AAEC/E253

AAEC/E253

AUSTRALIAN ATOMIC ENERGY COMMISSION

RESEARCH ESTABLISHMENT
LUCAS HEIGHTS

SPLINS & SMOOTH
TWO FORTRAN SMOOTHING ROUTINES

by

J.M. BARRY

January 1973

ISBN 0 642 99532 X

AUSTRALIAN ATOMIC ENERGY COMMISSION

RESEARCH ESTABLISHMENT

LUCAS HEIGHTS

SPLINS & SMOOTH

TWO FORTRAN SMOOTHING ROUTINES

by

J. M. BARRY

ABSTRACT

Two FORTRAN subroutines are presented that determine interpolating functions for experimental data. The first produces a spline where the smoothness of the function and the degree to which it fits the data is determined by the user. The second allows subdivision of the data, with individual polynomials being used to approximate the data on each interval and the polynomials are constrained to be piecewise continuous along with their first derivatives. An option to make the second derivative continuous is also available.

National Library of Australia card number and ISBN 0 642 99532 X

The following descriptors have been selected from the INIS Thesaurus to describe the subject content of this report for information retrieval purposes. For further details please refer to IAEA–INIS–12 (INIS: Manual for Indexing) and IAEA–INIS–13 (INIS: Thesaurus) published in Vienna by the International Atomic Energy Agency.

DATA; FORTRAN; INTERPOLATION; LEAST SQUARE FIT; POLYNOMIALS; S CODES

CONTENTS

1. INTRODUCTION

Two FORTRAN subroutines SPLINS and SMOOTH are described that allow smooth approximations to be made to experimental data. The first approximates the data on each segment with a cubic spline that rather than minimize 'curvature' alone as with the more conventional spline, seeks the minimization of 'curvature' and least squares error combined. The second allows the user to subdivide the data into a number of segments and cubic polynomials are fitted by least squares technique subject to the constraint that they be piecewise continuous together with their derivatives of up to the second order at each segment joint.

2. SUBROUTINE SPLINS

2.1 Definition of the Interpolating Spline

In normal cubic spline approximation we consider an interval $a \leq x \leq b$ to be subdivided into a set of mesh points.

$$\Delta \; : \; a = x_o < x_1 < \ldots < x_n = b \quad ,$$

to which is associated a set of ordinates,

$$Y \; : \; y_o, y_1, \ldots, y_n \quad .$$

A function $S_\Delta(x)$ which is continuous together with its first and second derivatives on $[a, b]$, and which coincides with a cubic in each subinterval $x_{j-1} \leq x \leq x_j$ $(j = 1,2, \ldots, n)$ and satisfies $S_\Delta(x_j) = y_j (j = 0,1, \ldots, n)$ is said to be a spline of order 3. (The third derivative undergoes step changes on the set of mesh points Δ.) For various specific conditions applied to the first and second derivatives at $x = a$ and b such splines are known to possess properties of smoothness; that is they may be shown (Ahlberg, Nilson and Walsh 1967) for all functions $f(x)$ such that $f(x_j) = y_j$ $(j = 0,1, \ldots, n)$ to possess an absolutely continuous first derivative on $[a,b]$ and a second derivative in $L^2(a,b)$ to minimize $\int_a^b \{f''(x)\}^2 dx$. Considering experimental data where errors exist in the observations $y_j (j = 0, 1, \ldots, n)$ we seek a 'smooth least squares fit' function $\phi(x)$ that has its first derivative absolutely continuous on $[a,b]$ and a second derivative in $L^2(a,b)$ that minimizes

$$\int_a^b \left\{\phi''(x)\right\}^2 dx + \sum_{j=0}^{n} \lambda_j \left\{\phi(x_j) - y_j\right\}^2 \quad , \qquad \ldots(1)$$

where the λ_j are chosen to obtain a balance between the smoothness of the function and the extent to which it fits the data. Such a function can be shown (Hanscomb 1965) to possess the descriptive conditions given for the

spline of order 3 but instead of $\phi(x_j) = y_j$ (j=0,1, . . . , n) the condition

$$\phi'''(x_{j+}) - \phi'''(x_{j-}) = -\lambda_j \left\{ \phi(x_j) - y_j \right\} \qquad \ldots(2)$$

applies.

2.2 Formulation of the Interpolating Spline

Let $S_\Delta(x)$ denote a spline possessing all the properties (2) and let M_j denote the 'moment' $S_\Delta''(x_j)$ of $S_\Delta(x)$ at x_j (j=0,1, . . . , n). From the linearity of the second derivative on x_{j-1}, x_j

$$S_\Delta''(x) = \frac{M_{j-1}}{6} \frac{x_j - x}{h_j} + \frac{M_j}{6} \frac{x - x_{j-1}}{h_j}$$

where $h_j = x_j - x_{j-1}$. On integration twice we obtain

$$S_\Delta(x) = M_{j-1} \frac{(x_j - x)^3}{h_j} + M_j \frac{(x - x_{j-1})^3}{h_j} + A_{j-1} x + B_{j-1} \text{ for } x_{j-1} \leqslant x \leqslant x_j \qquad \ldots(3)$$

