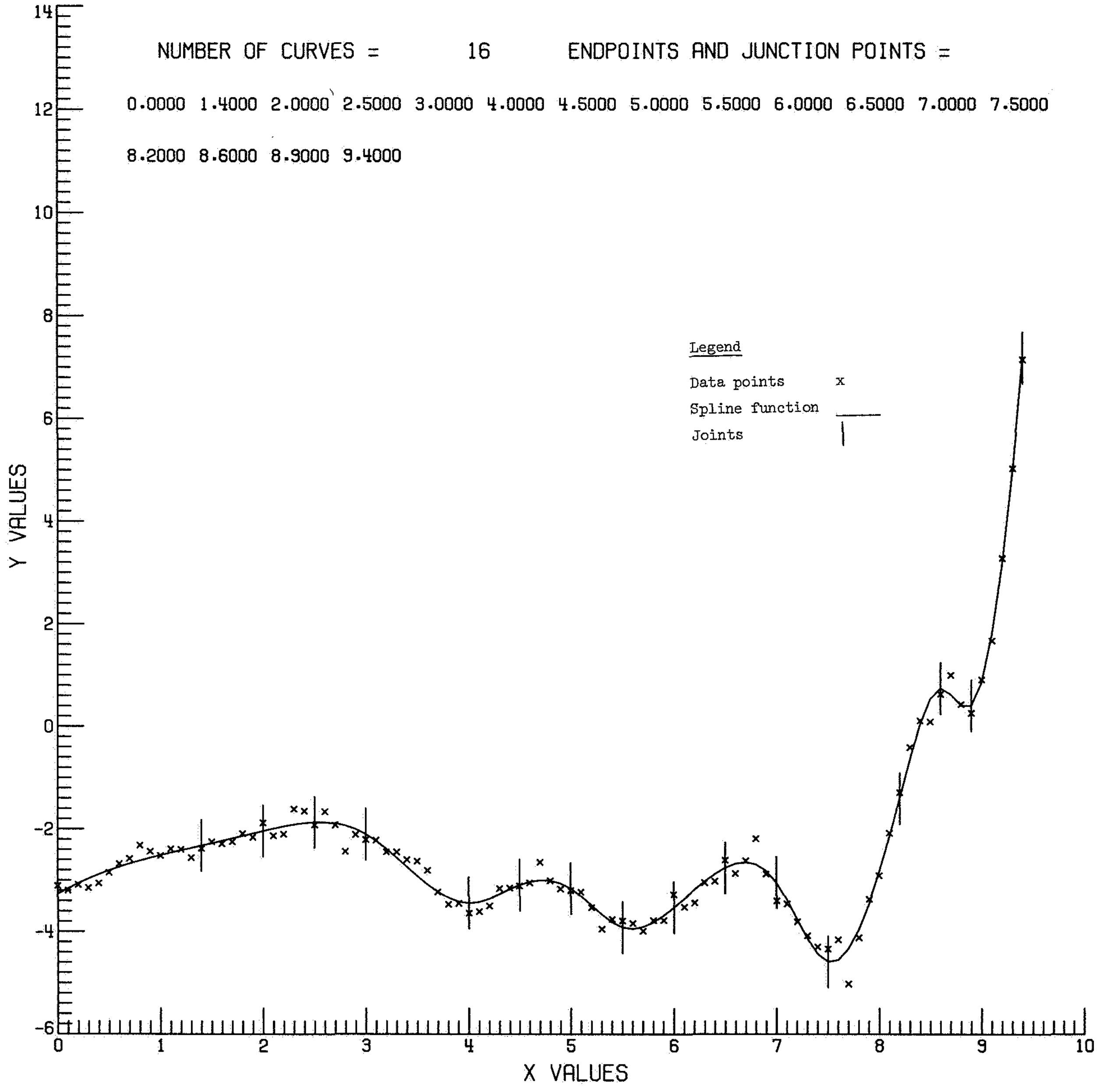

NUMBER OF CURVES = 2 ENDPOINTS AND JUNCTION POINTS =
0.0000 3.5000 9.4000
Legend
Data points x
Spline function
Joints
Y VALUES
X VALUES
14
12
10
8
6
4
2
0
-2
-4
-6
0
1
2
3
4
5
6
7
8
9
10

```
               ***************
               DATA FOR X VS Y
               ***************

NUMBER OF CURVES =   2           NUMBER OF POINTS =  95
ENDPOINTS AND JUNCTION POINTS =
      0.                 3.5000000E+00        9.4000000E+00

            X                  Y               COMPUTED Y          RESIDUALS
 1    0.                 -3.1100000E+00      -3.1077998E+00      -2.2001626E-03
 2    1.0000000E-01      -3.2000000E+00      -3.0521385E+00      -1.4786148E-01
 3    2.0000000E-01      -3.0900000E+00      -2.9962368E+00      -9.3763178E-02
 4    3.0000000E-01      -3.1500000E+00      -2.9402950E+00      -2.0970500E-01
 5    4.0000000E-01      -3.0600000E+00      -2.8845133E+00      -1.7548667E-01
 6    5.0000000E-01      -2.8500000E+00      -2.8290920E+00      -2.0907952E-02
 7    6.0000000E-01      -2.6800000E+00      -2.7742314E+00       9.4231431E-02
 8    7.0000000E-01      -2.5800000E+00      -2.7201317E+00       1.4013173E-01
 9    8.0000000E-01      -2.3200000E+00      -2.6669932E+00       3.4699322E-01
            .                  .                  .                  .
            .                  .                  .                  .
            .                  .                  .                  .
90    8.9000000E+00       2.6000000E-01       1.5417671E+00      -1.2817671E+00
91    9.0000000E+00       9.1000000E-01       2.0511187E+00      -1.1411187E+00
92    9.1000000E+00       1.6700000E+00       2.5879385E+00      -9.1793855E-01
93    9.2000000E+00       3.2800000E+00       3.1528456E+00       1.2715441E-01
94    9.3000000E+00       5.0300000E+00       3.7464586E+00       1.2835414E+00
95    9.4000000E+00       7.1500000E+00       4.3693965E+00       2.7806035E+00

      STANDARD DEVIATION= 7.33621E-01

CURVE  1
             -.033376710(X**3)+          .022032038(X**2)+           .554743737(X)+          -3.107799837

CURVE  2
              .103141705(X**3)+        -1.411411313(X**2)+          5.571795465(X)+          -8.961026854
```

A better fit to the data can be realized by adding more cubics to the spline function, thus introducing the following changes in the input data:

```
NKR      =  16,

P        =  C.0,  0.14E+01,  C.2E+01,  0.25E+01,  0.3E+01,  0.4E+01,  0.45E+01,
            0.5E+01,  0.55E+01,  0.6E+01,  0.65E+01,  0.7E+C1,  0.75E+01,
            0.82E+C1,  C.86E+C1,  0.89E+C1,  C.94E+01,
```

The output from this improved fit is as follows:


NUMBER OF CURVES = 16 ENDPOINTS AND JUNCTION POINTS =
0.0000 1.4000 2.0000 2.5000 3.0000 4.0000 4.5000 5.0000 5.5000 6.0000 6.5000 7.0000 7.5000
8.2000 8.6000 8.9000 9.4000
Legend
Data points x
Spline function
Joints
Y VALUES
14
12
10
8
6
4
2
0
-2
-4
-6
0
1
2
3
4
5
6
7
8
9
10
X VALUES

```
          ***************
          DATA FOR X VS Y
          ***************


NUMBER OF CURVES =  16          NUMBER OF POINTS =  95
ENDPOINTS AND JUNCTION POINTS =
      0.                1.4000000E+00      2.0000000E+00      2.5000000E+00      3.0000000E+00      4.0000000E+00
      4.5000000E+00     5.0000000E+00      5.5000000E+00      6.0000000E+00      6.5000000E+00      7.0000000E+00
      7.5000000E+00     8.2000000E+00      8.6000000E+00      8.9000000E+00      9.4000000E+00

            X                 Y             COMPUTED Y          RESIDUALS
  1   0.               -3.1100000E+00     -3.2816370E+00      1.7163702E-01
  2   1.0000000E-01    -3.2000000E+00     -3.1651243E+00     -3.4875708E-02
  3   2.0000000E-01    -3.0900000E+00     -3.0594144E+00     -3.0585606E-02
  4   3.0000000E-01    -3.1500000E+00     -2.9636361E+00     -1.8636393E-01
  5   4.0000000E-01    -3.0600000E+00     -2.8769181E+00     -1.8308194E-01
  6   5.0000000E-01    -2.8500000E+00     -2.7983891E+00     -5.1610896E-02
  7   6.0000000E-01    -2.6800000E+00     -2.7271779E+00      4.7177948E-02
  8   7.0000000E-01    -2.5800000E+00     -2.6624133E+00      8.2413332E-02
  9   8.0000000E-01    -2.3200000E+00     -2.6032240E+00      2.8322400E-01
            .                 .                 .                  .
            .                 .                 .                  .
            .                 .                 .                  .
 90   8.9000000E+00     2.6000000E-01      4.0586134E-01     -1.4586134E-01
 91   9.0000000E+00     9.1000000E-01      8.6234627E-01      4.7653733E-02
 92   9.1000000E+00     1.6700000E+00      1.8009450E+00     -1.3094503E-01
 93   9.2000000E+00     3.2800000E+00      3.1887811E+00      9.1218872E-02
 94   9.3000000E+00     5.0300000E+00      4.9929780E+00      3.7021957E-02
 95   9.4000000E+00     7.1500000E+00      7.1806593E+00     -3.0659266E-02

        STANDARD DEVIATION= 1.84735E-01

CURVE  1
                .145209670(X**3)+        -.583704423(X**2)+        1.222045627(X)+        -3.281637020

CURVE  2
                .001345748(X**3)+         .020524051(X**2)+         .376125762(X)+        -2.886874417

      .
      .
      .

CURVE 16
              -5.479418546(X**3)+     172.049992849(X**2)+     -1758.370951912(X)+     5884.647612931
```

