OT REPORT 75-70

A METHOD OF BIVARIATE INTERPOLATION AND SMOOTH SURFACE FITTING FOR VALUES GIVEN AT IRREGULARLY DISTRIBUTED POINTS

HIROSHI AKIMA

U.S. DEPARTMENT OF COMMERCE
Rogers C. B. Morton, Secretary

Betsy Ancker-Johnson, Ph. D.
Assistant Secretary for Science and Technology

OFFICE OF TELECOMMUNICATIONS
John M. Richardson, Acting Director

August 1975

For sale by the Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402

UNITED STATES DEPARTMENT OF COMMERCE
OFFICE OF TELECOMMUNICATIONS

STATEMENT OF MISSION

The mission of the Office of Telecommunications in the Department of Commerce is to assist the Department in fostering, serving, and promoting the nation's economic development and technological advancement by improving man's comprehension of telecommunication science and by assuring effective use and growth of the nation's telecommunication resources.

In carrying out this mission, the Office

- Conducts research needed in the evaluation and development of policy as required by the Department of Commerce
- Assists other government agencies in the use of telecommunications
- Conducts research, engineering, and analysis in the general field of telecommunication science to meet government needs
- Acquires, analyzes, synthesizes, and disseminates information for the efficient use of the nation's telecommunication resources.
- Performs analysis, engineering, and related administrative functions responsive to the needs of the Director of the Office of Telecommunications Policy, Executive Office of the President, in the performance of his responsibilities for the management of the radio spectrum
- Conducts research needed in the evaluation and development of telecommunication policy as required by the Office of Telecommunications Policy, pursuant to Executive Order 11556

USCOMM - ERL

TABLE OF CONTENTS

A METHOD OF BIVARIATE INTERPOLATION AND SMOOTH SURFACE FITTING FOR VALUES GIVEN AT IRREGULARLY DISTRIBUTED POINTS

Hiroshi Akima *

Abstract — A method of bivariate interpolation and smooth surface fitting is developed for z values given at points irregularly distributed in the x-y plane. The interpolating function is a fifth-degree polynomial in x and y defined in each triangular cell which has projections of three data points in the x-y plane as its vertexes. Each polynomial is determined by the given values of z and estimated values of partial derivatives at the vertexes of the triangle. Procedures for dividing the x-y plane into a number of triangles, for estimating partial derivatives at each data point, and for determining the polynomial in each triangle are described. A simple example of the application of the proposed method is shown. User information and Fortran listings are given on a computer subprogram package that implements the proposed method.

Key Words and Phrases — Bivariate interpolation, interpolation, partial derivative, polynomial, smooth surface fitting.

---

* The author is with the Institute for Telecommunication Sciences, Office of Telecommunications, U.S. Department of Commerce, Boulder, Colorado 80302.

1. INTRODUCTION

In a previous study (Akima, 1974a, b), we developed a method of bivariate interpolation and smooth surface fitting. The method was designed in such a way that the resulting surface would pass through all the given data points. Adopting local procedures, it successfully suppressed undulations in the resulting surface which are very likely to appear in surfaces fitted by other methods. Like many other methods, however, this method also has a serious drawback. Applicability is restricted to cases where the values of the function are given at rectangular grid points in a plane; i.e., the values of $z = z(x,y)$ must be given as $z_{ij} = z(x_i, y_j)$ in the x-y plane, where $i = 1, 2, \ldots, n_x$ and $j = 1, 2, \ldots, n_y$. This restriction prevents application to cases where collection of data at rectangular grid points is impossible or otherwise impractical.

The subject of the present study is bivariate interpolation and smooth surface fitting in the general case where the values of the function are given at irregularly distributed points in a plane; i.e., the case where the z values are given as $z_i = z(x_i, y_i)$, where $i = 1, 2, \ldots, n$. Despite potentially wide applicability of a method of bivariate interpolation and smooth surface fitting for irregularly distributed points, studies for developing such a method have not been active in the past.

Two types of approaches are possible; one using a single global function, and the other based on a collection of local functions. In the former approach, the procedure often becomes too complicated to manage as the number of given data points increases. Moreover, the resulting surface from the former sometimes exhibits excessive undulations. For these reasons, only the latter approach is considered in the present study.

Bengtsson and Nordbeck (1964) suggested a method based on partitioning the x-y plane into a number of triangles (each triangle having projections of three data points in the x-y plane as its vertexes) and on fitting a plane to the surface in each triangle. Obviously, the resulting surface is not smooth on the sides of the triangles although it is continuous. In addition, their suggestion for partitioning so that the sum of the lengths of the sides of these triangles be minimized is too complicated to implement.