To evaluate the constants of integration the continuity condition (2) is employed at $x=x_j$ on $[x_{j-1}, x_j]$ to give

$$\frac{M_{j+1}}{h_{j+1}} - \frac{M_j}{h_{j+1}} + \frac{M_{j-1}}{h_j} - \frac{M_j}{h_j} = -\lambda_j \left[\frac{M_j h_j^2}{6} + A_{j-1} x_j + B_{j-1} - y_{j-1} \right].$$

Similarly at $x=x_{j-1}$ on $[x_{j-1}, x_j]$

$$\frac{M_j}{h_j} - \frac{M_{j-1}}{h_j} + \frac{M_{j-2}}{h_{j-1}} - \frac{M_{j-1}}{h_{j-1}} = -\lambda_{j-1} \left[\frac{M_{j-1} h_j^2}{6} + A_{j-1} x_{j-1} + B_{j-1} - y_{j-1} \right].$$

Elimination of B_{j-1} produces

$$A_{j-1} = -\frac{1}{\lambda_j h_j} \left[\frac{M_{j+1}}{h_{j+1}} - \frac{M_j}{h_{j+1}} + \frac{M_{j-1}}{h_j} - \frac{M_j}{h_j} \right] + \frac{1}{\lambda_{j-1} h_j} \left[\frac{M_j}{h_j} - \frac{M_{j-1}}{h_j} + \frac{M_{j-2}}{h_{j-1}} - \frac{M_{j-1}}{h_{j-1}} \right]$$

$$- \frac{1}{h_j} \left[\frac{M_j h_j^2}{6} - \frac{M_{j-1} h_j^2}{6} - y_j + y_{j-1} \right]$$

and consequently

$$B_{j-1} = -\frac{1}{\lambda_j} \left[\frac{M_{j+1}}{h_{j+1}} - \frac{M_j}{h_{j+1}} + \frac{M_{j-1}}{h_j} - \frac{M_j}{h_j} \right] - \frac{M_j h_j^2}{6} - A_{j-1} x_j + y_j .$$

A recurrence relationship can now be derived through the continuity of the

first derivatives at each knot x_j ,

$$S_\Delta'(x_{j-}) = S_\Delta'(x_{j+})$$

which is expressed as

$$-\frac{M_{j+1}h_{j+1}}{2} + A_j = \frac{M_j h_j}{2} + A_{j-1} \cdot$$

The following recurrence relationship is obtained

$$M_{j-2}\left[-\frac{1}{\lambda_{j-1}h_j h_{j-1}}\right]$$

$$+M_{j-1}\left[\frac{1}{\lambda_j h_{j+1} h_j} + \frac{1}{\lambda_j h_j^2} + \frac{1}{\lambda_{j-1}h_j^2} + \frac{1}{\lambda_{j-1}h_j h_{j-1}} - \frac{h_j}{6}\right]$$

$$+ M_j\left[-\frac{h_{j+1}}{3} - \frac{1}{\lambda_{j+1}h_{j+1}^2} - \frac{1}{\lambda_j h_{j+1}^2} - \frac{1}{\lambda_j h_{j+1} h_j} - \frac{h_j}{3} - \frac{1}{\lambda_j h_j h_{j+1}}\right.$$

$$\left. - \frac{1}{\lambda_j h_j^2} - \frac{1}{\lambda_{j-1}h_j^2}\right]$$

$$+ M_{j+1}\left[\frac{1}{\lambda_{j+1}h_{j+1}h_{j+2}} + \frac{1}{\lambda_{j+1}h_{j+1}^2} + \frac{1}{\lambda_j h_{j+1}^2} - \frac{h_{j+1}}{6} + \frac{1}{\lambda_j h_j h_{j+1}}\right]$$

$$+ M_{j+2}\left[-\frac{1}{\lambda_{j+1}h_{j+1}h_{j+2}}\right]$$

$$= \frac{1}{h_{j+1}}\left[y_j - y_{j+1}\right] - \frac{1}{h_j}\left[y_{j-1} - y_j\right] \cdot$$

By applying this recurrence relationship to each of the knots x_j (j=2,3, . . . , n-2) in turn, n-4 linear equations may then be immediately specified in the n unknown moments M_j. By introducing two fictitious external knots x_{-1}, and x_{n+1} two additional equations may be added to the system when the recurrence relationship is applied at x_1 and x_{n-1}. The four remaining conditions may be specified somewhat arbitrarily as:

$$C_1 M_{-1} + C_2 M_o = C_3$$

$$C_4 M_o + C_5 M_1 = C_6$$

$$C_7 M_{n-1} + C_8 M_n = C_9$$

$$C_{10} M_n + C_{11} M_{n+1} = C_{12}$$

By the selection of appropriate coefficients C_i various end conditions may be applied. The FORTRAN program SPLINS allows three options to the user:

(i) The default setting of $M_{-1} = M_o = M_1$ and $M_{n-1} = M_n = M_{n+1}$ where the constant moment in the two segments at each end results in the approximating curve having quadratic run-off there. It is to be expected from (1) that such splines will pass through the end points. ...(4)

(ii) The user may specify the coefficients if he so desires. ...(5)

(iii) Often knowledge of the behaviour of the derivatives is available at the end points. Differentiation of (3) gives

$$S'_\Delta(x) = \frac{M_{j-1}(x_j-x)^2}{2h_j} + \frac{M_j(x-x_{j-1})^2}{2h_j} + A_{j-1} \quad .$$