Case 2

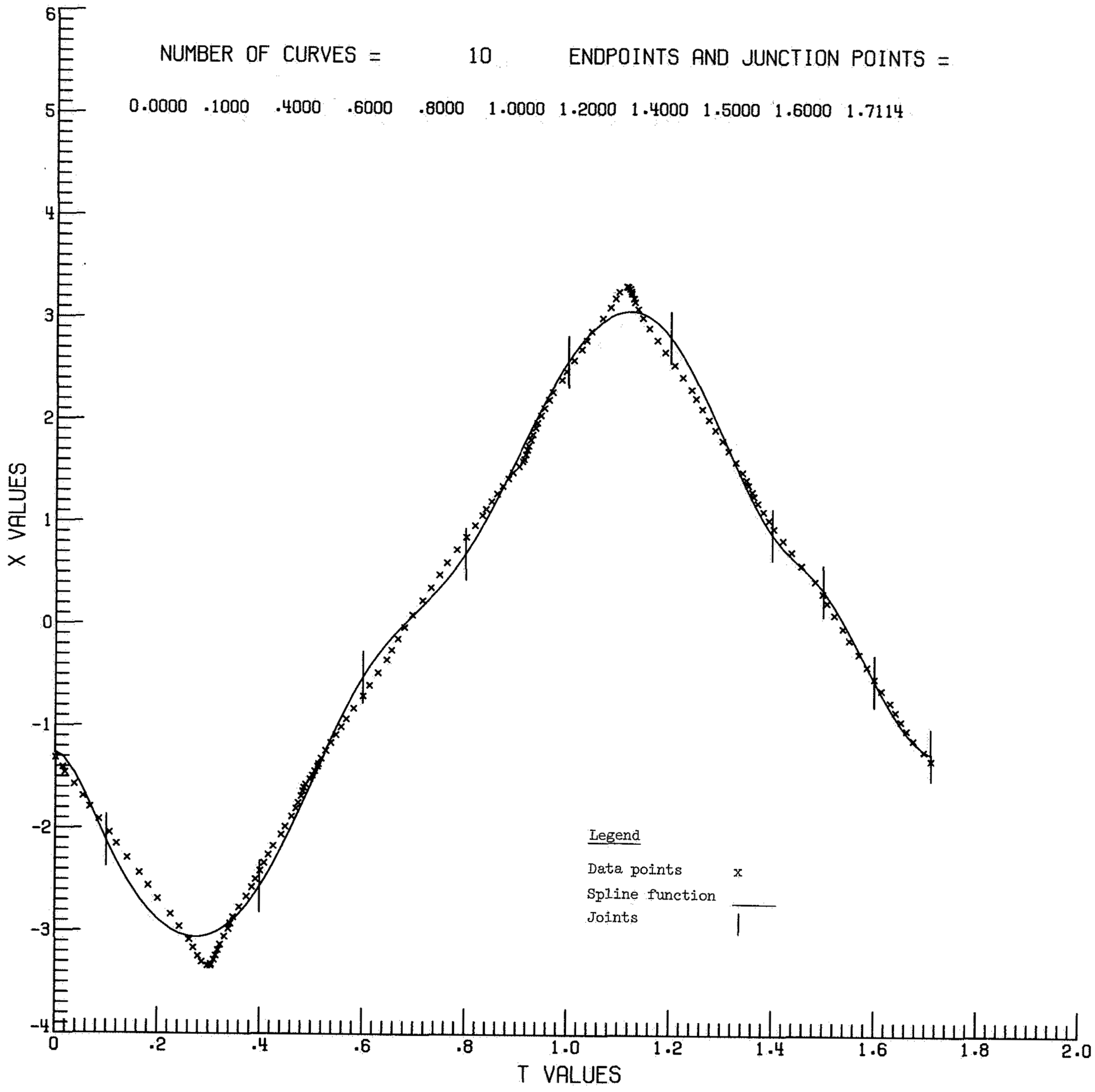
Case 2 is a parametric algorithm, closed curve. The input with junction values for $S_{\Delta_X}(t)$ is as follows:

```
$NAM1

X        = -0.13143E+01, -0.14067E+01, -0.1457E+01, -0.15725E+01, -0.16897E+01,
           -0.1794E+01, -0.19143E+01, -0.20442E+01, -0.21521E+01, -0.22876E+01,
           -0.2434E+01, -0.25576E+01, -0.2686E+01, -0.28353E+01, -0.29571E+01,
           -0.30821E+01, -0.31622E+01, -0.32439E+01, -0.3301E+01, -0.33361E+01,
           -0.33374E+01, -0.33229E+01, -0.3279E+01, -0.32374E+01, -0.31852E+01,
           -0.31324E+01, -0.30542E+01, -0.29771E+01, -0.2926E+01, -0.28629E+01,
           -0.27668E+01, -0.26617E+01, -0.25698E+01, -0.24946E+01, -0.24114E+01,
           -0.23348E+01, -0.22534E+01, -0.21679E+01, -0.20568E+01, -0.19795E+01,
           -0.18808E+01, -0.18024E+01, -0.17466E+01, -0.16796E+01, -0.16311E+01,
           -0.15972E+01, -0.15667E+01, -0.15156E+01, -0.1478E+01, -0.14357E+01,
           -0.13907E+01, -0.13655E+01, -0.13124E+01, -0.12351E+01, -0.11578E+01,
           -0.10804E+01, -0.10031E+01, -0.9252E+00, -0.8219E+00, -0.6998E+00,
           -0.5958E+00, -0.4733E+00, -0.3467E+00, -0.2514E+00, -0.1417E+00,
           -0.291E-01,  0.942E-01,  0.2336E+00,  0.3594E+00,  0.4849E+00,
            0.6045E+00,  0.7325E+00,  0.8549E+00,  0.9692E+00,  0.10688E+01,
            0.11303E+01,  0.12053E+01,  0.12804E+01,  0.13554E+01,  0.14304E+01,
            0.1489E+01,  0.15502E+01,  0.15999E+01,  0.16194E+01,  0.16684E+01,
            0.17136E+01,  0.17529E+01,  0.18105E+01,  0.18631E+01,  0.19281E+01,
            0.19786E+01,  0.20503E+01,  0.21218E+01,  0.22045E+01,  0.22802E+01,
            0.24005E+01,  0.24836E+01,  0.25913E+01,  0.27E+01,  0.27862E+01,
            0.28747E+01,  0.30052E+01,  0.31123E+01,  0.31965E+01,  0.32621E+01,
            0.33141E+01,  0.33142E+01,  0.33116E+01,  0.32931E+01,  0.32671E+01,
            0.32446E+01,  0.31976E+01,  0.31609E+01,  0.30936E+01,  0.30068E+01,
            0.29053E+01,  0.27879E+01,  0.26726E+01,  0.25456E+01,  0.24276E+01,
            0.23074E+01,  0.22205E+01,  0.21177E+01,  0.20142E+01,  0.19131E+01,
            0.1806E+01,  0.17078E+01,  0.16002E+01,  0.14991E+01,  0.14256E+01,
            0.13736E+01,  0.13077E+01,  0.12653E+01,  0.11979E+01,  0.11148E+01,
            0.103E+01,  0.9484E+00,  0.8326E+00,  0.7162E+00,  0.5863E+00,
            0.4325E+00,  0.3126E+00,  0.221E+00,  0.1028E+00, -0.276E-01,
           -0.1404E+00, -0.2773E+00, -0.4004E+00, -0.5174E+00, -0.6302E+00,
           -0.7499E+00, -0.8416E+00, -0.9317E+00, -0.10207E+01, -0.11146E+01,
           -0.12254E+01, -0.13143E+01,

YY       = -0.12983E+01, -0.12234E+01, -0.11862E+01, -0.11143E+01, -0.10539E+01,
           -0.10027E+01, -0.9433E+00, -0.8826E+00, -0.8362E+00, -0.7824E+00,
           -0.7278E+00, -0.6848E+00, -0.6396E+00, -0.5839E+00, -0.5354E+00,
           -0.4818E+00, -0.4443E+00, -0.3928E+00, -0.3293E+00, -0.2288E+00,
           -0.1503E+00, -0.1278E+00, -0.651E-01, -0.257E-01,  0.149E-01,
            0.518E-01,  0.1053E+00,  0.1544E+00,  0.1859E+00,  0.2219E+00,
            0.2733E+00,  0.3265E+00,  0.373E+00,  0.4114E+00,  0.4534E+00,
            0.4947E+00,  0.5364E+00,  0.5792E+00,  0.6364E+00,  0.6778E+00,
            0.7327E+00,  0.7764E+00,  0.8095E+00,  0.8458E+00,  0.8715E+00,
            0.8966E+00,  0.9466E+00,  0.10226E+01,  0.10791E+01,  0.11339E+01,
            0.11936E+01,  0.12241E+01,  0.1272E+01,  0.13349E+01,  0.13978E+01,
            0.14607E+01,  0.15236E+01,  0.15852E+01,  0.16456E+01,  0.1703E+01,
            0.1745E+01,  0.17826E+01,  0.18121E+01,  0.18276E+01,  0.18406E+01,
            0.18471E+01,  0.18447E+01,  0.18329E+01,  0.18126E+01,  0.17841E+01,
            0.17483E+01,  0.16984E+01,  0.16409E+01,  0.15796E+01,  0.15141E+01,
            0.14598E+01,  0.13875E+01,  0.13152E+01,  0.12429E+01,  0.11706E+01,
            0.10981E+01,  0.10046E+01,  0.9358E+00,  0.8828E+00,  0.8567E+00,
            0.8276E+00,  0.803E+00,  0.7685E+00,  0.7384E+00,  0.7009E+00,
            0.6719E+00,  0.6314E+00,  0.5915E+00,  0.5474E+00,  0.5076E+00,
            0.4431E+00,  0.3985E+00,  0.3417E+00,  0.2864E+00,  0.2436E+00,
            0.1995E+00,  0.1317E+00,  0.705E-01,  0.183E-01, -0.321E-01,
           -0.1436E+00, -0.1502E+00, -0.2038E+00, -0.2556E+00, -0.2957E+00,
           -0.3213E+00, -0.3639E+00, -0.392E+00, -0.4334E+00, -0.4773E+00,
           -0.5248E+00, -0.5738E+00, -0.6198E+00, -0.6671E+00, -0.7109E+00,
           -0.7563E+00, -0.7898E+00, -0.8317E+00, -0.8794E+00, -0.9277E+00,
           -0.9793E+00, -0.10268E+01, -0.10806E+01, -0.11347E+01, -0.11823E+01,
           -0.12238E+01, -0.12779E+01, -0.13155E+01, -0.13722E+01, -0.14393E+01,
           -0.15012E+01, -0.15563E+01, -0.16263E+01, -0.16848E+01, -0.17368E+01,
           -0.17826E+01, -0.18109E+01, -0.18263E+01, -0.18417E+01, -0.18495E+01,
           -0.1846E+01, -0.18313E+01, -0.18083E+01, -0.17788E+01, -0.17413E+01,
           -0.16917E+01, -0.16446E+01, -0.15918E+01, -0.15333E+01, -0.14644E+01,
           -0.13747E+01, -0.12983E+01,
NKR    =  10,

R      =  0.0,  0.1E+00,  0.4E+00,  0.6E+00,  0.8E+00,  0.1E+01,  0.12E+01,
          0.14E+01,  0.15E+01,  0.16E+01,  0.17114073E+01,

NMAT   =  157,

IFLAG  =  1,

W      =  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,
          0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,
          0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,
          0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,
          0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,
          0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,
          0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,
          0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,
          0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,
          0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,
          0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,
          0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,
          0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,
          0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,
          0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,
          0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,
          0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,
          0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,
          0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,
          0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,
          0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,
          0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,
          0.1E+01,  0.1E+01,  0.1E+01,

KO     =  4,

KLOSE  =  1,

$END
```

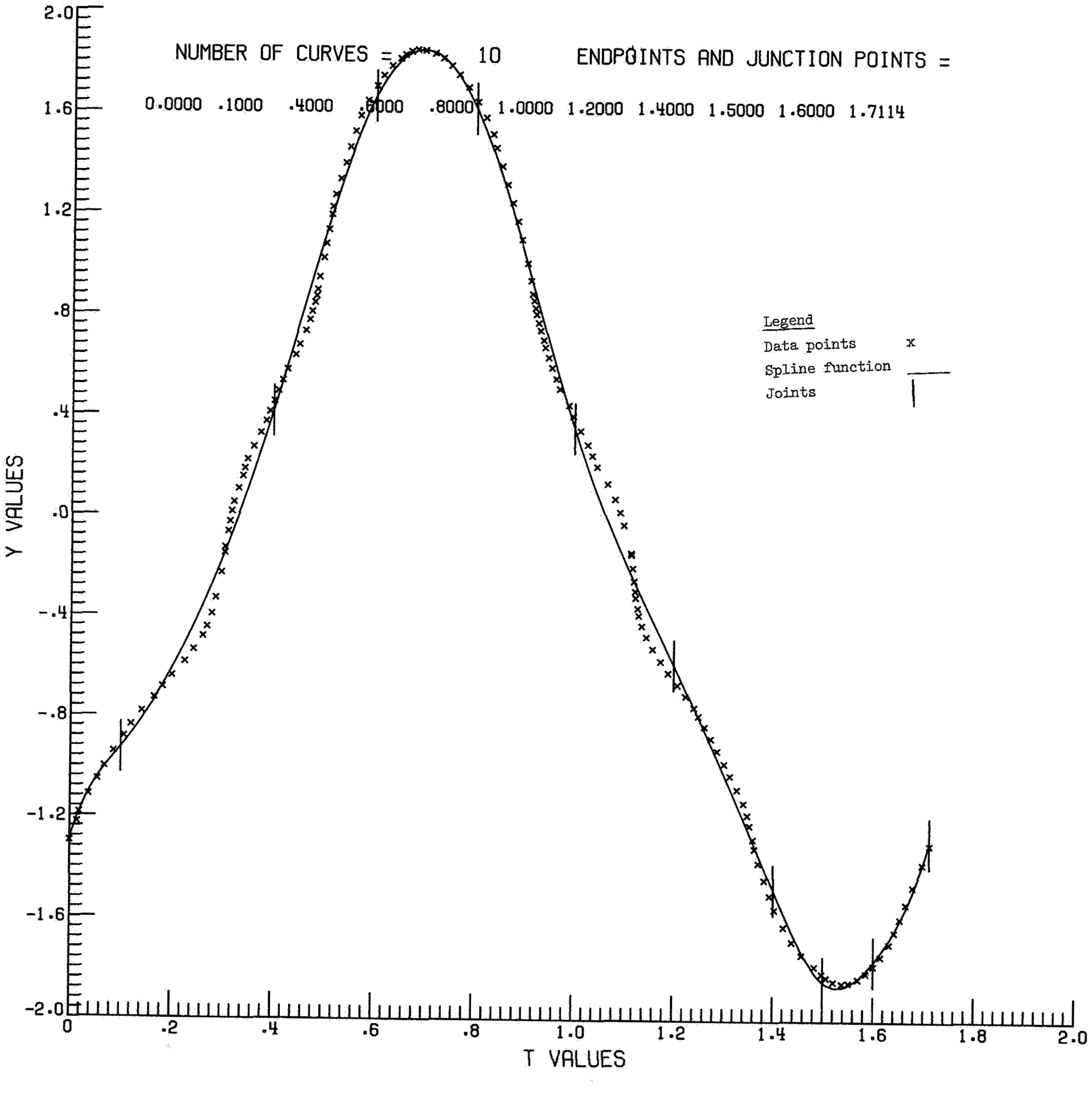
The additional input junction values for $S_{\Delta_y}(t)$ are as follows:

```
$NAM1

NKR    =  10,

R      =  0.0,  0.1E+00,  0.4E+00,  0.6E+00,  0.8E+00,  0.1E+01,  0.12E+01,
          0.14E+01,  0.15E+01,  0.16E+01,  0.17114073E+01,

$END
```

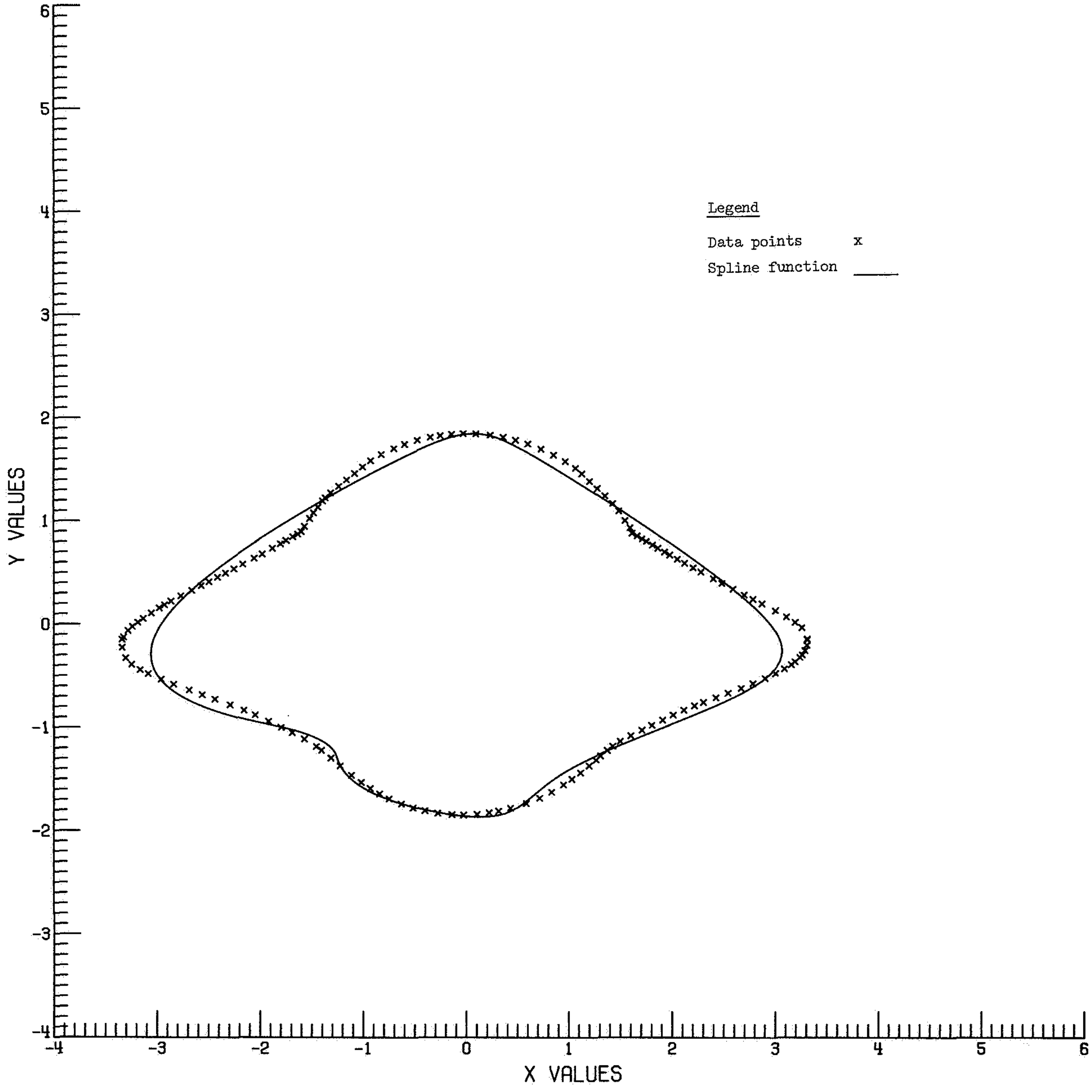
The output for this case is as follows:


NUMBER OF CURVES = 10
ENDPOINTS AND JUNCTION POINTS =
0.0000 .1000 .4000 .6000 .8000 1.0000 1.2000 1.4000 1.5000 1.6000 1.7114
X VALUES
6
5
4
3
2
1
0
-1
-2
-3
-4
0
.2
.4
.6
.8
1.0
1.2
1.4
1.6
1.8
2.0
T VALUES
Legend
Data points x
Spline function
Joints

```
          ***************
          DATA FOR X VS T
          ***************


NUMBER OF CURVES =  10          NUMBER OF POINTS = 157
ENDPOINTS AND JUNCTION POINTS =
      0.                  1.0000000E-01       4.0000000E-01       6.0000000E-01       8.0000000E-01       1.0000000E+00
      1.2000000E+00       1.4000000E+00       1.5000000E+00       1.6000000E+00       1.7114073E+00

           T                  X                 COMPUTED X          RESIDUALS
  1   0.                 -1.3143000E+00      -1.2562252E+00      -5.8074812E-02
  2   1.4147770E-02      -1.4067000E+00      -1.3046593E+00      -1.0204072E-01
  3   1.8061700E-02      -1.4570000E+00      -1.3251786E+00      -1.3182137E-01
  4   3.6571560E-02      -1.5725000E+00      -1.4566541E+00      -1.1584587E-01
  5   5.3955560E-02      -1.6897000E+00      -1.6198388E+00      -6.9861198E-02
  6   6.7455490E-02      -1.7940000E+00      -1.7630946E+00      -3.0905410E-02
  7   8.5455940E-02      -1.9143000E+00      -1.9631309E+00       4.8830882E-02
  8   1.0601444E-01      -2.0442000E+00      -2.1842750E+00       1.4007496E-01
  9   1.1980981E-01      -2.1521000E+00      -2.3197551E+00       1.6765514E-01
           .                  .                   .                   .
           .                  .                   .                   .
           .                  .                   .                   .
152   1.6415309E+00      -8.4160000E-01      -8.9874500E-01       5.7144997E-02
153   1.6524367E+00      -9.3170000E-01      -9.7929420E-01       4.7594197E-02
154   1.6637800E+00      -1.0207000E+00      -1.0549685E+00       3.4268525E-02
155   1.6773444E+00      -1.1146000E+00      -1.1329901E+00       1.8390071E-02
156   1.6976671E+00      -1.2254000E+00      -1.2202940E+00      -5.1060488E-03
157   1.7114073E+00      -1.3143000E+00      -1.2562252E+00      -5.8074812E-02

       STANDARD DEVIATION= 1.26096E-01

CURVE  1
              485.572227656(X**3)+        -116.361517047(X**2)+         -1.874379744(X)+         -1.256225188

CURVE  2
                3.477456440(X**3)+          28.266914317(X**2)+        -16.337222880(X)+          -.774130417

       .
       .
       .

CURVE 10
              133.893616320(X**3)+        -631.694313242(X**2)+        983.809098345(X)+        -505.926027359
```


NUMBER OF CURVES = 10
ENDPOINTS AND JUNCTION POINTS =
0.0000 .1000 .4000 .6000 .8000 1.0000 1.2000 1.4000 1.5000 1.6000 1.7114
Legend
Data points x
Spline function ——
Joints |
2.0
1.6
1.2
.8
.4
.0
-.4
-.8
-1.2
-1.6
-2.0
Y VALUES
0
.2
.4
.6
.8
1.0
1.2
1.4
1.6
1.8
2.0
T VALUES

```
          ***************
          DATA FOR  Y VS T
          ***************


NUMBER OF CURVES =  10          NUMBER OF POINTS = 157
ENDPOINTS AND JUNCTION POINTS =
       0.                 1.0000000E-01        4.0000000E-01        6.0000000E-01        8.0000000E-01        1.0000000E+00
       1.2000000E+00      1.4000000E+00        1.5000000E+00        1.6000000E+00        1.7114073E+00

              T                 Y               COMPUTED Y            RESIDUALS
  1    0.                 -1.2983000E+00       -1.2913831E+00      -6.9168856E-03
  2    1.4147770E-02      -1.2234000E+00       -1.2065720E+00      -1.6827952E-02
  3    1.8061700E-02      -1.1862000E+00       -1.1860888E+00      -1.1120324E-04
  4    3.6571560E-02      -1.1143000E+00       -1.1039707E+00      -1.0329347E-02
  5    5.3955560E-02      -1.0539000E+00       -1.0445381E+00      -9.3619269E-03
  6    6.7455490E-02      -1.0027000E+00       -1.0064855E+00       3.7855377E-03
  7    8.5455940E-02      -9.4330000E-01       -9.6175850E-01       1.8458498E-02
  8    1.0601444E-01      -8.8260000E-01       -9.1166467E-01       2.9064673E-02
  9    1.1980981E-01      -8.3620000E-01       -8.7546362E-01       3.9263621E-02
              .                 .                    .                    .
              .                 .                    .                    .
              .                 .                    .                    .
152    1.6415309E+00      -1.6446000E+00       -1.6390773E+00      -5.5226583E-03
153    1.6524367E+00      -1.5918000E+00       -1.5980793E+00       6.2792564E-03
154    1.6637800E+00      -1.5333000E+00       -1.5508711E+00       1.7571122E-02
155    1.6773444E+00      -1.4644000E+00       -1.4876617E+00       2.3261702E-02
156    1.6976671E+00      -1.3747000E+00       -1.3775245E+00       2.8244839E-03
157    1.7114073E+00      -1.2983000E+00       -1.2913831E+00      -6.9168856E-03

        STANDARD DEVIATION= 6.29658E-02

CURVE  1
            180.611015264(X**3)+        -47.963838251(X**2)+          6.637090021(X)+          -1.291383114

CURVE  2
              1.605097452(X**3)+          5.737937092(X**2)+          1.266912487(X)+          -1.112377197

       .
       .
       .

CURVE 10
             53.313361361(X**3)+       -246.223232730(X**2)+        380.962664149(X)+        -199.343525843
```



Legend
Data points x
Spline function ____
Y VALUES
6
5
4
3
2
1
0
-1
-2
-3
-4
X VALUES
-4
-3
-2
-1
0
1
2
3
4
5
6

```
          T           X          Y COMPUTED X COMPUTED Y  RES X      RES Y

  1  0.        -1.31E+00 -1.30E+00 -1.26E+00 -1.29E+00  5.81E-02  6.92E-03
  2  1.41E-02 -1.41E+00 -1.22E+00 -1.30E+00 -1.21E+00  1.02E-01  1.68E-02
  3  1.81E-02 -1.46E+00 -1.19E+00 -1.33E+00 -1.19E+00  1.32E-01  1.11E-04
  4  3.66E-02 -1.57E+00 -1.11E+00 -1.46E+00 -1.10E+00  1.16E-01  1.03E-02
  5  5.40E-02 -1.69E+00 -1.05E+00 -1.62E+00 -1.04E+00  6.99E-02  9.36E-03
  6  6.75E-02 -1.79E+00 -1.00E+00 -1.76E+00 -1.01E+00  3.09E-02  3.79E-03
  7  8.55E-02 -1.91E+00 -9.43E-01 -1.96E+00 -9.62E-01  4.88E-02  1.85E-02
  8  1.06E-01 -2.04E+00 -8.83E-01 -2.18E+00 -9.12E-01  1.40E-01  2.91E-02
  9  1.20E-01 -2.15E+00 -8.36E-01 -2.32E+00 -8.75E-01  1.68E-01  3.93E-02
 10  1.41E-01 -2.29E+00 -7.82E-01 -2.51E+00 -8.15E-01  2.19E-01  3.26E-02
 11  1.65E-01 -2.43E+00 -7.28E-01 -2.69E+00 -7.38E-01  2.54E-01  1.05E-02
 12  1.83E-01 -2.56E+00 -6.85E-01 -2.79E+00 -6.80E-01  2.36E-01  4.87E-03
 13  2.01E-01 -2.69E+00 -6.40E-01 -2.89E+00 -6.12E-01  2.02E-01  2.72E-02
 14  2.27E-01 -2.84E+00 -5.84E-01 -2.98E+00 -5.12E-01  1.49E-01  7.16E-02
 15  2.44E-01 -2.96E+00 -5.35E-01 -3.03E+00 -4.40E-01  6.93E-02  9.58E-02
 16  2.62E-01 -3.08E+00 -4.82E-01 -3.05E+00 -3.57E-01  3.03E-02  1.25E-01
 17  2.70E-01 -3.16E+00 -4.44E-01 -3.06E+00 -3.20E-01  1.06E-01  1.24E-01
 18  2.79E-01 -3.24E+00 -3.93E-01 -3.06E+00 -2.76E-01  1.88E-01  1.17E-01
 19  2.87E-01 -3.30E+00 -3.29E-01 -3.05E+00 -2.40E-01  2.48E-01  8.94E-02
 20  2.98E-01 -3.34E+00 -2.29E-01 -3.04E+00 -1.83E-01  2.96E-01  4.59E-02
         .         .         .         .         .         .         .
         .         .         .         .         .         .         .
         .         .         .         .         .         .         .
147  1.57E+00 -2.77E-01 -1.83E+00 -2.53E-01 -1.83E+00  2.39E-02  3.45E-03
148  1.59E+00 -4.00E-01 -1.81E+00 -4.03E-01 -1.80E+00  2.75E-03  1.10E-02
149  1.60E+00 -5.17E-01 -1.78E+00 -5.41E-01 -1.76E+00  2.31E-02  1.56E-02
150  1.61E+00 -6.30E-01 -1.74E+00 -6.70F-01 -1.73E+00  3.93E-02  1.54E-02
151  1.63E+00 -7.50E-01 -1.69E+00 -8.14E-01 -1.68E+00  6.40E-02  1.64E-02
152  1.64E+00 -8.42E-01 -1.64E+00 -8.99E-01 -1.64E+00  5.71E-02  5.52E-03
153  1.65E+00 -9.32E-01 -1.59E+00 -9.79E-01 -1.60E+00  4.76E-02  6.28E-03
154  1.66E+00 -1.02E+00 -1.53E+00 -1.05E+00 -1.55E+00  3.43E-02  1.76E-02
155  1.68E+00 -1.11E+00 -1.46E+00 -1.13E+00 -1.49E+00  1.84E-02  2.33E-02
156  1.70E+00 -1.23E+00 -1.37E+00 -1.22E+00 -1.38E+00  5.11E-03  2.82E-03
157  1.71E+00 -1.31E+00 -1.30E+00 -1.26E+00 -1.29E+00  5.81E-02  6.92E-03

MAXIMUM RESIDUAL =       3.0704053E-01
```

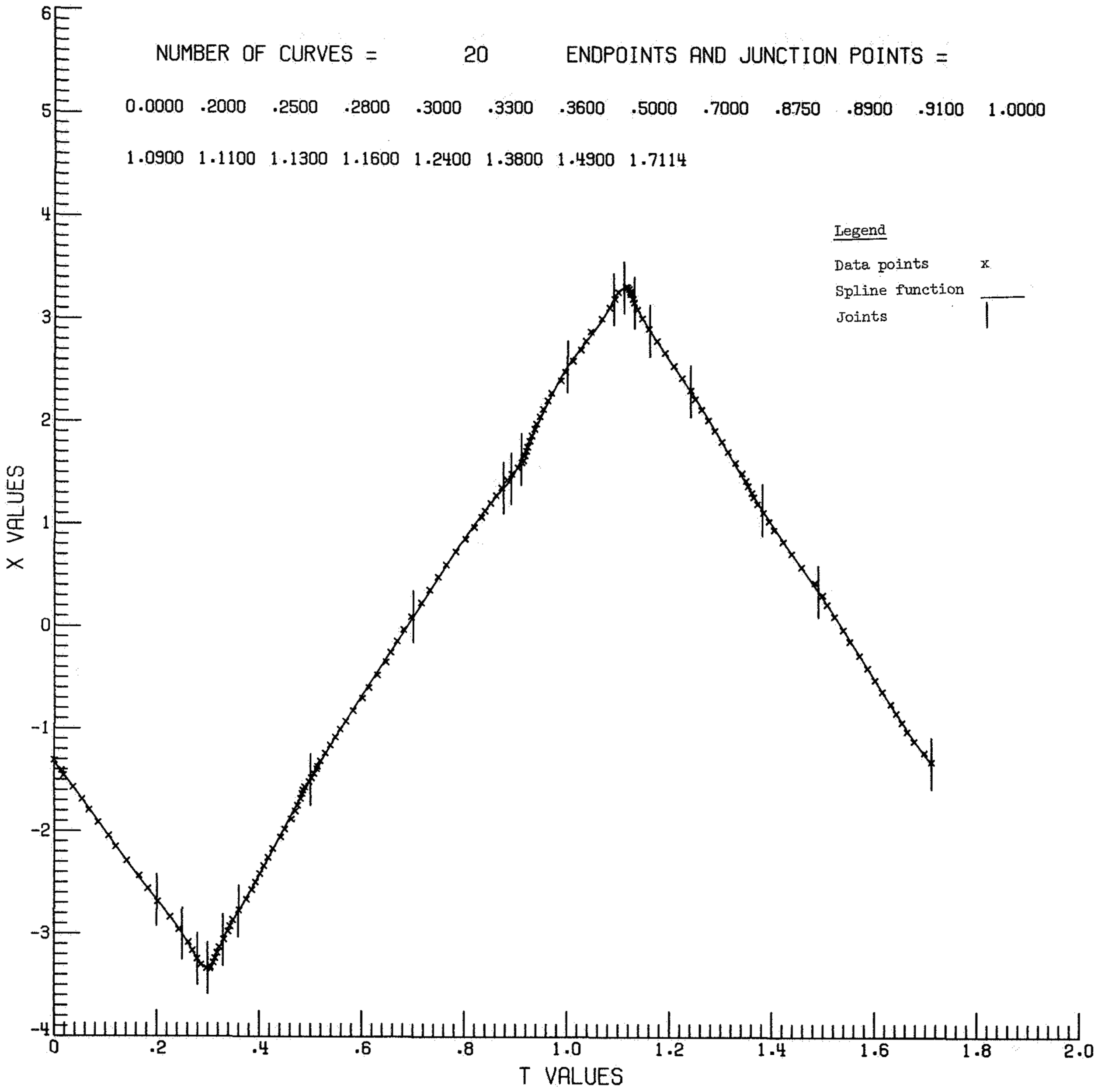
A better fit can be realized by changing the joints for $S_{\Delta_x}(t)$ to the following values:

```
NKR      =  20,

R        =  0.0,  0.2E+00,  0.25E+00,  0.28E+00,  0.3E+C0,  0.33E+00,  0.36E+00,
            0.5E+0C,  0.7E+00,  0.875E+00,  0.89E+0C,  0.91E+0C,  0.1E+01,
            0.1C9E+01,  0.111E+01,  0.113E+01,  C.116E+01,  0.124E+01,
            C.138E+01,  0.149E+01,  0.17114073E+01,
```