Shepard (1968) suggested a method based on weighted averages of the given z values. The basic weighting function is the square of the reciprocal of the distance between the projection of each data point and that of the point at which interpolation is to be performed. The actual weighting function is an improvement of this basic weighting function in that the actual function corresponding to a distant data point vanishes. Through this improvement the originally global procedures in this method became local. This method has several desirable properties. It takes into account the "shadowing" of the influence of a data point by a nearer one in the same direction. It yields reasonable slopes at the given data points. However, it fails to produce a plane when all the given data points lie in a slanted plane; this property is considered to be a serious drawback.

In conjunction with variational problems containing second-order derivatives, Zlamal (1968) discussed an approximation procedure using fifth-degree polynomials in x and y over triangular regions in the x-y plane. To determine the coefficients of the polynomial for each triangle, he uses, in addition to the z values and the first and second partial derivatives (i.e., z_x, z_y, z_{xx}, z_{xy}, and z_{yy}) at the three vertexes of the triangle, three partial derivatives, each differentiated in the direction normal to one of the three sides of the triangle at the

midpoint of the side in question. The theory was generalized to (4m + 1)st-degree polynomials for functions m-times continuously differentiable on a closed triangular domain by Zenisek (1970). Although a comprehensive interpolation method is not suggested in their papers, their papers were instrumental in stimulating portions of the ideas developed here.

In the present study, we develop and propose a method of bivariate interpolation and smooth surface fitting that is applicable to z values given at irregularly distributed points in the x-y plane. As in the method for rectangular grid points developed in the previous study (Akima, 1974 a, b), the interpolating function used in the method proposed in the present study is also a smooth function; i.e., the interpolating function and its first-order partial derivatives are continuous. The proposed method is also based on local procedures. The surface resulting from the proposed method will pass through all the given data points.

In this report, the proposed method is outlined in section 2, with some mathematical details in Appendix A. A simple example that illustrates the application of the proposed method is shown in section 3. Some pertinent remarks are addressed in section 4. In Appendix B, user information and Fortran listings are given on the IDBVIP/IDSFFT subprogram package that implements the proposed method.

2. DESCRIPTION OF THE METHOD

In this method the x-y plane is divided into a number of triangular cells; each having projections of three data points in the plane as its vertexes, and a bivariate fifth-degree polynomial in x and y is applied to each triangular cell.

For a unique partitioning of the plane, the x-y plane is divided into triangles by the following steps. First, determine the nearest pair of data points and draw a line segment between the points. Next, find the nearest pair of data points among the remaining pairs and draw a line segment between these points if the line segment to be drawn does not cross any other line segment already drawn. Repeat the second step until all possible pairs are exhausted.

The z value in a triangle is interpolated with a bivariate fifth-degree polynomial in x and y, i.e.,

$$z(x,y) = \sum_{j=0}^{5} \sum_{k=0}^{5-j} q_{jk} x^j y^k . \qquad (1)$$

The coefficients of the polynomial are determined by the given z values at the three vertexes of the triangle and the estimated values of partial derivatives z_x, z_y, z_{xx}, z_{xy}, and z_{yy} at the vertexes, together with the imposed condition that the partial derivative of z by the variable measured in the direction perpendicular to each side of the triangle be a polynomial of degree three, at most, in the variable measured along the side. The procedure for interpolation in a triangle including determination of the coefficients of the polynomial is described in detail in Appendix A. Smoothness of the interpolated values and therefore smoothness of the resulting surface along each side of the triangle is proved also in the Appendix.

Procedures for estimating the five partial derivatives locally at each data point are not unique. The derivatives could be determined as partial derivatives of a second-degree polynomial in x and y that coincides with the given z values at six data points consisting of five data points the projections of which are nearest to the projection of the data point in question and the data point itself. This procedure is a bivariate extension of the one used in the univariate osculatory interpolation (Ackland, 1915). Adoption of this procedure has an advantage that, when z is a second-degree polynomial in x and y, the method yields exact results. As will be shown in section 3, however, this procedure sometimes yields very unreasonable results.