Using the notation m_j to represent the first derivative at x_j we may express the derivatives at the end points of $[x_{j-1}, x_j]$ as

$$m_j = M_{j+1}\left[-\frac{1}{\lambda_j h_j h_{j+1}}\right]$$

$$+ M_j \left[\frac{1}{\lambda_{j-1}h_j^2} + \frac{1}{\lambda_j h_j h_{j+1}} + \frac{1}{\lambda_j h_j^2}\right]$$

$$+ M_{j-1}\left[\frac{h_j}{6} - \frac{1}{\lambda_{j-1}h_j^2} - \frac{1}{\lambda_{j-1}h_j h_{j-1}} - \frac{1}{\lambda_j h_j^2}\right]$$

$$+ M_{j-2}\left[\frac{1}{\lambda_j h_j h_{j-1}}\right] + \frac{y_j}{h_j} - \frac{y_{j-1}}{h_{j-1}}$$

and

$$m_{j-1} = M_{j+1}\left[-\frac{1}{\lambda_j h_j h_{j+1}}\right]$$

$$+ M_j \left[\frac{1}{\lambda_{j-1}h_j^2} + \frac{1}{\lambda_j h_j h_{j+1}} + \frac{1}{\lambda_j h_j^2} - \frac{h_j}{6}\right]$$

$$+ M_{j-1}\left[\frac{2h_j}{3} - \frac{1}{\lambda_{j-1}h_j^2} - \frac{1}{\lambda_{j-1}h_j h_{j-1}} - \frac{1}{\lambda_j h_j^2} + \frac{h_j}{6}\right]$$

$$+ M_{j-2}\left[\frac{1}{\lambda_{j-1}h_j h_{j-1}}\right] + \frac{y_j}{h_j} - \frac{y_{j-1}}{h_j} \quad .$$

Consequently the user may specify derivative conditions of the form

$$\begin{aligned} c_1 m_o + c_2 m_1 &= c_3 \\ c_4 m_{n-1} + c_5 m_n &= c_6 \quad , \end{aligned} \qquad \ldots(6)$$

which can be transformed by the subroutine into moment conditions. The two derivative conditions are not sufficient on their own and quadratic run-off is permitted in the external segments to form a fully determined system of equations by the automatic setting of $M_{-1} = M_o$ and $M_n = M_{n+1}$.

Application of the recurrence relationship produces a set of (n+2) linear equations with a diagonal band width of 5. Only the non zero elements need be kept and the solution can be obtained through forward elimination and back substitution. The main diagonal contains the dominating terms so the procedure causes errors to be damped out rapidly.

2.3 Calling Procedure

The subroutine SPLINS is first called to set up the interpolating spline function and the computed moments are returned in a vector. The user must preserve these along with the experimental data and weighting system to carry out interpolation through the use of an associated FORTRAN function EVSPLS.

Often it is desired to determine the first derivative of the approximating curve. Differentiation of (3) gives a convenient continuous interpolating scheme of second order available through the function EVDSPL.

CALL SPLINS (N,A,B,C,D,E,RHS,AM,ALAM,X,Y,IØPT,CØEF)

where N is the number of tabulated data points +2,

A,B,C,D,E,RHS are real vectors of length N used by SPLINS,
AM is a real vector of length N that contains the computed moments $M_{-1}, M_0, \ldots, M_{n+1}$ on return from SPLINS,
ALAM is a real vector of length N-2 containing the positive constants for each knot, chosen to obtain the appropriate balance between the least squares and curvature components of the functional (1).
(In the selection of these weights experience will be an advantage. The spline should not be forced to pass through data points by the selection of excessive weights, as other spline formulations are readily available when the data points are error free, and the interpolating function may take on a cubic appearance between adjacent points that are too highly

weighted.),

X,Y are real vectors of length N-2 containing the experimental data,

IØPT = 1 default end conditions (4) apply,

= 2 the end condition (5) applies and the user must specify the coefficients C_i (i = 1,2, . . . , 12) in the vector CØEF,

= 3 the derivative conditions (6) apply and the coefficients C_i (i = 1,2, . . . , 6) must be specified in the vector CØEF and

CØEF is a vector of length 6 or 12 and need only be used when IØPT = 2 or 3.

Interpolation is carried out by using the appropriate FORTRAN function viz.,

YPRED = EVSPLS (N,AM,ALAM,X,Y,XX) for function interpolation

YDERIV = EVDSPL (N,AM,ALAM,X,Y,XX) for derivative interpolation.

N,M,X,Y,ALAM are the corresponding arguments used in the original call to SPLINS.

XX is the independent variable for which interpolation is required.

3. SUBROUTINE SMOOTH

3.1 Description of the Interpolating Function

Given a set of n observed data values y_i, x_i (i = 1,2, . . . , n) and a set of m break points x'_i (i = 1,2, . . . , m) then (m+1) cubic polynomials are constructed that give a best least squares fit in each segment subject to the constraints of continuity placed on these polynomials and their derivatives at the joints. The user may specify that either the first or second derivatives are to be continuous.