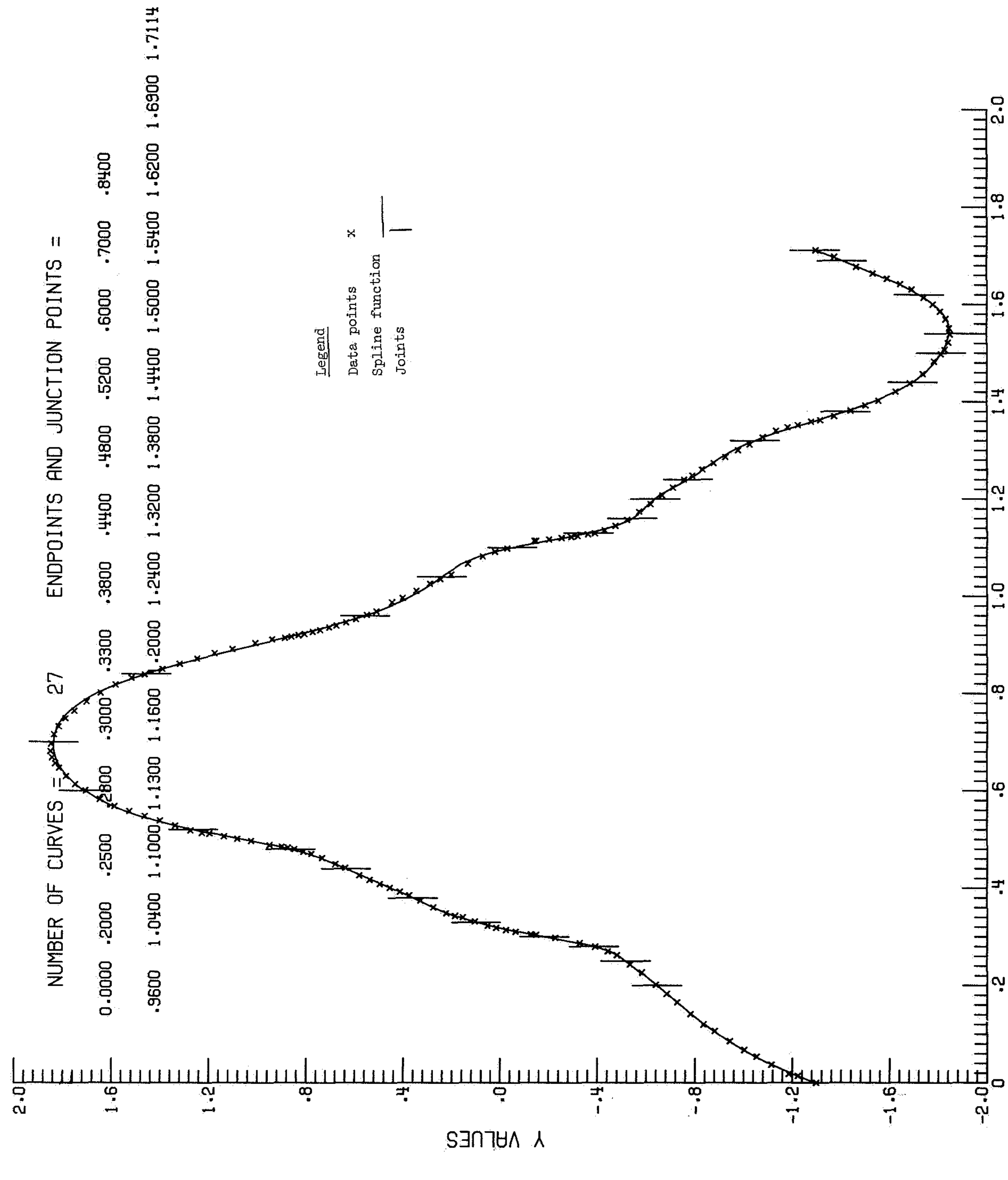
and changing the joints for $S_{\Delta_y}(t)$ to the following values:

```
NKR      =  27,

R        =  0.0,  C.2E+00,  0.25E+00,  0.28E+00,  0.3E+00,  0.33E+00,  0.38E+00,
            0.44E+00,  0.48E+CC,  0.52E+CC,  0.6E+00,  0.7E+00,  0.84E+00,
            C.96E+CC,  0.104E+01,  0.11E+01,  0.113E+01,  0.116E+01,  0.12E+01,
            0.124E+01,  0.132E+01,  0.138E+01,  C.144E+C1,  0.15E+01,
            0.154E+01,  0.162E+01,  0.169E+01,  0.17114073E+01,
```

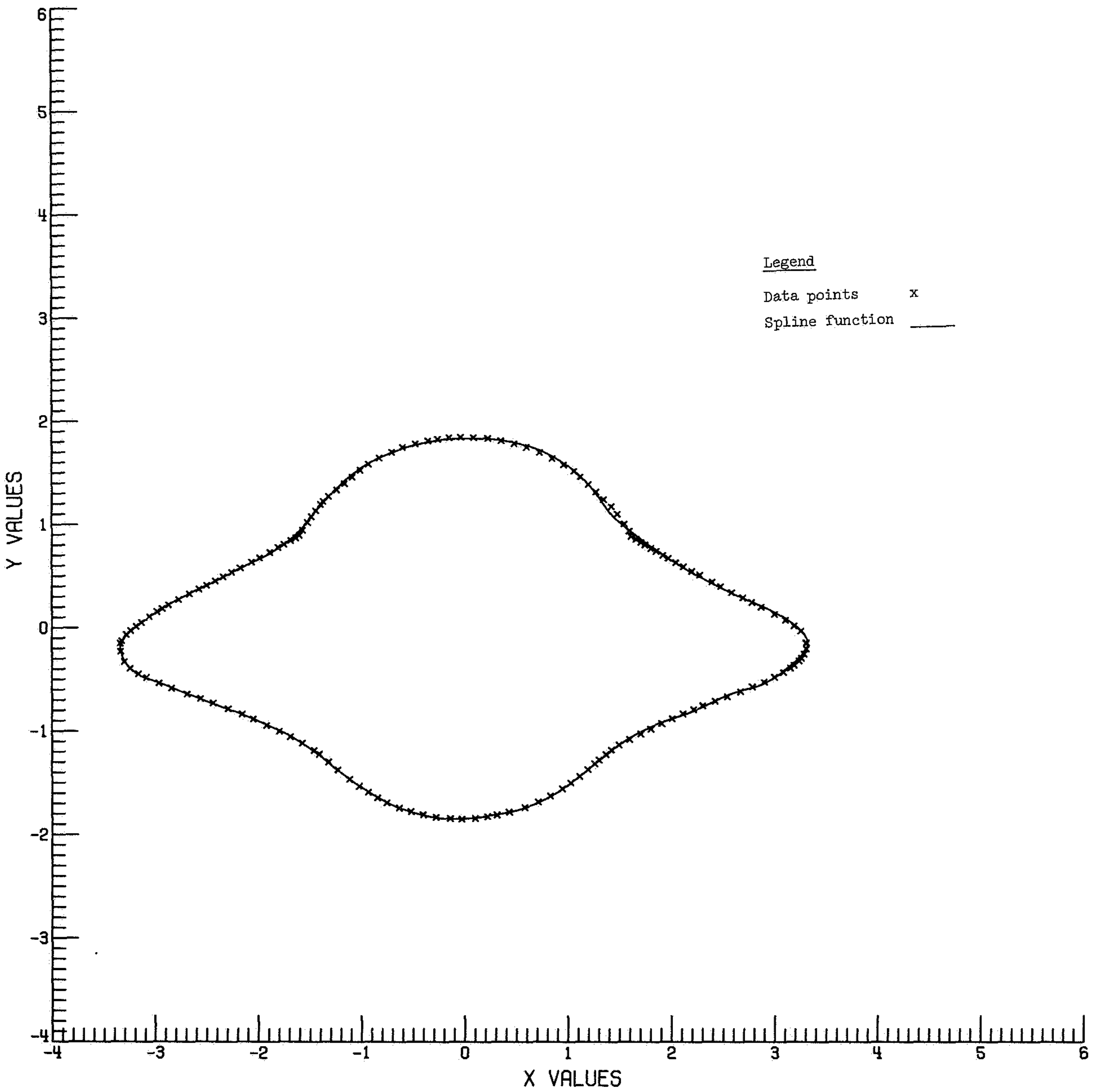
The output with these improved spline curves is as follows:


NUMBER OF CURVES = 20 ENDPOINTS AND JUNCTION POINTS =
0.0000 .2000 .2500 .2800 .3000 .3300 .3600 .5000 .7000 .8750 .8900 .9100 1.0000
1.0900 1.1100 1.1300 1.1600 1.2400 1.3800 1.4900 1.7114
Legend
Data points x
Spline function ____
Joints |
X VALUES
6
5
4
3
2
1
0
-1
-2
-3
-4
0
.2
.4
.6
.8
1.0
1.2
1.4
1.6
1.8
2.0
T VALUES

```
        ***************
        DATA FOR X VS T
        ***************


NUMBER OF CURVES =  20          NUMBER OF POINTS = 157
ENDPOINTS AND JUNCTION POINTS =
          0.                2.0000000E-01       2.5000000E-01       2.8000000E-01       3.0000000E-01       3.3000000E-01
          3.6000000E-01     5.0000000E-01       7.0000000E-01       8.7500000E-01       8.9000000E-01       9.1000000E-01
          1.0000000E+00     1.0900000E+00       1.1100000E+00       1.1300000E+00       1.1600000E+00       1.2400000E+00
          1.3800000E+00     1.4900000E+00       1.7114073E+00

                T                  X               COMPUTED X          RESIDUALS
   1      0.               -1.3143000E+00      -1.3308337E+00       1.6533691E-02
   2      1.4147770E-02    -1.4067000E+00      -1.4205916E+00       1.3891582E-02
   3      1.8061700E-02    -1.4570000E+00      -1.4459161E+00      -1.1083911E-02
   4      3.6571560E-02    -1.5725000E+00      -1.5681561E+00      -4.3438975E-03
   5      5.3955560E-02    -1.6897000E+00      -1.6859911E+00      -3.7088768E-03
   6      6.7455490E-02    -1.7940000E+00      -1.7789511E+00      -1.5048857E-02
   7      8.5455940E-02    -1.9143000E+00      -1.9041071E+00      -1.0192926E-02
   8      1.0601444E-01    -2.0442000E+00      -2.0475878E+00       3.3878043E-03
   9      1.1980981E-01    -2.1521000E+00      -2.1435005E+00      -8.5994835E-03
                .                  .                   .                   .
                .                  .                   .                   .
                .                  .                   .                   .
 152      1.6415309E+00    -8.4160000E-01      -8.4159103E-01      -8.9668784E-06
 153      1.6524367E+00    -9.3170000E-01      -9.2375797E-01      -7.9420259E-03
 154      1.6637800E+00    -1.0207000E+00      -1.0073254E+00      -1.3374592E-02
 155      1.6773444E+00    -1.1146000E+00      -1.1043325E+00      -1.0267510E-02
 156      1.6976671E+00    -1.2254000E+00      -1.2427150E+00       1.7315003E-02
 157      1.7114073E+00    -1.3143000E+00      -1.3308337E+00       1.6533691E-02

         STANDARD DEVIATION= 1.72912E-02

CURVE  1
            27.781371352(X**3)+          -7.873066482(X**2)+          -6.238488052(X)+          -1.330833691

CURVE  2
          -352.934906912(X**3)+         220.556700476(X**2)+         -51.924441443(X)+           1.714896535

        .
        .
        .

CURVE 20
            32.004001117(X**3)+        -151.159273734(X**2)+         229.940689879(X)+        -112.542534222
```


NUMBER OF CURVES = 27
ENDPOINTS AND JUNCTION POINTS =
0.0000 .2000 .2500 .2800 .3000 .3300 .3800 .4400 .4800 .5200 .6000 .7000 .8400
.9600 1.0400 1.1000 1.1300 1.1600 1.2000 1.2400 1.3200 1.3800 1.4400 1.5000 1.5400 1.6200 1.6900 1.7114
Legend
Data points x
Spline function
Joints
Y VALUES
2.0
1.6
1.2
.8
.4
.0
-.4
-.8
-1.2
-1.6
-2.0
T VALUES
0
.2
.4
.6
.8
1.0
1.2
1.4
1.6
1.8
2.0

```
               ***************
               DATA FOR Y VS T
               ***************

NUMBER OF CURVES =  27           NUMBER OF POINTS = 157
ENDPOINTS AND JUNCTION POINTS =
      0.                2.0000000E-01       2.5000000E-01       2.8000000E-01       3.0000000E-01       3.3000000E-01
      3.8000000E-01     4.4000000E-01       4.8000000E-01       5.2000000E-01       6.0000000E-01       7.0000000E-01
      8.4000000E-01     9.6000000E-01       1.0400000E+00       1.1000000E+00       1.1300000E+00       1.1600000E+00
      1.2000000E+00     1.2400000E+00       1.3200000E+00       1.3800000E+00       1.4400000E+00       1.5000000E+00
      1.5400000E+00     1.6200000E+00       1.6900000E+00       1.7114073E+00

            T                  Y               COMPUTED Y          RESIDUALS
  1   0.               -1.2983000E+00      -1.2952526E+00      -3.0473787E-03
  2   1.4147770E-02    -1.2234000E+00      -1.2218258E+00      -1.5741947E-03
  3   1.8061700E-02    -1.1862000E+00      -1.2026934E+00       1.6493358E-02
  4   3.6571560E-02    -1.1143000E+00      -1.1186202E+00       4.3202315E-03
  5   5.3955560E-02    -1.0539000E+00      -1.0484368E+00      -5.4632055E-03
  6   6.7455490E-02    -1.0027000E+00      -9.9908663E-01      -3.6133655E-03
  7   8.5455940E-02    -9.4330000E-01      -9.3933379E-01      -3.9662074E-03
  8   1.0601444E-01    -8.8260000E-01      -8.7812072E-01      -4.4792826E-03
  9   1.1980981E-01    -8.3620000E-01      -8.4039121E-01       4.1912142E-03
            .                  .                   .                   .
            .                  .                   .                   .
            .                  .                   .                   .
152   1.6415309E+00    -1.6446000E+00      -1.6397672E+00      -4.8327998E-03
153   1.6524367E+00    -1.5918000E+00      -1.5913531E+00      -4.4688381E-04
154   1.6637800E+00    -1.5333000E+00      -1.5377677E+00       4.4677228E-03
155   1.6773444E+00    -1.4644000E+00      -1.4707188E+00       6.3187759E-03
156   1.6976671E+00    -1.3747000E+00      -1.3675283E+00      -7.1716700E-03
157   1.7114073E+00    -1.2983000E+00      -1.2952526E+00      -3.0473787E-03

        STANDARD DEVIATION= 1.37491E-02

CURVE  1
              34.553310237(X**3)+       -17.816605078(X**2)+          5.435141266(X)+        -1.295252621

CURVE  2
             -32.232805608(X**3)+        22.255064429(X**2)+         -2.579192635(X)+        -.760963695

      .
      .
      .

CURVE 27
             247.340347753(X**3)+      -1253.767846843(X**2)+       2123.533508889(X)+       -1203.156536949
```