We will take a different approach and estimate the partial derivatives in two steps; i.e., the first-order derivatives in the first step and the second-order derivatives in the second step. To estimate the first-order partial derivatives at data point P_0 we use several additional data points P_i ($i = 1, 2, \ldots, n_n$) the projections of which are nearest to the projection of P_0 selected from all data points other than P_0. We take two data points P_i and P_j out of the n_n points and construct the vector product of $\overline{P_0P_i}$ and $\overline{P_0P_j}$; i.e., a vector that is perpendicular to both $\overline{P_0P_i}$ and $\overline{P_0P_j}$ with the right-hand rule and has a magnitude equal to the area of the parallelogram formed by $\overline{P_0P_i}$ and $\overline{P_0P_j}$. We take P_i and P_j in such a way that the resulting vector product always points upward (i.e., the z component of the vector product is always positive). We construct vector products for all possible combinations of $\overline{P_0P_i}$ and $\overline{P_0P_j}$ ($i \neq j$) and take a vector sum of all the vector products thus constructed. Then, we assume that the first-order partial derivatives z_x and z_y at P_0 are estimated as those of a plane that is normal to the resultant vector sum thus composed. Note that, when $n_n = 2$, the estimated z_x and z_y are equal to the partial

derivatives of a plane that passes through P_0, P_1, and P_2. Also note that, when $n_n = 3$ and the projection of P_0 in the x-y plane lies inside the triangle formed by the projections of P_1, P_2, and P_3, the estimated z_x and z_y are equal to the partial derivatives of a plane that passes through P_1, P_2, and P_3.

In the second step, we apply the procedure of "partial differentiation" described in the preceding paragraph to the estimated z_x values at P_i ($i = 0, 1, 2, \ldots, n_n$) and obtain estimates of $z_{xx} = (z_x)_x$ and $z_{xy} = (z_x)_y$ at P_0. We repeat the same procedure for the estimated z_y values and obtain estimates of $z_{xy} = (z_y)_x$ and $z_{yy} = (z_y)_y$. We adopt a simple arithmetic mean of two z_{xy} values thus estimated as our estimate for z_{xy} at P_0.

The selection of n_n is again not unique. Obviously, n_n cannot be less than 2. Also, it must be less than the total number of data points. Other than those, there seems to exist no theory that dictates a definite value for n_n. The best we can say is that, based on the example to be shown in section 3 and on some others, we recommend a number between 3 and 5 (inclusive) for n_n.

3. APPLICATIONS

Using a simple example taken from the previous study (Akima, 1974 a, b), we illustrate the application of the proposed method. We take a quarter of the surface shown in the example in the previous study and sample 50 data points from the surface randomly. The coordinate values of the sampled data points are shown in table 1. Knowing from the physical nature of the phenomenon that $z(x, y)$ is a single-valued smooth function of x and y, we try to interpolate the z values and to fit a smooth surface to the given data points.

Figure 1 depicts contour maps of the surfaces resulting from the 30 data points with asterisks in table 1, while figure 2, from all the 50 data points in the table. In these contour maps, projections of the data points are marked with encircled points. In each figure, the original surface from which the data points were sampled is shown in (a). The surface fitted with piecewise planes (i.e., the surface consisting of a number of pieces of planes, each applicable to one triangle) is shown in (b). Of course, such a surface is continuous but not smooth. The surface fitted by the method that estimates the partial derivatives with a second-degree polynomial is shown in (c). The surfaces fitted by the proposed method using three, four, and five additional data points for estimation of partial derivatives at each data point are shown in (d), (e), and (f), respectively. In drawing these contour maps, the z values were interpolated by their respective methods at the nodes of a grid consisting of 100 by 80 squares; in each square, the z values were interpolated linearly.

Figures 1 and 2 indicate that the proposed method yields reasonable results although these results might not necessarily be satisfactory for some applications. In these figures very little difference is

Table 1. An example set of data points.

(Thirty points with asterisks are used in figure 1, while all 50 points are used in figure 2.)