3.2 Formulation of the Interpolating Function

For any set of k observations, the coefficients of the cubic polynomial that give the minimum least squares error with weighting are given by the solution to the linear system of equations represented in matrix notation as

$$X^T \underset{\sim}{w}^T X \underset{\sim}{a} = \underset{\sim}{w}^T \underset{\sim}{y} .$$

Here X is a kx4 matrix formed from the experimental data values of the independent variable, $\underset{\sim}{w}^T$ is a vector containing the k weights associated with each observation, $\underset{\sim}{y}$ contains the observed values, and $\underset{\sim}{a}$ will give the four coefficients of the cubic polynomial on solution of such a system. With (m+1) cubic polynomials and no restrictions placed at each of the m break points a least squares problem would be given as

$$\begin{bmatrix} X_1^T \underset{\sim}{w}_1^T X_1 & & & \\ & X_2^T \underset{\sim}{w}_2^T X & & \\ & & & \\ & & & X_{m+1}^T \underset{\sim}{w}_{m+1}^T X \end{bmatrix} \begin{bmatrix} \uparrow \\ \underset{\sim}{a}_1 \\ \downarrow \\ \uparrow \\ \underset{\sim}{a}_2 \\ \downarrow \\ \\ \uparrow \\ \underset{\sim}{a}_{m+1} \\ \downarrow \end{bmatrix} = \begin{bmatrix} \underset{\sim}{w}_1^{T\uparrow} y_1 \\ \underset{\sim}{w}_{1\downarrow} \\ \underset{\sim}{w}_2^{T\uparrow} y_2 \\ \underset{\sim}{w}_{2\downarrow} \\ \\ \underset{\sim}{w}_{m+1}^{T\uparrow} y_{m+1} \\ \downarrow \end{bmatrix}$$

which in future we shall write as

$$X^T \underset{\sim}{w}^T X \underset{\sim}{a} = \underset{\sim}{w}^T \underset{\sim}{y} \quad .$$

In the case of continuity constraints being placed on the first two derivatives and the functions at the j^{th} break point, the three relations imposed on the coefficients are

$$a_{4j-3} + a_{4j-2}\, x_j + a_{4j-1}\, x_j^2 + a_{4j}\, x_j^3 = a_{4j+1} + a_{4j+2}\, x + a_{4j+3}\, x^2 + a_{4j+4}\, x^3$$

$$a_{4j-2} + 2a_{4j-1}\, x_j + 3a_{4j} x_j^2 = a_{4j+2} + 2a_{4j+3}\, x + 3a_{4j+4}\, x^2$$

$$2a_{4j-1} + 6a_{4j}\, x_j = 2a_{4j+3} + 6a_{4j+4} \,. \qquad \ldots(7)$$

In matrix notation we may write these as

$$P_j \begin{bmatrix} \underset{\sim}{a}_{j-1} \\ \\ \underset{\sim}{a}_j \end{bmatrix} = 0$$

where P_j is a 3 x 8 matrix formed from (7) and $\underset{\sim}{a}_j$ denotes the four coefficients of the polynomial in the region $x_{j+1} > x > x_j$. The full set of constraints can be expressed as

$$\begin{bmatrix} P_1 & & & \\ & P_2 & & \\ & & \ddots & \\ & & & P_m \end{bmatrix} \begin{bmatrix} \underset{\sim}{a}_1 \\ \underset{\sim}{a}_2 \\ \\ \underset{\sim}{a}_{m-1} \\ \underset{\sim}{a}_m \end{bmatrix} = \begin{bmatrix} \uparrow \\ 0 \\ \downarrow \end{bmatrix} ,$$

which for convenience we shall now denote as

$$P \underset{\sim}{a} = \underset{\sim}{p} \quad .$$

When the first derivative only is constrained to be continuous at each break point the third equation of (7) is neglected in forming the reduced P matrix. The approximation problem consists of obtaining a weighted least squares solution to the overdetermined system

$$X \underset{\sim}{a} = \underset{\sim}{y}$$

that satisfies the side conditions

$$P \underset{\sim}{a} = \underset{\sim}{p} \quad .$$

By the introduction of a vector $\underset{\sim}{\ell}$ containing 3m Lagrange multipliers the required solution is found by solving

$$\left[\begin{array}{c|c} X^T \underset{\sim}{w}^T X & P^T \\ \hline P^T & 0 \end{array}\right] \left[\begin{array}{c} \underset{\sim}{a} \\ \hline \underset{\sim}{\ell} \end{array}\right] = \left[\begin{array}{c} \underset{\sim}{w}^T \underset{\sim}{y} \\ \hline p \end{array}\right] \quad . \qquad \ldots(8)$$

This system involves 4(m+1) + 3m equations with a large number of zero elements structured in such a way to make the following scheme more attractive than direct inversion or solution of (8). By considering the separate block components of (8) we may write

$$X^T \underset{\sim}{w}^T X \underset{\sim}{a} + P^T \underset{\sim}{\ell} = \underset{\sim}{w}^T \underset{\sim}{y} \qquad \ldots(9)$$

and

$$P \underset{\sim}{a} = \underset{\sim}{p} \qquad \ldots(10)$$

From (9) we obtain

$$\underset{\sim}{a} = (X^T \underset{\sim}{w}^T X)^{-1} \quad (\underset{\sim}{w}^T \underset{\sim}{y} - P^T \underset{\sim}{\ell})$$

which when substituted in (10) gives the following expression for $\underset{\sim}{\ell}$, the vector of Lagrange multipliers,

$$\underset{\sim}{\ell} = \left(P(X^T \underset{\sim}{w}^T X)^{-1} P^T \right)^{-1} \left(P(X^T \underset{\sim}{w} X)^{-1} \underset{\sim}{w}^T \underset{\sim}{y} - \underset{\sim}{p} \right) .$$