Legend
Data points x
Spline function ____
Y VALUES
X VALUES
6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4
5
6

```
           T           X           Y COMPUTED X COMPUTED Y  RES X      RES Y

  1  0.         -1.31E+00 -1.30E+00 -1.33E+00 -1.30E+00  1.65E-02  3.05E-03
  2  1.41E-02 -1.41E+00 -1.22E+00 -1.42E+00 -1.22E+00  1.39E-02  1.57E-03
  3  1.81E-02 -1.46E+00 -1.19E+00 -1.45E+00 -1.20E+00  1.11E-02  1.65E-02
  4  3.66E-02 -1.57E+00 -1.11E+00 -1.57E+00 -1.12E+00  4.34E-03  4.32E-03
  5  5.40E-02 -1.69E+00 -1.05E+00 -1.69E+00 -1.05E+00  3.71E-03  5.46E-03
  6  6.75E-02 -1.79E+00 -1.00E+00 -1.78E+00 -9.99E-01  1.50E-02  3.61E-03
  7  8.55E-02 -1.91E+00 -9.43E-01 -1.90E+00 -9.39E-01  1.02E-02  3.97E-03
  8  1.06E-01 -2.04E+00 -8.83E-01 -2.05E+00 -8.78E-01  3.39E-03  4.48E-03
  9  1.20E-01 -2.15E+00 -8.36E-01 -2.14E+00 -8.40E-01  8.60E-03  4.19E-03
 10  1.41E-01 -2.29E+00 -7.82E-01 -2.29E+00 -7.86E-01  1.95E-03  3.69E-03
 11  1.65E-01 -2.43E+00 -7.28E-01 -2.45E+00 -7.27E-01  1.89E-02  6.44E-04
 12  1.83E-01 -2.56E+00 -6.85E-01 -2.56E+00 -6.86E-01  5.78E-03  1.67E-03
 13  2.01E-01 -2.69E+00 -6.40E-01 -2.68E+00 -6.42E-01  7.94E-03  2.07E-03
 14  2.27E-01 -2.84E+00 -5.84E-01 -2.83E+00 -5.78E-01  3.10E-03  6.01E-03
 15  2.44E-01 -2.96E+00 -5.35E-01 -2.95E+00 -5.34E-01  8.58E-03  1.13E-03
 16  2.62E-01 -3.08E+00 -4.82E-01 -3.10E+00 -4.84E-01  1.44E-02  2.56E-03
 17  2.70E-01 -3.16E+00 -4.44E-01 -3.16E+00 -4.52E-01  9.14E-04  7.97E-03
 18  2.79E-01 -3.24E+00 -3.93E-01 -3.24E+00 -3.94E-01  3.25E-03  9.70E-04
 19  2.87E-01 -3.30E+00 -3.29E-01 -3.30E+00 -3.28E-01  5.59E-03  1.08E-03
 20  2.98E-01 -3.34E+00 -2.29E-01 -3.34E+00 -2.06E-01  2.46E-03  2.28E-02
        .         .         .         .         .          .         .
        .         .         .         .         .          .         .
        .         .         .         .         .          .         .
147  1.57E+00 -2.77E-01 -1.83E+00 -2.74E-01 -1.84E+00  3.28E-03  3.88E-03
148  1.59E+00 -4.00E-01 -1.81E+00 -4.00E-01 -1.81E+00  3.16E-04  4.41E-03
149  1.60E+00 -5.17E-01 -1.78E+00 -5.17E-01 -1.78E+00  7.35E-04  2.93E-03
150  1.61E+00 -6.30E-01 -1.74E+00 -6.29E-01 -1.74E+00  1.38E-03  9.19E-04
151  1.63E+00 -7.50E-01 -1.69E+00 -7.60E-01 -1.68E+00  1.01E-02  8.52E-03
152  1.64E+00 -8.42E-01 -1.64E+00 -8.42E-01 -1.64E+00  8.97E-06  4.83E-03
153  1.65E+00 -9.32E-01 -1.59E+00 -9.24E-01 -1.59E+00  7.94E-03  4.47E-04
154  1.66E+00 -1.02E+00 -1.53E+00 -1.01E+00 -1.54E+00  1.34E-02  4.47E-03
155  1.68E+00 -1.11E+00 -1.46E+00 -1.10E+00 -1.47E+00  1.03E-02  6.32E-03
156  1.70E+00 -1.23E+00 -1.37E+00 -1.24E+00 -1.37E+00  1.73E-02  7.17E-03
157  1.71E+00 -1.31E+00 -1.30E+00 -1.33E+00 -1.30E+00  1.65E-02  3.05E-03

MAXIMUM RESIDUAL =         5.4167443E-02
```

Case 3

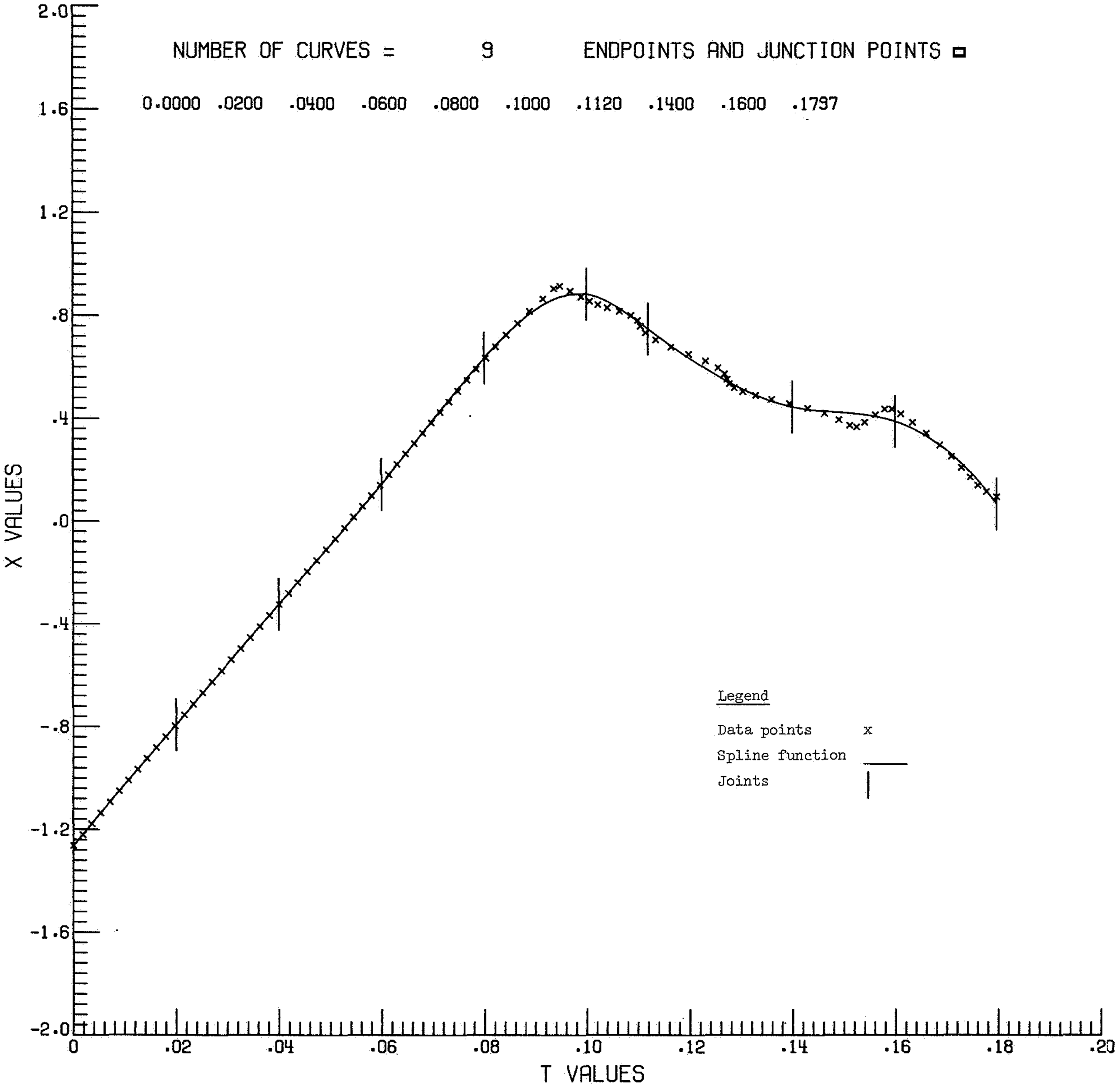
Case 3 is a parametric algorithm, arbitrary curve. The input with junction points for $S_{\Delta_X}(t)$ is as follows:

```
$NAM1

X         = -0.1263E+01, -0.122072E+01, -0.117843E+01, -0.113614E+01,
            -0.109383E+01, -0.10515E+01, -0.100915E+01, -0.96676E+00,
            -0.92434E+00, -0.88189E+00, -0.83938E+00, -0.79683E+00, -0.75423E+00,
            -0.71156E+00, -0.66883E+00, -0.62603E+00, -0.58316E+00, -0.54021E+00,
            -0.49719E+00, -0.45412E+00, -0.41104E+00, -0.36795E+00, -0.32489E+00,
            -0.28189E+00, -0.23896E+00, -0.19613E+00, -0.15343E+00, -0.11088E+00,
            -0.6851E-01, -0.2634E-01,  0.1561E-01,  0.5731E-01,  0.9873E-01,
             0.13986E+00,  0.18067E+00,  0.22124E+00,  0.26166E+00,  0.30202E+00,
             0.34242E+00,  0.38295E+00,  0.4237E+00,  0.46477E+00,  0.50625E+00,
             0.54824E+00,  0.59082E+00,  0.6341E+00,  0.67816E+00,  0.7231E+00,
             0.76901E+00,  0.81599E+00,  0.86401E+00,  0.90429E+00,  0.91447E+00,
             0.89488E+00,  0.87221E+00,  0.85497E+00,  0.84167E+00,  0.82972E+00,
             0.8162E+00,  0.79989E+00,  0.78051E+00,  0.75792E+00,  0.73271E+00,
             0.70571E+00,  0.67775E+00,  0.64967E+00,  0.62229E+00,  0.59644E+00,
             0.57297E+00,  0.55253E+00,  0.53474E+00,  0.5188E+00,  0.50396E+00,
             0.48942E+00,  0.47442E+00,  0.45816E+00,  0.43988E+00,  0.41878E+00,
             0.39439E+00,  0.37277E+00,  0.36649E+00,  0.38424E+00,  0.41364E+00,
             0.43579E+00,  0.43654E+00,  0.41777E+00,  0.3847E+00,  0.34252E+00,
             0.29646E+00,  0.25107E+00,  0.2086E+00,  0.17075E+00,  0.13925E+00,
             0.1147E+00,  0.94E-01,

YY        =  0.0, -0.19E-03, -0.36E-03, -0.5E-03, -0.61E-03, -0.65E-03,
            -0.62E-03, -0.51E-03, -0.3E-03,  0.3E-04,  0.49E-03,  0.109E-02,
             0.185E-02,  0.278E-02,  0.389E-02,  0.521E-02,  0.674E-02,
             0.85E-02,  0.1046E-01,  0.1256E-01,  0.1471E-01,  0.1686E-01,
             0.1893E-01,  0.2085E-01,  0.2254E-01,  0.2393E-01,  0.2496E-01,
             0.2554E-01,  0.2562E-01,  0.2511E-01,  0.2395E-01,  0.2206E-01,
             0.1937E-01,  0.1581E-01,  0.1135E-01,  0.618E-02,  0.56E-03,
            -0.521E-02, -0.1089E-01, -0.1618E-01, -0.2082E-01, -0.2454E-01,
            -0.2706E-01, -0.2811E-01, -0.2741E-01, -0.247E-01, -0.197E-01,
            -0.1214E-01, -0.175E-02,  0.1176E-01,  0.2867E-01,  0.5103E-01,
             0.8355E-01,  0.12427E+00,  0.16399E+00,  0.2014E+00,  0.23937E+00,
             0.28074E+00,  0.3274E+00,  0.37205E+00,  0.40202E+00,  0.40676E+00,
             0.38812E+00,  0.35237E+00,  0.30578E+00,  0.2546E+00,  0.20512E+00,
             0.16359E+00,  0.13628E+00,  0.12845E+00,  0.13925E+00,  0.16554E+00,
             0.20418E+00,  0.25203E+00,  0.30595E+00,  0.3628E+00,  0.41944E+00,
             0.47271E+00,  0.51962E+00,  0.55997E+00,  0.59635E+00,  0.6313E+00,
             0.66623E+00,  0.70223E+00,  0.73973E+00,  0.77696E+00,  0.81168E+00,
             0.84165E+00,  0.86463E+00,  0.87984E+00,  0.89169E+00,  0.90577E+00,
             0.9277E+00,  0.96068E+00,  0.1E+01,

NKR       =  9,

R         =  0.0,  0.2E-01,  0.4E-01,  0.6E-01,  0.8E-01,  0.1E+00,  0.112E+00,
             0.14E+00,  0.16E+00,  0.1797541E+00,

NMAT      =  95,
IFLAG    = -40306863533168859 7,

W        = -0.58944216614664E-07,  0.11527306719054-275, -0.23553149769543+140,
            0.24645084938579-104, -0.17365204637693-163, -0.35884981876725+135,
           -0.26496288110192-168,  0.25929515873337-229, -0.4043010315618-173,
           -0.91865312533783+137, -0.13142742952639-200,  0.20445924810045-231,
            0.13757355873905-275,  0.47168012250524-219,  0.48274593183088-219,
            0.4910452888251-219,   0.49934464581933-219,  0.50764400281356-219,
            0.52424271680201-219,  0.51970981213919-219,  0.53254207379624-219,
            0.54084143079047-219,  0.54914078778469-219,  0.54914078778469-219,
           -0.59272771575679E-07,  0.11527306719054-275, -0.39735813734271+147,
            0.11730037971322-275, -0.90457929503658E-12, -0.38071476425322-125,
            0.13228490334101-275, -0.45119336581121E-12, -0.59272316828328E-07,
            0.13186622350189-275, -0.15521444967826+145,  0.11730037971322-275,
           -0.90456194780182E-12, -0.38071476425322-125,  0.13228490334101-275,
           -0.45119336581121E-12, -0.60630411753296+142,  0.11730037971322-275,
           -0.90454807001401E-12, -0.38071476425322-125,  0.13228490334101-275,
           -0.45119336581121E-12, -0.23683663711337+140,  0.11730037971322-275,
           -0.90453419222262E-12, -0.38071476425322-125,  0.13228490334101-275,
           -0.45119336581121E-12, -0.92513956373211+137,  0.11730037971322-275,
           -0.90451684499144E-12, -0.38071476425322-125,  0.13228490334101-275,
           -0.45119336581121E-12, -0.39734289442987+147,  0.11730037971322-275,
           -0.15521147254684+145,  0.11730037971322-275, -0.90448561996888E-12,
           -0.38071476425322-125,  0.13228490334101-275, -0.45119336581121E-12,
           -0.90450296720364E-12, -0.38071476425322-125,  0.13228490334101-275,
           -0.45119336581121E-12, -0.60629248811337+142,  0.11730037971322-275,
           -0.90447174218107E-12, -0.38071476425322-125,  0.13228490334101-275,
           -0.45119336581121E-12, -0.23683209437134+140,  0.11730037971322-275,
           -0.90446133384021E-12, -0.38071476425322-125,  0.13228490334101-275,
           -0.45119336581121E-12, -0.92512181864607+137,  0.11730037971322-275,
           -0.9044474560524E-12, -0.38071476425322-125,  0.13228490334101-275,
           -0.45119336581121E-12, -0.39733527297344+147,  0.11730037971322-275,
           -0.15412913210712+145,  0.76435957915019+299, -0.35884771720364+135,
            0.10996484795022-123, -0.59196601394405E-07,

KG       =  4,

KLOSE    =  0,

$END
```