i		x_i	y_i	z_i	i		x_i	y_i	z_i
1	*	11.16	1.24	22.15	26		3.22	16.78	39.93
2	*	24.20	16.23	2.83	27	*	0.00	0.00	58.20
3		12.85	3.06	22.11	28	*	9.66	20.00	4.73
4	*	19.85	10.72	7.97	29		2.56	3.02	50.55
5	*	10.35	4.11	22.33	30	*	5.22	14.66	40.36
6		24.67	2.40	10.25	31	*	11.77	10.47	13.62
7	*	19.72	1.39	16.83	32		17.25	19.57	6.43
8		15.91	7.74	15.30	33	*	15.10	17.19	12.57
9	*	0.00	20.00	34.60	34	*	25.00	3.87	8.74
10	*	20.87	20.00	5.74	35		12.13	10.79	13.71
11		6.71	6.26	30.97	36	*	25.00	0.00	12.00
12		3.45	12.78	41.24	37		22.33	6.21	10.25
13	*	19.99	4.62	14.72	38		11.52	8.53	15.74
14		14.26	17.87	10.74	39	*	14.59	8.71	14.81
15	*	10.28	15.16	21.59	40	*	15.20	0.00	21.60
16	*	4.51	20.00	15.61	41		7.54	10.69	19.31
17		17.43	3.46	18.60	42	*	5.23	10.72	26.50
18		22.80	12.39	5.47	43		17.32	13.78	12.11
19	*	0.00	4.48	61.77	44	*	2.14	15.03	53.10
20		7.58	1.98	29.87	45	*	0.51	8.37	49.43
21	*	16.70	19.65	6.31	46		22.69	19.63	3.25
22	*	6.08	4.58	35.74	47	*	25.00	20.00	0.60
23		1.99	5.60	51.81	48		5.47	17.13	28.63
24	*	25.00	11.87	4.40	49	*	21.67	14.36	5.52
25	*	14.90	3.12	21.70	50	*	3.31	0.13	44.08