The solution consequently commences with the inversion of the 4(m+1) x 4(m+1) system $X^T \underset{\sim}{w}^T X$ where the diagonal block structure is exploited by inversion of m+1 separate 4x4 matrices. In the calculation of the vector of Lagrange multipliers, a 3m system of equations is involved and for $m \geqslant 2$ further attention may be given to the block tridiagonal structure of $P(X^T \underset{\sim}{w}^T X) \; P^T$.

Despite the additional operations involved in the multiplication of matrices in this approach there is an overall saving due to the reduction in

the size of matrices to be inverted. Experience gained has shown that single precision arithmetic on the IBM360 does not give sufficient accuracy in the modified method of solution outlined here. Tests have also shown that while direct solution of (7) in single precision arithmetic is improved, significant differences still exist between it and a double precision solution. The modified scheme does not involve any penalty in storage as all the matrix operations can be carried out as is now shown in the space assigned to store the original matrix defined in (7).

Step 1

$$\left[\begin{array}{c|c|c} \underset{\sim}{X}^T\underset{\sim}{w}\underset{\sim}{X} & P^T & \underset{\sim}{w}^T\underset{\sim}{y} \\ \hline P & 0 & \underset{\sim}{p} \end{array}\right]$$

Step 2

$$\left[\begin{array}{c|c|c} (\underset{\sim}{X}^T\underset{\sim}{w}^T\underset{\sim}{X})^{-1} & P^T & \underset{\sim}{w}^T\underset{\sim}{y} \\ \hline P & 0 & \underset{\sim}{p} \end{array}\right]$$

Step 3

$$\left[\begin{array}{c|c|c} (\underset{\sim}{X}^T\underset{\sim}{w}^T\underset{\sim}{X})^{-1} & P^T & \underset{\sim}{w}^T\underset{\sim}{y} \\ \hline P(\underset{\sim}{X}^T\underset{\sim}{w}^T\underset{\sim}{X})^{-1} & 0 & \underset{\sim}{p} \end{array}\right]$$

Step 4

$$\left[\begin{array}{c|c|c} (\underset{\sim}{X}^T\underset{\sim}{w}^T\underset{\sim}{X})^{-1} & P^T & \underset{\sim}{w}^T\underset{\sim}{y} \\ \hline P(\underset{\sim}{X}^T\underset{\sim}{w}^T\underset{\sim}{X})^{-1} & P(\underset{\sim}{X}^T\underset{\sim}{w}^T\underset{\sim}{X})^{-1}P^T & P(\underset{\sim}{X}^T\underset{\sim}{w}^T\underset{\sim}{X})^{-1}\underset{\sim}{w}^T\underset{\sim}{y}-\underset{\sim}{p} \end{array}\right]$$

Step 5

$$\left[\begin{array}{c|c|c} (\underset{\sim}{X}^T\underset{\sim}{w}^T\underset{\sim}{X})^{-1} & P^T & \underset{\sim}{w}^T\underset{\sim}{y} \\ \hline P(\underset{\sim}{X}^T\underset{\sim}{w}^T\underset{\sim}{X})^{-1} & (P(\underset{\sim}{X}^T\underset{\sim}{w}^T\underset{\sim}{X})P^T)^{-1} & P(\underset{\sim}{X}^T\underset{\sim}{w}^T\underset{\sim}{X})^{-1}\underset{\sim}{w}^T\underset{\sim}{y}-\underset{\sim}{p} \end{array}\right]$$

Step 6

$$\left[\begin{array}{c|c|c} (\underset{\sim}{X}^T\underset{\sim}{w}^T\underset{\sim}{X})^{-1} & P^T & \underset{\sim}{w}^T\underset{\sim}{y} \\ \hline P(\underset{\sim}{X}^T\underset{\sim}{w}^T\underset{\sim}{X})^{-1} & (P(\underset{\sim}{X}^T\underset{\sim}{w}^T\underset{\sim}{X})P^T)^{-1} & \underset{\sim}{\ell} \end{array}\right]$$

Step 7

$$\left[\begin{array}{c|c|c} (\underset{\sim}{X}^T\underset{\sim}{w}^T\underset{\sim}{X})^{-1} & P^T & \underset{\sim}{w}^T\underset{\sim}{y} - P^T\underset{\sim}{\ell} \\ \hline P(\underset{\sim}{X}^T\underset{\sim}{w}^T\underset{\sim}{X})^{-1} & (P(\underset{\sim}{X}^T\underset{\sim}{w}^T\underset{\sim}{X})^{-1}P^T)^{-1} & \underset{\sim}{\ell} \end{array}\right]$$