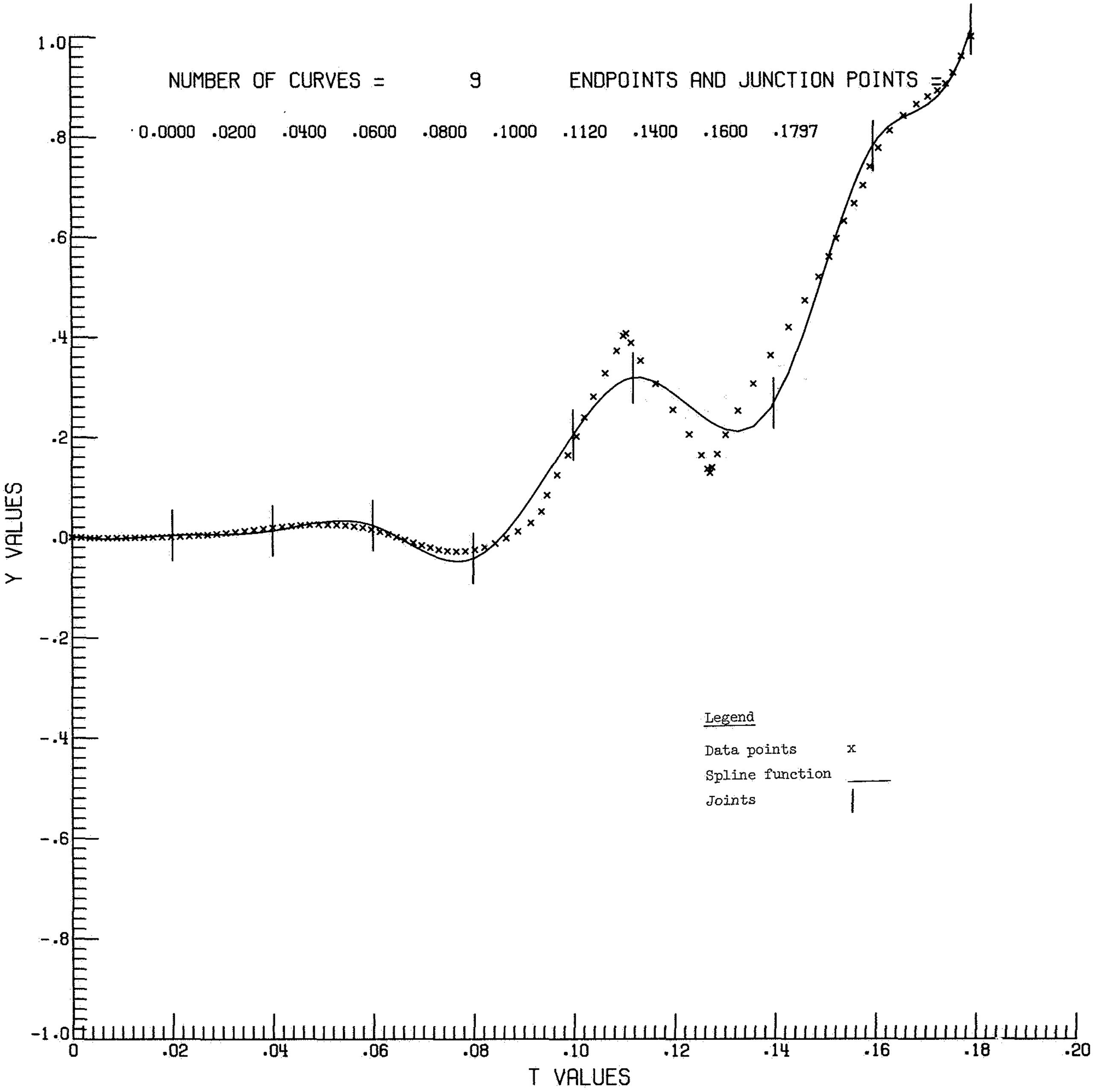
The additional input junction values for $S_{\Delta y}(t)$ are as follows:

```
$NAM1

NKR      =  9,

R        =  0.0,  0.2E-01,  0.4E-01,  0.6E-01,  0.8E-01,  0.1E+00,  0.112E+00,
            0.14E+00,  0.16E+00,  0.1797541E+00,

$END
```

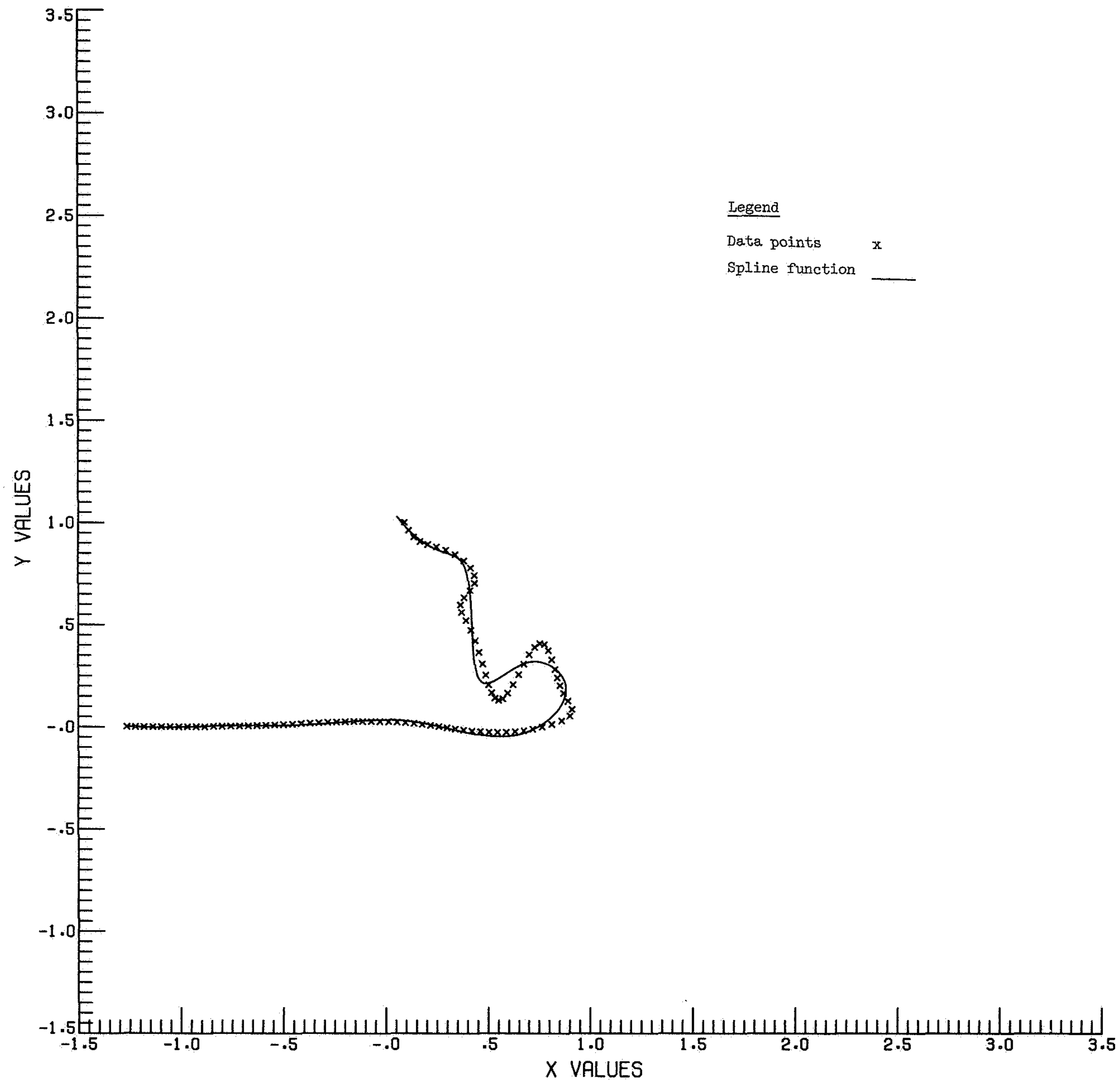
The output for this case is as follows:


NUMBER OF CURVES = 9
ENDPOINTS AND JUNCTION POINTS □
0.0000 .0200 .0400 .0600 .0800 .1000 .1120 .1400 .1600 .1797
2.0
1.6
1.2
.8
.4
.0
-.4
-.8
-1.2
-1.6
-2.0
X VALUES
0
.02
.04
.06
.08
.10
.12
.14
.16
.18
.20
T VALUES
Legend
Data points x
Spline function
Joints

```
        ***************
        DATA FOR X VS T
        ***************


NUMBER OF CURVES =   9          NUMBER OF POINTS =  95
ENDPOINTS AND JUNCTION POINTS =
      0.                 2.0000000E-02        4.0000000E-02        6.0000000E-02        8.0000000E-02        1.0000000E-01
      1.1200000E-01      1.4000000E-01        1.6000000E-01        1.7975410E-01

            T                  X               COMPUTED X            RESIDUALS
  1   0.                 -1.2630000E+00       -1.2638473E+00        8.4733050E-04
  2   1.7876345E-03      -1.2207200E+00       -1.2207474E+00        2.7383592E-05
  3   3.5761075E-03      -1.1784300E+00       -1.1779448E+00       -4.8516582E-04
  4   5.3645712E-03      -1.1361400E+00       -1.1354175E+00       -7.2245487E-04
  5   7.1547194E-03      -1.0938300E+00       -1.0930829E+00       -7.4710145E-04
  6   8.9465499E-03      -1.0515000E+00       -1.0508988E+00       -6.0123208E-04
  7   1.0740073E-02      -1.0091500E+00       -1.0088225E+00       -3.2751496E-04
  8   1.2536997E-02      -9.6676000E-01       -9.6677146E-01        1.1458540E-05
  9   1.4336498E-02      -9.2434000E-01       -9.2472216E-01        3.8215856E-04
            .                  .                    .                    .
            .                  .                    .                    .
            .                  .                    .                    .
 90   1.7104099E-01       2.5107000E-01        2.6022659E-01       -9.1565915E-03
 91   1.7298512E-01       2.0860000E-01        2.2465760E-01       -1.6057597E-02
 92   1.7461599E-01       1.7075000E-01        1.9152934E-01       -2.0779337E-02
 93   1.7608916E-01       1.3925000E-01        1.5894552E-01       -1.9695520E-02
 94   1.7777954E-01       1.1470000E-01        1.1836138E-01       -3.6613832E-03
 95   1.7975410E-01       9.4000000E-02        6.6503802E-02        2.7496198E-02

      STANDARD DEVIATION= 1.62458E-02

CURVE  1
           1242.716068766(X**3)+        -56.325568122(X**2)+         24.206762515(X)+          -1.263847330

CURVE  2
           -921.937666077(X**3)+         73.553655969(X**2)+         21.609178033(X)+          -1.246530101

      .
      .
      .

CURVE  9
          -3383.062036687(X**3)+       1190.230551909(X**2)+       -127.463734029(X)+           4.169722620
```


NUMBER OF CURVES = 9
ENDPOINTS AND JUNCTION POINTS =
0.0000 .0200 .0400 .0600 .0800 .1000 .1120 .1400 .1600 .1797
Legend
Data points x
Spline function
Joints
Y VALUES
1.0
.8
.6
.4
.2
.0
-.2
-.4
-.6
-.8
-1.0
T VALUES
0
.02
.04
.06
.08
.10
.12
.14
.16
.18
.20

```
          ***************
          DATA FOR Y VS T
          ***************

NUMBER OF CURVES =   9          NUMBER OF POINTS =  95
ENDPOINTS AND JUNCTION POINTS =
      0.               2.0000000E-02       4.0000000E-02       6.0000000E-02       8.0000000E-02       1.0000000E-01
      1.1200000E-01    1.4000000E-01       1.6000000E-01       1.7975410E-01

           T                 Y              COMPUTED Y          RESIDUALS
  1   0.               0.                  2.3129292E-03      -2.3129292E-03
  2   1.7876345E-03   -1.9000000E-04      -1.2547901E-04      -6.4520985E-05
  3   3.5761075E-03   -3.6000000E-04      -1.6845926E-03       1.3245926E-03
  4   5.3645712E-03   -5.0000000E-04      -2.4823795E-03       1.9823795E-03
  5   7.1547194E-03   -6.1000000E-04      -2.6380693E-03       2.0280693E-03
  6   8.9465499E-03   -6.5000000E-04      -2.2696910E-03       1.6196910E-03
  7   1.0740073E-02   -6.2000000E-04      -1.4959436E-03       8.7594355E-04
  8   1.2536997E-02   -5.1000000E-04      -4.3509735E-04      -7.4902648E-05
  9   1.4336498E-02   -3.0000000E-04       7.9250616E-04      -1.0925062E-03
            .                 .                   .                   .
            .                 .                   .                   .
            .                 .                   .                   .
 90   1.7104099E-01    8.7984000E-01       8.6413546E-01       1.5704539E-02
 91   1.7298512E-01    8.9169000E-01       8.8020927E-01       1.1480726E-02
 92   1.7461599E-01    9.0577000E-01       8.9948065E-01       6.2893453E-03
 93   1.7608916E-01    9.2770000E-01       9.2289605E-01       4.8039479E-03
 94   1.7777954E-01    9.6068000E-01       9.5841989E-01       2.2601089E-03
 95   1.7975410E-01    1.0000000E+00       1.0139942E+00      -1.3994181E-02

      STANDARD DEVIATION= 3.59662E-02

CURVE  1
        -3476.842730578(X**3)+        156.308334295(X**2)+        -1.632353253(X)+        .002312929

CURVE  2
         2602.593718057(X**3)+       -208.457852623(X**2)+         5.662970486(X)+       -.046322562

     .
     .
     .

CURVE  9
        63602.322967982(X**3)+     -31994.186201467(X**2)+      5369.352921124(X)+       -299.778527129
```