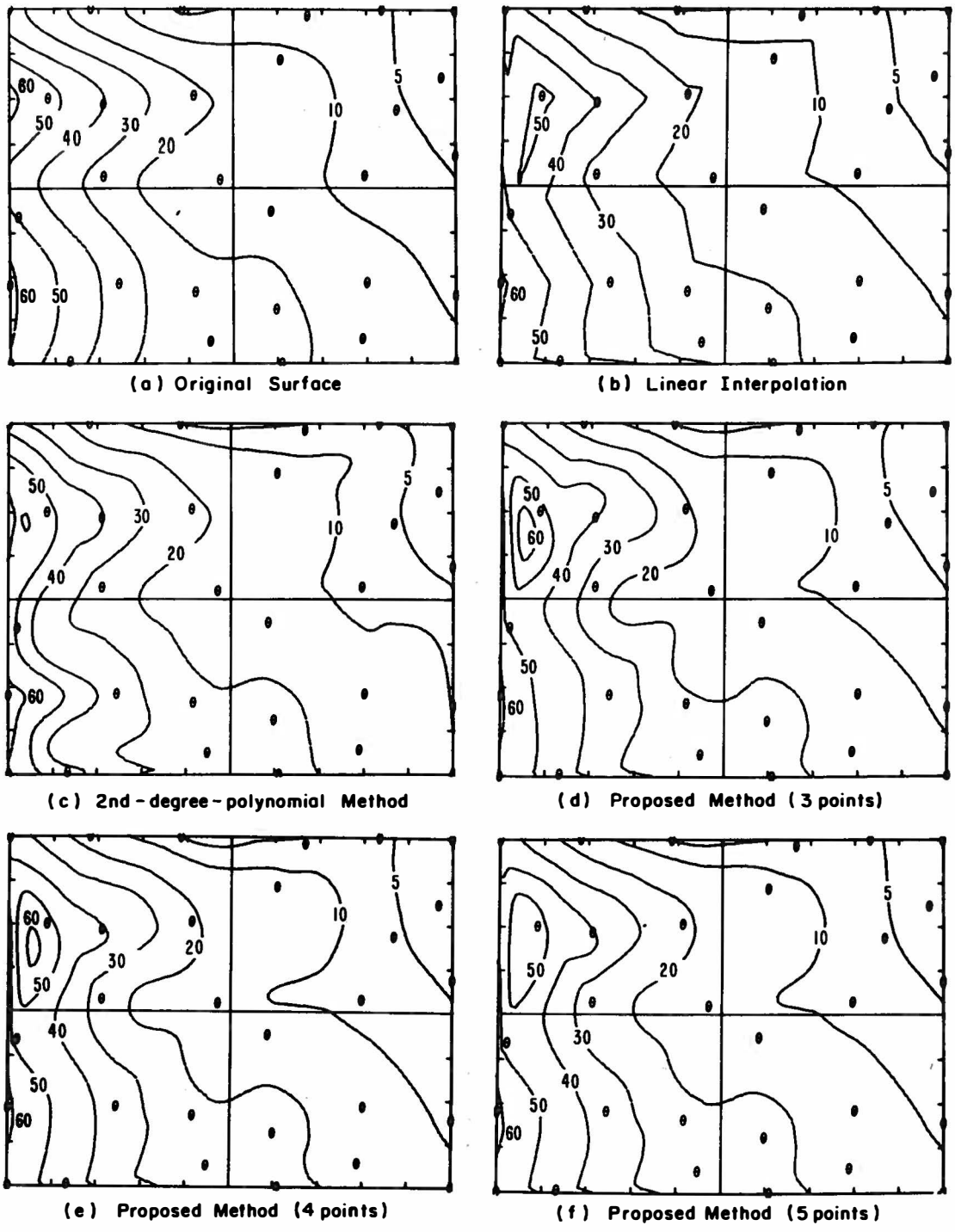

Figure 1. Contour maps for the surfaces fitted to 30 data points given with asterisks in table 1.

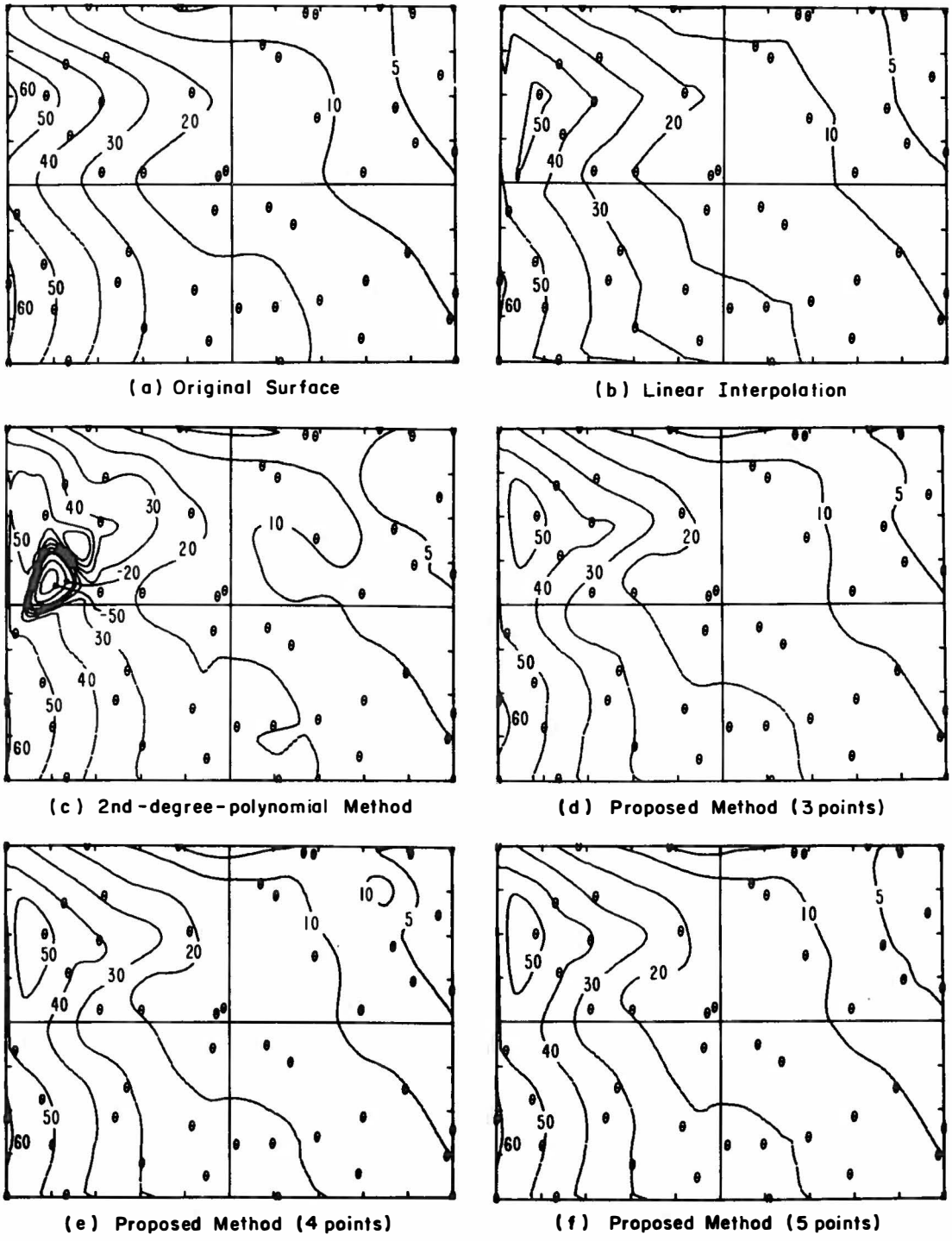

Figure 2. Contour maps for the surfaces fitted to 50 data points given in table 1.

exhibited in the resulting surfaces due to the difference in the number of data points used for the estimation of partial derivatives in the proposed method. Figures 1(c) and 2(c) demonstrate a peculiar idiosyncracy of the method based on second-degree polynomials; more data points yield a much worse result in this example.