Step 8

$$\left[\begin{array}{c|c|c} (\underset{\sim}{X}^T\underset{\sim}{w}^T\underset{\sim}{X})^{-1} & P^T & \underset{\sim}{a} \\ \hline P(\underset{\sim}{X}^T\underset{\sim}{w}^T\underset{\sim}{X})^{-1} & (P(\underset{\sim}{X}^T\underset{\sim}{w}^T\underset{\sim}{X})^{-1}P^T)^{-1} & \underset{\sim}{\ell} \end{array}\right]$$

3.3 Calling Procedure

CALL SMOOTH (X,Y,N,XSEG,CØEF,KNOTS,Nl,NlPl,W,WT,IØPT)

where X,Y are real vectors of length N containing the experimental data,

XSEG is a vector containing the break points. When the break points do not coincide with values of the independent variable for which the experimental observations were recorded, then extrapolation is used in each segment to derive the constraints that force piecewise continuity into the smoothed approximating function. When break points are required that do coincide care must be taken in the writing of the FORTRAN code to achieve this,

e.g. Code XSEG(K) = X(L),

otherwise I/O conversion limitations may cause the observed values intended to be breakpoints to only be included in one of the two adjoining segments, with extrapolation being employed to obtain the constraints),

CØEF is a real vector of length 4 x (KNOTS+1) that contains the coefficients on return from SMOOTH,

KNOTS specifies the number of break points,

Nl must be set $\geq$ 7xKNOTS+4,

NlPl must be set $\geq$ 7xKNOTS+5,

W is a double precision matrix dimensioned to be NlxNlPl by the user and is used by SMOOTH to perform all matrix operations,

$\underset{\sim}{w}^T$ is a real vector of length n to which weights corresponding to each observation are assigned, and

IØPT is set as 1 or 2, and is used to indicate the order of the derivatives constrained to be continuous at each segment joint.

The interpolating polynomial may be evaluated through the use of a FORTRAN function EVSMTH,

YPRED = EVSMTH (CØEF,XSEG,KNØTS,XX)

where XX is the value for which interpolation is desired.

4. EXAMPLES OF THE USE OF THE SUBROUTINE SPLINS AND SMOOTH

In Figure 1 approximating curves are shown for data generated for the function $y = \frac{1}{1 + x^2}$ with 20% random noise. Subroutine SPLINS is used with a uniform weighting scheme of $\lambda = 8$ (Plot A). The close degree to which the approximating spline follows the data is shown except for the peak observations. In (1) the component of curvature dominates the least squares component for smaller values of λ in the neighbourhood of the peak. Any attempt to 'lift' the peak by a uniform increase in the weight λ will cause the approximating

curve to wander about in a cubic fashion in each segment. Increasing to 32 the weight associated with the peak point alone ensures that the required lift (Plot B) is obtained and smoothness preserved.

In the subroutine SMOOTH considerable skill is necessary to position the break points to produce a satisfactory result as Figure 2 demonstrates (IØPT = 1). The break point at x = -1.5 gives an acceptable form to the left of the peak, while the second break point (shown at x = 0.25) would have produced a monotonic decreasing function on the right of the peak if it were moved farther to the right. When the second derivative was constrained to be continuous as well, the peak proved easy to handle but great difficulty was experienced with cubic fluctuations for $|x| > 3$.

In Figure 3 two break points at x = 1.3, and 2.3 were chosen with continuity being required in both derivatives (Plot B) and the first derivative (Plot A). The additional constraint while having desirable properties of smoothness in an approximating function makes such a function stiffer.

5. REFERENCES

Ahlberg, J.H., Nilson, E.N., Walsh, J.L., (1967) - The Theory of Splines and Their Applications, Academic Press, N.Y.

Handscomb, D.C., (1965) - Methods of Numerical Approximation, (Ed. Handscomb, D.C.) Pergamon Press.