Legend
Data points x
Spline function ____
3.5
3.0
2.5
2.0
1.5
1.0
.5
-.0
-.5
-1.0
-1.5
Y VALUES
-1.5
-1.0
-.5
-.0
.5
1.0
1.5
2.0
2.5
3.0
3.5
X VALUES

```
             T            X            Y COMPUTED X COMPUTED Y  RES X       RES Y

  1  0.          -1.26E+00  0.          -1.26E+00  2.31E-03  8.47E-04  2.31E-03
  2  1.79E-03 -1.22E+00 -1.90E-04 -1.22E+00 -1.25E-04  2.74E-05  6.45E-05
  3  3.58E-03 -1.18E+00 -3.60E-04 -1.18E+00 -1.68E-03  4.85E-04  1.32E-03
  4  5.36E-03 -1.14E+00 -5.00E-04 -1.14E+00 -2.48E-03  7.22E-04  1.98E-03
  5  7.15E-03 -1.09E+00 -6.10E-04 -1.09E+00 -2.64E-03  7.47E-04  2.03E-03
  6  8.95E-03 -1.05E+00 -6.50E-04 -1.05E+00 -2.27E-03  6.01E-04  1.62E-03
  7  1.07E-02 -1.01E+00 -6.20E-04 -1.01E+00 -1.50E-03  3.28E-04  8.76E-04
  8  1.25E-02 -9.67E-01 -5.10E-04 -9.67E-01 -4.35E-04  1.15E-05  7.49E-05
  9  1.43E-02 -9.24E-01 -3.00E-04 -9.25E-01  7.93E-04  3.82E-04  1.09E-03
 10  1.61E-02 -8.82E-01  3.00E-05 -8.83E-01  2.07E-03  7.41E-04  2.04E-03
 11  1.79E-02 -8.39E-01  4.90E-04 -8.40E-01  3.26E-03  1.01E-03  2.77E-03
 12  1.98E-02 -7.97E-01  1.09E-03 -7.98E-01  4.26E-03  1.17E-03  3.17E-03
 13  2.16E-02 -7.54E-01  1.85E-03 -7.55E-01  4.96E-03  1.17E-03  3.11E-03
 14  2.34E-02 -7.12E-01  2.78E-03 -7.13E-01  5.39E-03  1.00E-03  2.61E-03
 15  2.52E-02 -6.69E-01  3.89E-03 -6.70E-01  5.66E-03  7.03E-04  1.77E-03
 16  2.71E-02 -6.26E-01  5.21E-03 -6.26E-01  5.84E-03  2.97E-04  6.34E-04
 17  2.89E-02 -5.83E-01  6.74E-03 -5.83E-01  6.05E-03  1.86E-04  6.89E-04
 18  3.07E-02 -5.40E-01  8.50E-03 -5.39E-01  6.38E-03  7.26E-04  2.12E-03
 19  3.26E-02 -4.97E-01  1.05E-02 -4.96E-01  6.92E-03  1.28E-03  3.54E-03
 20  3.45E-02 -4.54E-01  1.26E-02 -4.52E-01  7.78E-03  1.77E-03  4.78E-03
         .          .          .          .          .          .          .
         .          .          .          .          .          .          .
         .          .          .          .          .          .          .
 85  1.59E-01  4.37E-01  7.40E-01  3.92E-01  7.72E-01  4.44E-02  3.19E-02
 86  1.61E-01  4.18E-01  7.77E-01  3.81E-01  7.98E-01  3.71E-02  2.10E-02
 87  1.63E-01  3.85E-01  8.12E-01  3.61E-01  8.22E-01  2.34E-02  9.92E-03
 88  1.66E-01  3.43E-01  8.42E-01  3.32E-01  8.39E-01  1.01E-02  3.02E-03
 89  1.69E-01  2.96E-01  8.65E-01  2.97E-01  8.51E-01  4.07E-04  1.34E-02
 90  1.71E-01  2.51E-01  8.80E-01  2.60E-01  8.64E-01  9.16E-03  1.57E-02
 91  1.73E-01  2.09E-01  8.92E-01  2.25E-01  8.80E-01  1.61E-02  1.15E-02
 92  1.75E-01  1.71E-01  9.06E-01  1.92E-01  8.99E-01  2.08E-02  6.29E-03
 93  1.76E-01  1.39E-01  9.28E-01  1.59E-01  9.23E-01  1.97E-02  4.80E-03
 94  1.78E-01  1.15E-01  9.61E-01  1.18E-01  9.58E-01  3.66E-03  2.26E-03
 95  1.80E-01  9.40E-02  1.00E+00  6.65E-02  1.01E+00  2.75E-02  1.40E-02

MAXIMUM RESIDUAL =          1.0569066E-01
```

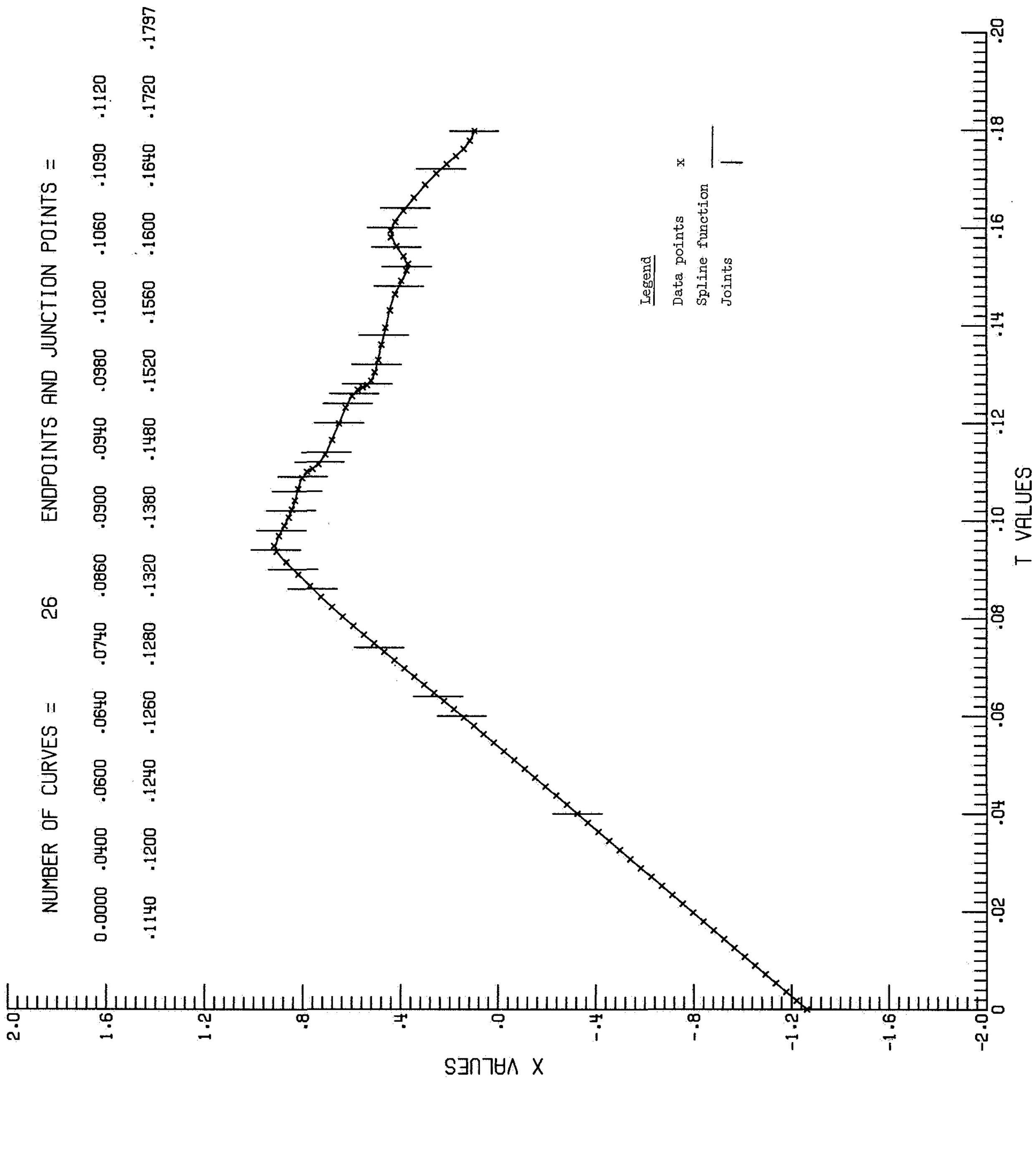
A better fit can be obtained by changing the joints for $S_{\Delta_x}(t)$ to the following values:

```
NKR       =  26,

R         =  0.0,  0.4E-01,  0.6E-01,  0.64E-01,  0.74E-01,  0.86E-01,  0.9E-01,
             0.94E-01,  0.98E-01,  0.102E+00,  0.106E+00,  0.109E+00,  0.112F+00,
             0.114E+00,  0.12E+00,  0.124E+00,  0.126E+00,  0.129E+00,
             0.132E+00,  0.138E+00,  0.148E+00,  0.152E+00,  0.156E+00,
             0.16E+00,  0.164E+00,  0.172F+00,  0.1797541E+00,
```

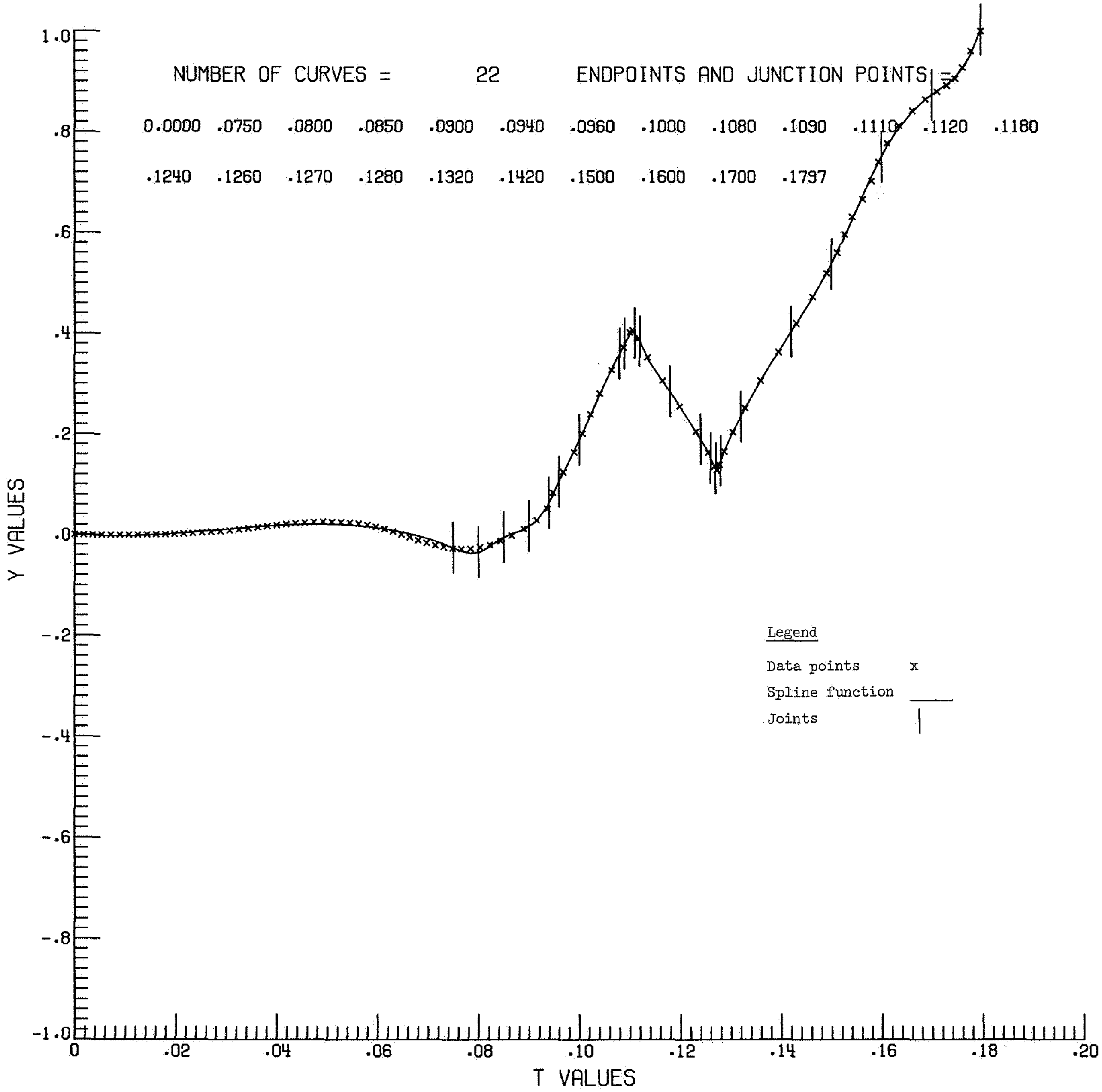
and changing the joints for $S_{\Delta_y}(t)$ to the following values:

```
NKR       =  22,

R         =  0.0,  0.75E-01,  0.8E-01,  0.85E-01,  0.9E-01,  0.94E-01,  0.96E-01,
             0.1E+00,  0.108E+00,  0.109E+00,  0.111E+00,  0.112E+00,  0.118E+00,
             0.124E+00,  0.126E+00,  0.127E+00,  0.128E+00,  0.132E+00,
             0.142E+00,  0.15E+00,  0.16E+00,  0.17E+00,  0.1797541E+00,
```

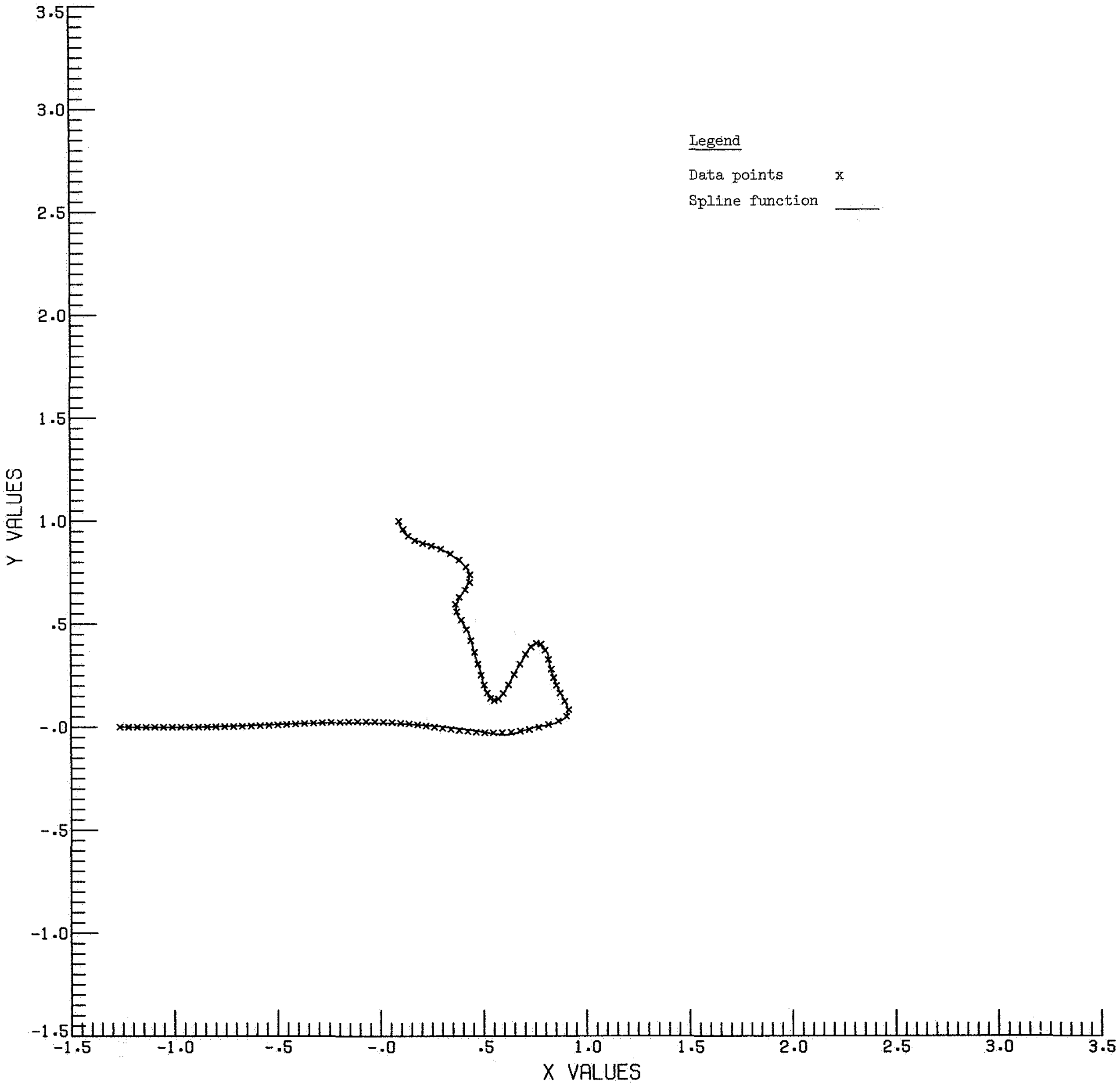
The output with these improved spline curves is as follows:


NUMBER OF CURVES = 26 ENDPOINTS AND JUNCTION POINTS =
0.0000 .0400 .0600 .0640 .0740 .0860 .0900 .0940 .0980 .1020 .1060 .1090 .1120
.1140 .1200 .1240 .1260 .1280 .1320 .1380 .1480 .1520 .1560 .1600 .1640 .1720 .1797
Legend
Data points x
Spline function
Joints
X VALUES
2.0
1.6
1.2
.8
.4
.0
-.4
-.8
-1.2
-1.6
-2.0
T VALUES
0
.02
.04
.06
.08
.10
.12
.14
.16
.18
.20

```
               ***************
               DATA FOR X VS T
               ***************


NUMBER OF CURVES =  26          NUMBER OF POINTS =  95
ENDPOINTS AND JUNCTION POINTS =
      0.               4.0000000E-02      6.0000000E-02      6.4000000E-02      7.4000000E-02      8.6000000E-02
      9.0000000E-02    9.4000000E-02      9.8000000E-02      1.0200000E-01      1.0600000E-01      1.0900000E-01
      1.1200000E-01    1.1400000E-01      1.2000000E-01      1.2400000E-01      1.2600000E-01      1.2800000E-01
      1.3200000E-01    1.3800000E-01      1.4800000E-01      1.5200000E-01      1.5600000E-01      1.6000000E-01
      1.6400000E-01    1.7200000E-01      1.7975410E-01

             T                 X              COMPUTED X          RESIDUALS
  1   0.               -1.2630000E+00     -1.2635442E+00      5.4421203E-04
  2   1.7876345E-03    -1.2207200E+00     -1.2209271E+00      2.0709078E-04
  3   3.5761075E-03    -1.1784300E+00     -1.1783817E+00     -4.8322527E-05
  4   5.3645712E-03    -1.1361400E+00     -1.1359242E+00     -2.1582856E-04
  5   7.1547194E-03    -1.0938300E+00     -1.0935104E+00     -3.1955436E-04
  6   8.9465499E-03    -1.0515000E+00     -1.0511367E+00     -3.6326156E-04
  7   1.0740073E-02    -1.0091500E+00     -1.0087990E+00     -3.5103670E-04
  8   1.2536997E-02    -9.6676000E-01     -9.6645305E-01     -3.0695273E-04
  9   1.4336498E-02    -9.2434000E-01     -9.2411468E-01     -2.2531822E-04
            .                  .                  .                  .
            .                  .                  .                  .
            .                  .                  .                  .
 90   1.7104099E-01     2.5107000E-01      2.5154592E-01     -4.7591732E-04
 91   1.7298512E-01     2.0860000E-01      2.0926127E-01     -6.6127270E-04
 92   1.7461599E-01     1.7075000E-01      1.7180109E-01     -1.0510943E-03
 93   1.7608916E-01     1.3925000E-01      1.4039198E-01     -1.1419846E-03
 94   1.7777954E-01     1.1470000E-01      1.1176822E-01      2.9317774E-03
 95   1.7975410E-01     9.4000000E-02      9.5004800E-02     -1.0047997E-03

        STANDARD DEVIATION= 1.87069E-03

CURVE  1
             117.502094970(X**3)+        -14.967494159(X**2)+          23.866333235(X)+          -1.263544212

CURVE  2
             947.281273489(X**3)+       -114.540995581(X**2)+          27.849273291(X)+          -1.316650079

      .
      .
      .

CURVE 26
          178475.198454929(X**3)+     -92934.556033424(X**2)+       16107.662286888(X)+        -929.072397329
```


NUMBER OF CURVES = 22
ENDPOINTS AND JUNCTION POINTS =
0.0000 .0750 .0800 .0850 .0900 .0940 .0960 .1000 .1080 .1090 .1110 .1120 .1180
.1240 .1260 .1270 .1280 .1320 .1420 .1500 .1600 .1700 .1797
1.0
.8
.6
.4
.2
.0
-.2
-.4
-.6
-.8
-1.0
Y VALUES
0
.02
.04
.06
.08
.10
.12
.14
.16
.18
.20
T VALUES
Legend
Data points x
Spline function
Joints

```
          ***************
          DATA FOR  Y VS T
          ***************


NUMBER OF CURVES =   22         NUMBER OF POINTS =  95
ENDPOINTS AND JUNCTION POINTS =
      0.               7.5000000E-02    8.0000000E-02    8.5000000E-02    9.0000000E-02    9.4000000E-02
      9.6000000E-02    1.0000000E-01    1.0800000E-01    1.0900000E-01    1.1100000E-01    1.1200000E-01
      1.1800000E-01    1.2400000E-01    1.2600000E-01    1.2700000E-01    1.2800000E-01    1.3200000E-01
      1.4200000E-01    1.5000000E-01    1.6000000E-01    1.7000000E-01    1.7975410E-01

,            T                Y            COMPUTED Y        RESIDUALS
  1   0.               0.               1.5320325E-03   -1.5320325E-03
  2   1.7876345E-03   -1.9000000E-04    3.9417847E-05   -2.2941785E-04
  3   3.5761075E-03   -3.6000000E-04   -1.0670757E-03    7.0707567E-04
  4   5.3645712E-03   -5.0000000E-04   -1.8126380E-03    1.3126380E-03
  5   7.1547194E-03   -6.1000000E-04   -2.2235004E-03    1.6135004E-03
  6   8.9465499E-03   -6.5000000E-04   -2.3248133E-03    1.6748133E-03
  7   1.0740073E-02   -6.2000000E-04   -2.1418727E-03    1.5218727E-03
  8   1.2536997E-02   -5.1000000E-04   -1.6995836E-03    1.1895836E-03
  9   1.4336498E-02   -3.0000000E-04   -1.0232663E-03    7.2326630E-04
            .                .                .                .
            .                .                .                .
            .                .                .                .
 90   1.7104099E-01    8.7984000E-01    8.8164832E-01   -1.8083236E-03
 91   1.7298512E-01    8.9169000E-01    8.9466793E-01   -2.9779292E-03
 92   1.7461599E-01    9.0577000E-01    9.0856772E-01   -2.7977183E-03
 93   1.7608916E-01    9.2770000E-01    9.2597900E-01    1.7209984E-03
 94   1.7777954E-01    9.6068000E-01    9.5443434E-01    6.2456557E-03
 95   1.7975410E-01    1.0000000E+00    1.0032033E+00   -3.2033132E-03

        STANDARD DEVIATION= 3.65209E-03

CURVE  1
           -757.131197827(X**3)+        64.541828409(X**2)+          -.947924005(X)+           .001532032

CURVE  2
          107061.539534349(X**3)+    -24194.659086331(X**2)+       1818.492144601(X)+        -45.484469683

      .
      .
      .

CURVE 22
          109157.471800541(X**3)+    -56102.880710884(X**2)+       9618.034924860(X)+       -549.108645273
```


Legend
Data points x
Spline function ____
Y VALUES
3.5
3.0
2.5
2.0
1.5
1.0
.5
-.0
-.5
-1.0
-1.5
X VALUES
-1.5
-1.0
-.5
-.0
.5
1.0
1.5
2.0
2.5
3.0
3.5

```
           T          X          Y COMPUTED X COMPUTED Y  RES X     RES Y

  1  0.        -1.26E+00  0.        -1.26E+00  1.53E-03  5.44E-04  1.53E-03
  2  1.79E-03 -1.22E+00 -1.90E-04 -1.22E+00  3.94E-05  2.07E-04  2.29E-04
  3  3.58E-03 -1.18E+00 -3.60E-04 -1.18E+00 -1.07E-03  4.83E-05  7.07E-04
  4  5.36E-03 -1.14E+00 -5.00E-04 -1.14E+00 -1.81E-03  2.16E-04  1.31E-03
  5  7.15E-03 -1.09E+00 -6.10E-04 -1.09E+00 -2.22E-03  3.20E-04  1.61E-03
  6  8.95E-03 -1.05E+00 -6.50E-04 -1.05E+00 -2.32E-03  3.63E-04  1.67E-03
  7  1.07E-02 -1.01E+00 -6.20E-04 -1.01E+00 -2.14E-03  3.51E-04  1.52E-03
  8  1.25E-02 -9.67E-01 -5.10E-04 -9.66E-01 -1.70E-03  3.07E-04  1.19E-03
  9  1.43E-02 -9.24E-01 -3.00E-04 -9.24E-01 -1.02E-03  2.25E-04  7.23E-04
 10  1.61E-02 -8.82E-01  3.00E-05 -8.82E-01 -1.38E-04  1.11E-04  1.68E-04
 11  1.79E-02 -8.39E-01  4.90E-04 -8.39E-01  9.31E-04  2.14E-06  4.41E-04
 12  1.98E-02 -7.97E-01  1.09E-03 -7.97E-01  2.16E-03  1.28E-04  1.07E-03
 13  2.16E-02 -7.54E-01  1.85E-03 -7.54E-01  3.52E-03  2.52E-04  1.67E-03
 14  2.34E-02 -7.12E-01  2.78E-03 -7.12E-01  4.98E-03  3.49E-04  2.20E-03
 15  2.52E-02 -6.69E-01  3.89E-03 -6.69E-01  6.53E-03  4.22E-04  2.64E-03
 16  2.71E-02 -6.26E-01  5.21E-03 -6.26E-01  8.13E-03  4.54E-04  2.92E-03
 17  2.89E-02 -5.83E-01  6.74E-03 -5.84E-01  9.76E-03  4.39E-04  3.02E-03
 18  3.07E-02 -5.40E-01  8.50E-03 -5.41E-01  1.14E-02  3.59E-04  2.89E-03
 19  3.26E-02 -4.97E-01  1.05E-02 -4.97E-01  1.30E-02  2.19E-04  2.53E-03
 20  3.45E-02 -4.54E-01  1.26E-02 -4.54E-01  1.45E-02  4.06E-05  1.96E-03
         .         .         .         .         .         .         .
         .         .         .         .         .         .         .
         .         .         .         .         .         .         .
 85  1.59E-01  4.37E-01  7.40E-01  4.34E-01  7.40E-01  2.20E-03  1.49E-04
 86  1.61E-01  4.18E-01  7.77E-01  4.22E-01  7.72E-01  4.21E-03  5.40E-03
 87  1.63E-01  3.85E-01  8.12E-01  3.86E-01  8.07E-01  1.70E-03  4.58E-03
 88  1.66E-01  3.43E-01  8.42E-01  3.40E-01  8.40E-01  2.06E-03  1.45E-03
 89  1.69E-01  2.96E-01  8.65E-01  2.95E-01  8.65E-01  1.49E-03  6.08E-04
 90  1.71E-01  2.51E-01  8.80E-01  2.52E-01  8.82E-01  4.76E-04  1.81E-03
 91  1.73E-01  2.09E-01  8.92E-01  2.09E-01  8.95E-01  6.61E-04  2.98E-03
 92  1.75E-01  1.71E-01  9.06E-01  1.72E-01  9.09E-01  1.05E-03  2.80E-03
 93  1.76E-01  1.39E-01  9.28E-01  1.40E-01  9.26E-01  1.14E-03  1.72E-03
 94  1.78E-01  1.15E-01  9.61E-01  1.12E-01  9.54E-01  2.93E-03  6.25E-03
 95  1.80E-01  9.40E-02  1.00E+00  9.50E-02  1.00E+00  1.00E-03  3.20E-03

MAXIMUM RESIDUAL =        9.6992911E-03
```

REFERENCES

1. De Boor, Carl; and Rice, John R.: Least Squares Cubic Spline Approximation I – Fixed Knots. CSD TR 20, Purdue Univ., Apr. 1968.

2. De Boor, Carl; and Rice, John R.: Least Squares Cubic Spline Approximation II – Variable Knots. CSD TR 21, Purdue Univ., Apr. 1968.

3. Smith, Patricia J.: FITLOS: A FORTRAN Program for Fitting Low-Order Polynomial Splines by the Method of Least Squares. NASA TN D-6401, 1971.

4. Ahlberg, J. H.; Nilson, E. N.; and Walsh, J. L.: The Theory of Splines and Their Applications. Academic Press, Inc., 1967.

5. Bliss, Gilbert A.: Lectures on the Calculus of Variations. Univ. of Chicago Press, c.1946.

6. Guest, P. G.: Numerical Methods of Curve Fitting. Cambridge Univ. Press, 1961.

7. Hadley, G.: Nonlinear and Dynamic Programming. Addison-Wesley Pub. Co., Inc., c.1964, pp. 212-213.

NASA-Langley, 1974 L-8945

> *"The aeronautical and space activities of the United States shall be conducted so as to contribute . . . to the expansion of human knowledge of phenomena in the atmosphere and space. The Administration shall provide for the widest practicable and appropriate dissemination of information concerning its activities and the results thereof."*
>
> —National Aeronautics and Space Act of 1958

NASA SCIENTIFIC AND TECHNICAL PUBLICATIONS

TECHNICAL REPORTS: Scientific and technical information considered important, complete, and a lasting contribution to existing knowledge.

TECHNICAL NOTES: Information less broad in scope but nevertheless of importance as a contribution to existing knowledge.

TECHNICAL MEMORANDUMS: Information receiving limited distribution because of preliminary data, security classification, or other reasons. Also includes conference proceedings with either limited or unlimited distribution.

CONTRACTOR REPORTS: Scientific and technical information generated under a NASA contract or grant and considered an important contribution to existing knowledge.

TECHNICAL TRANSLATIONS: Information published in a foreign language considered to merit NASA distribution in English.

SPECIAL PUBLICATIONS: Information derived from or of value to NASA activities. Publications include final reports of major projects, monographs, data compilations, handbooks, sourcebooks, and special bibliographies.

TECHNOLOGY UTILIZATION PUBLICATIONS: Information on technology used by NASA that may be of particular interest in commercial and other non-aerospace applications. Publications include Tech Briefs, Technology Utilization Reports and Technology Surveys.

Details on the availability of these publications may be obtained from:

SCIENTIFIC AND TECHNICAL INFORMATION OFFICE

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Washington, D.C. 20546