Decision as to whether or not the proposed method is applicable to a particular problem rests on each prospective user of the method. The examples given here are expected to aid one in making such a decision. Comparison of (d), (e), or (f) fitted by the proposed method with (a) the original surface or (b) the piecewise-plane surface in each figure should be helpful for such a decision. Also, comparison of figures 1 and 2 gives one some idea on the dependence of the resulting surfaces upon the total number of data points and the complexity of original surfaces.

4. CONCLUDING REMARKS

We have described a method of bivariate interpolation and smooth surface fitting that is applicable when z values are given at points irregularly distributed in an x-y plane. For proper application of the method, the following remarks seem pertinent:

(i) The method does not smooth the data. In other words, the resulting surface passes through all the given points if the method is applied to smooth surface fitting. Therefore, the method is applicable only when the precise z values are given or when the errors are negligible.

(ii) As is true for any method of interpolation, the accuracy of interpolation cannot be guaranteed, unless the method in question has been checked in advance against precise values or a functional form.

(iii) The result of the method is invariant under a rotation of the x-y coordinate system.

(iv) The method is linear. In other words, if $z(x_i, y_i) = a\,z'(x_i, y_i) + b\,z''(x_i, y_i)$ for all i, the interpolated values satisfy $z(x, y) = a\,z'(x, y) + b\,z''(x, y)$, where a and b are arbitrary real constants.

(v) The method gives exact results when z(x,y) represents a plane; i.e., $z(x, y) = a_{00} + a_{10}x + a_{01}y$, where a_{00}, a_{10}, and a_{01} are arbitrary real constants.

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

A computer subprogram package that implements the proposed method is described in Appendix B.

5. ACKNOWLEDGMENTS

The author is grateful to James C. Ferguson of Teledyne-Ryan Aeronautical, San Diego, California, for his information concerning the past works in the subject field. He is also grateful to Rayner K. Rosich and Vincent D. Wayland of the Office of Telecommunications, Boulder, Colorado, for their helpful discussions and critical reviews of this report.

APPENDIX A

INTERPOLATION IN A TRIANGLE

Assuming that the plane is divided into a number of triangles, we describe a procedure for interpolating values of a function in each triangle. The primary emphasis is on the smoothness of the interpolated values not only inside of the triangle but also on the side of it; i.e., the interpolated values in a triangle must smoothly connect with those values in an adjacent triangle on the common side of two triangles.

Basic Assumptions.

Using a two-dimensional Cartesian coordinate system with x and y axes, we describe the basic assumptions as follows:

(i) The value of the function at point (x, y) in a triangle is interpolated by a bivariate fifth-degree polynomial in x and y; i.e.,

$$z(x, y) = \sum_{j=0}^{5} \sum_{k=0}^{5-j} q_{jk} x^j y^k . \qquad (A-1)$$

Note that there are 21 coefficients to be determined.

(ii) The values of the function and its first-order and second-order partial derivatives (i.e., z, z_x, z_y, z_{xx}, z_{xy}, and z_{yy}) are given at each vertex of the triangle. This assumption yields 18 independent conditions.

(iii) The partial derivative of the function differentiated in the direction perpendicular to each side of the triangle is a polynomial of degree three, at most, in the variable measured in the direction of the side of the triangle. In other words, when the coordinate system is transformed to another Cartesian system, which we call the s-t system, in such a

way that the s axis is parallel to each of the side of the triangle, the bivariate polynomial in s and t representing the z values must satisfy

$$z_{tssss} = 0 . \qquad (A-2)$$

Since a triangle has three sides, this assumption yields three additional conditions.

The purpose of the third assumption is two-fold. This assumption adds three independent conditions to the 18 conditions dictated by the second assumption and, thus, enables one to determine the 21 coefficients of the polynomial. It also assures smoothness of interpolated values as described in the following paragraph.