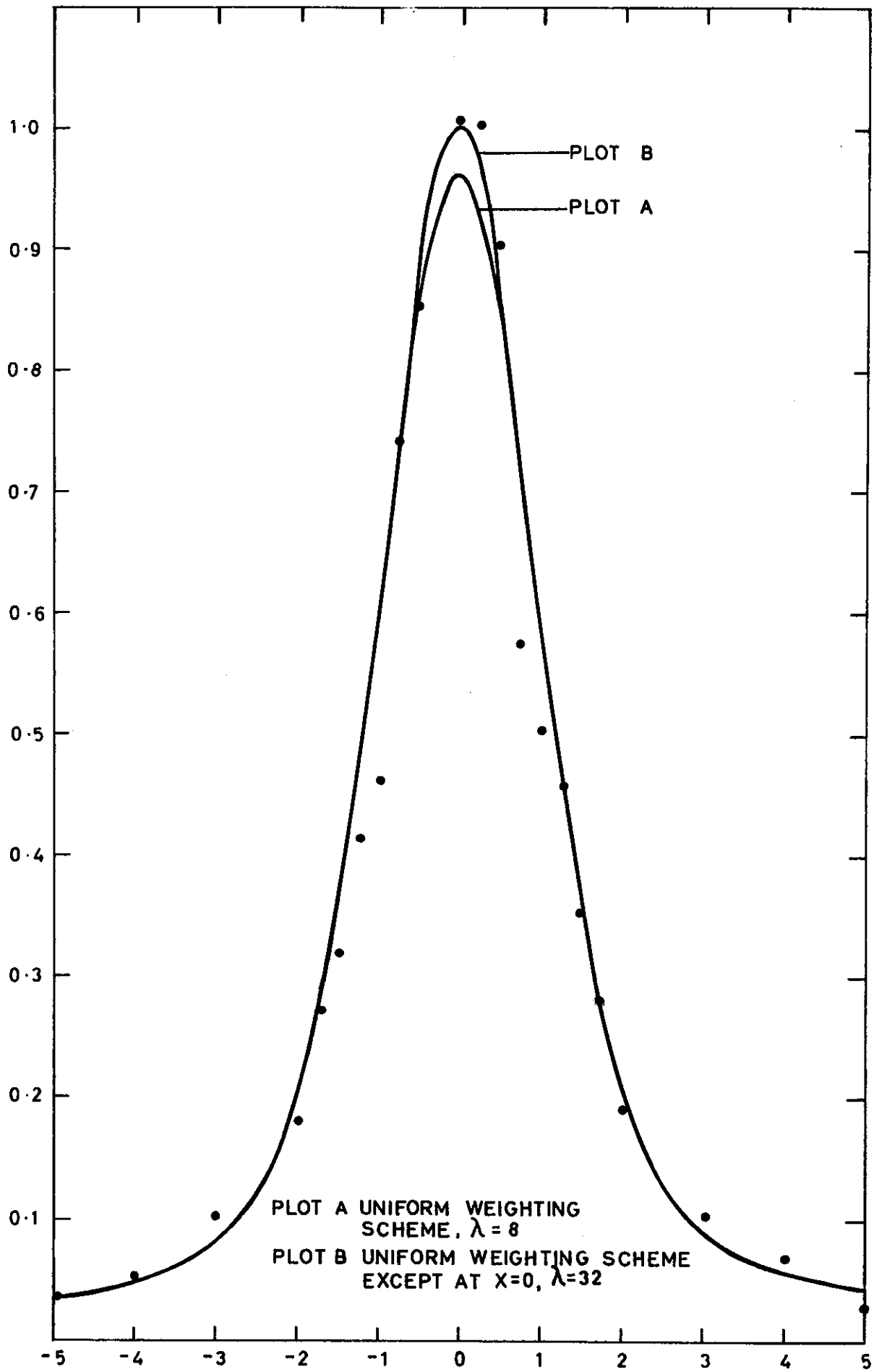

FIGURE 1. SAMPLE OUTPUT FROM SUBROUTINE SPLINS

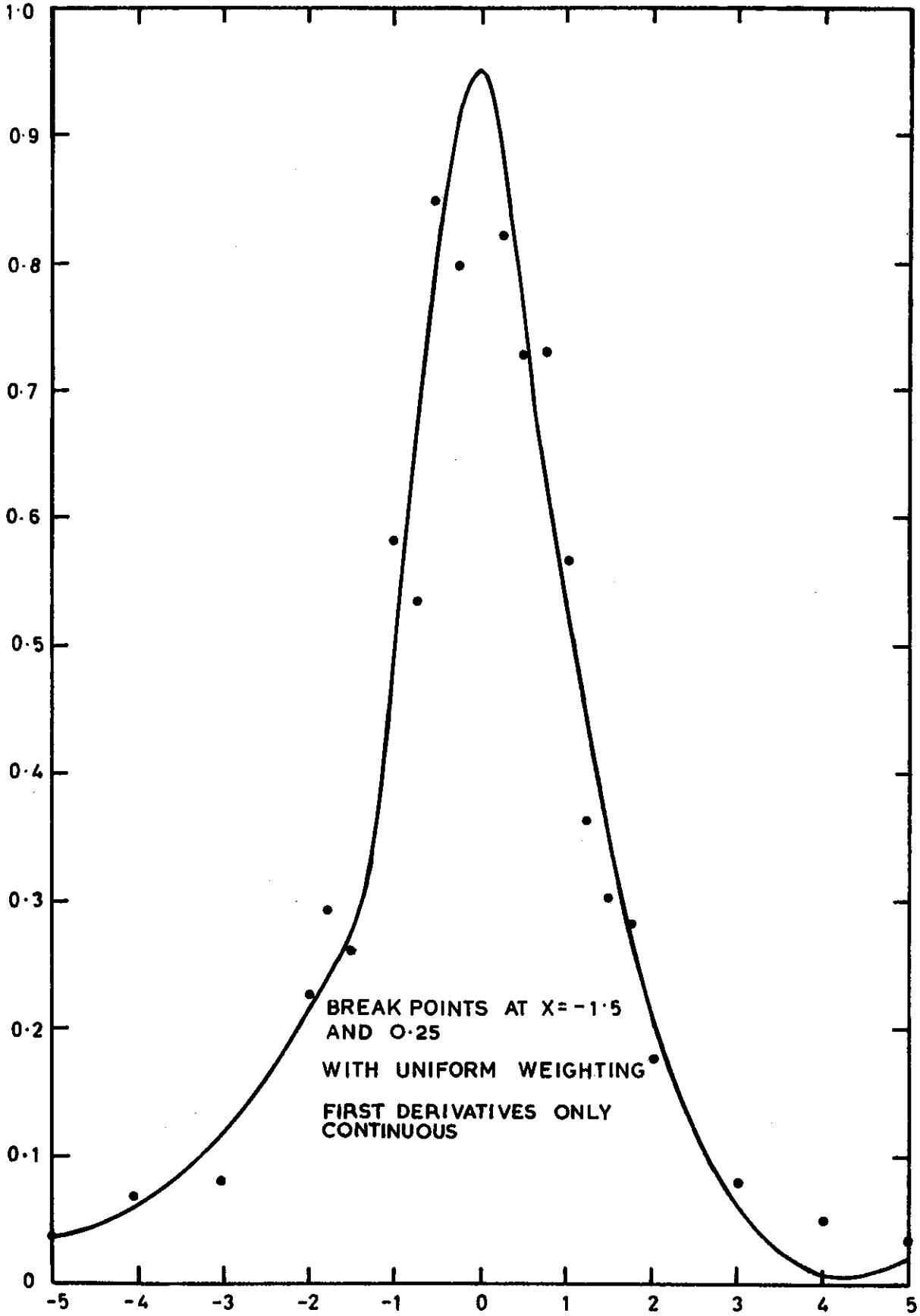