We will prove smoothness of the interpolated values and therefore smoothness of the resulting surface along the side of the triangle. Since the coordinate transformation between the x-y system and the s-t system is linear, the values of z_x, z_y, z_{xx}, z_{xy}, and z_{yy} at each vertex uniquely determine the values of z_s, z_t, z_{ss}, z_{st}, and z_{tt} at the same vertex, each of the latter as a linear combination of the former. Then, the z, z_s, and z_{ss} values at two vertexes uniquely determine a fifth-degree polynomial in s for z on the side between these vertexes. Since two fifth degree polynomials in x and y representing z values in two triangles that share the common side are reduced to fifth-degree polynomials in s on the side, these two polynomials in x and y coincide with each other on the common side. This proves continuity of the interpolated z values along a side of a triangle. Similarly, the values of z_t and $z_{st} = (z_t)_s$ at two vertexes uniquely determine a third-degree polynomial in s for z_t on the side. Since the polynomial representing z_t is assumed to be third degree at most with respect to s, two polynomials representing z_t in two triangles that share the common side also

coincide with each other on the side. This proves continuity of z_t and thus smoothness of z along the side of the triangle.

Coordinate System Associated With the Triangle.

We denote the vertexes of the triangle by V_1, V_2, and V_3 in a counter-clockwise order, and their respective coordinates in the x-y Cartesian coordinate system by (x_1, y_1), (x_2, y_2), and (x_3, y_3), as shown in figure A-1(a). We introduce a new coordinate system associated with the triangle, where the vertexes are represented by (0,0), (1,0), and (0,1) as shown in figure A-1(b). We call this new system the u-v system.

The coordinate transformation between the x-y system and the u-v system is represented by

$$\begin{aligned} x &= a\,u + b\,v + x_0 \,, \\ y &= c\,u + d\,v + y_0 \,, \end{aligned} \tag{A-3}$$

where

$$\begin{aligned} a &= x_2 - x_1 \,, \\ b &= x_3 - x_1 \,, \\ c &= y_2 - y_1 \,, \\ d &= y_3 - y_1 \,, \\ x_0 &= x_1 \,, \\ y_0 &= y_1 \,. \end{aligned} \tag{A-4}$$

The inverse relation is

$$\begin{aligned} u &= [\, d(x - x_0) - b(y - y_0)] \,/\, (a\,d - b\,c) \,, \\ v &= [-c(x - x_0) + a(y - y_0)] \,/\, (a\,d - b\,c) \,. \end{aligned} \tag{A-5}$$