FIGURE 2. SAMPLE OUTPUT FROM SUBROUTINE SMOOTH

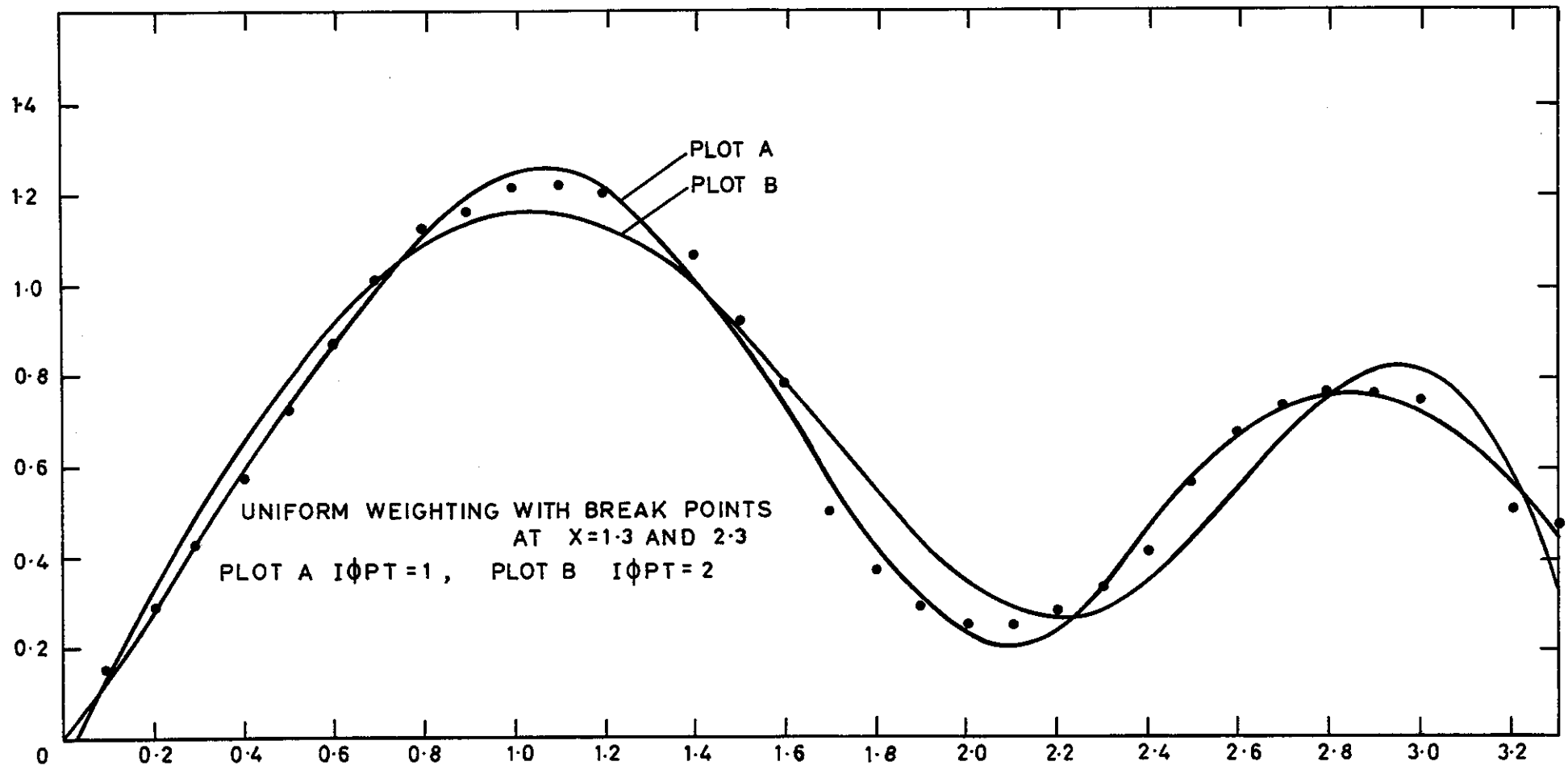

FIGURE 3. SAMPLE OUTPUT FROM SUBROUTINE SMOOTH