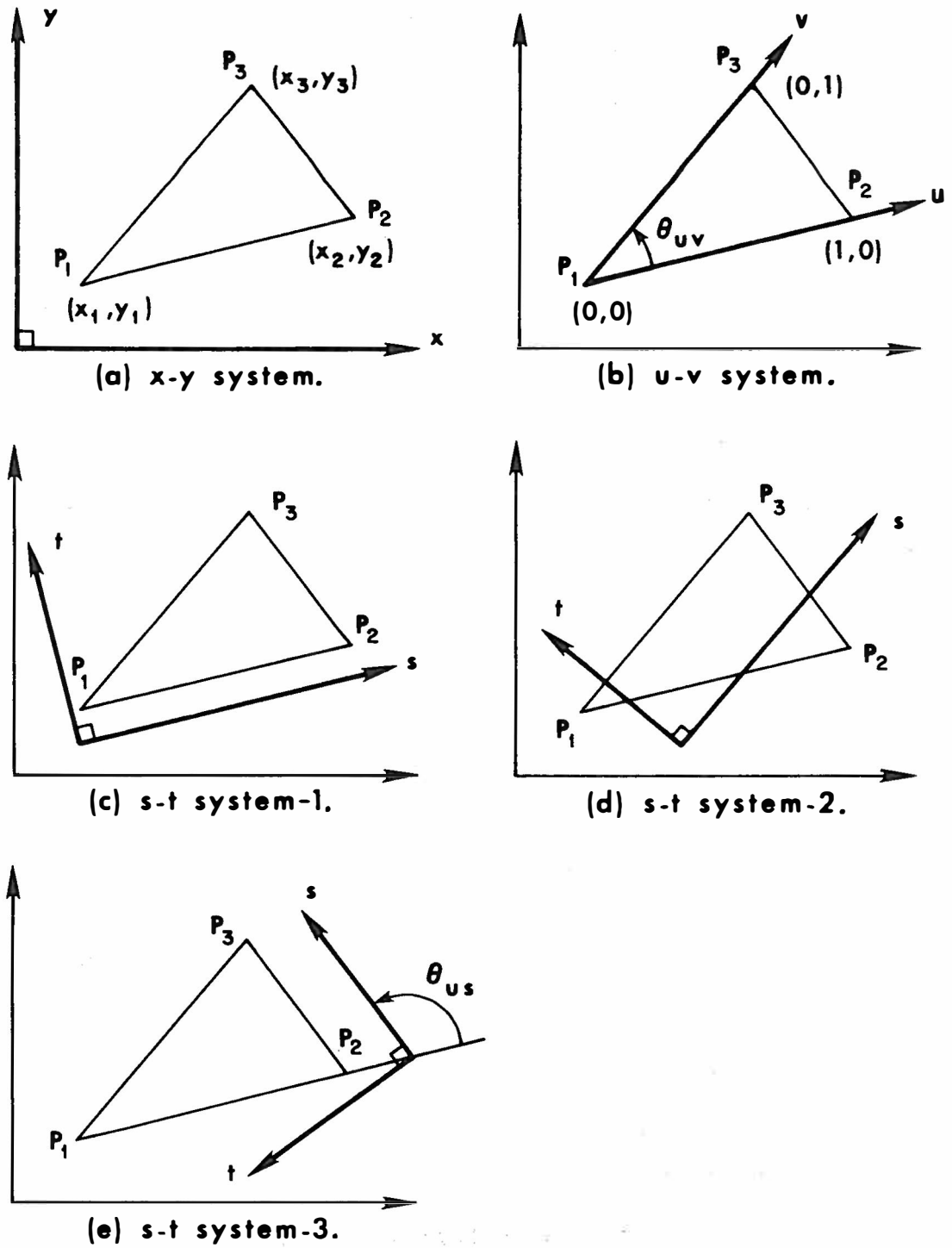

Figure A-1. Various coordinate systems.

The partial derivatives in the x-y system are transformed to the u-v system by

$$
\begin{aligned}
z_u &= a\,z_x + c\,z_y \,, \\
z_v &= b\,z_x + d\,z_y \,, \\
z_{uu} &= a^2 z_{xx} + 2\,a\,c\,z_{xy} + c^2 z_{yy} \,, \\
z_{uv} &= a\,b\,z_{xx} + (a\,d + b\,c)\,z_{xy} + c\,d\,z_{yy} \,, \\
z_{vv} &= b^2 z_{xx} + 2\,b\,d\,z_{xy} + d^2 z_{yy} \,.
\end{aligned}
\tag{A-6}
$$

Since this coordinate transformation is linear, the interpolating polynomial (A-1) is transformed to

$$
z(u,v) = \sum_{j=0}^{5} \sum_{k=0}^{5-j} p_{jk}\, u^j v^k \,. \tag{A-7}
$$

Since it is the p coefficients that are determined directly, as shown later, and are used for interpolating z values, it is unnecessary to relate the p coefficients to the q coefficients used in (A-1).

The partial derivatives of z(u, v) in the u-v system are expressed by

$$
\begin{aligned}
z_u(u,v) &= \sum_{j=1}^{5} \sum_{k=0}^{5-j} j\,p_{jk}\, u^{j-1} v^k \,, \\
z_v(u,v) &= \sum_{j=0}^{4} \sum_{k=1}^{5-j} k\,p_{jk}\, u^j v^{k-1} \,, \\
z_{uu}(u,v) &= \sum_{j=2}^{5} \sum_{k=0}^{5-j} j(j-1)\,p_{jk}\, u^{j-2} v^k \,, \\
z_{uv}(u,v) &= \sum_{j=1}^{4} \sum_{k=1}^{5-j} j\,k\,p_{jk}\, u^{j-1} v^{k-1} \,,
\end{aligned}
\tag{A-8}
$$

$$z_{vv}(u,v) = \sum_{j=0}^{3} \sum_{k=2}^{5-j} k(k-1)\, p_{jk}\, u^j v^{k-2} .$$

We denote the lengths of the unit vectors in the u-v system (i.e., the lengths of sides $\overline{V_1V_2}$ and $\overline{V_1V_3}$) by L_u and L_v, respectively, and the angle between the u and v axes by θ_{uv}. They are given by

$$\begin{aligned} L_u &= a^2 + c^2 , \\ L_v &= b^2 + d^2 , \\ \theta_{uv} &= \tan^{-1}(d/b) - \tan^{-1}(c/a) , \end{aligned} \tag{A-9}$$

where a, b, c, and d are constants given in (A-4).

Implementation of the Third Assumption.

We represent the third assumption (A-2) in the u-v system and derive useful equations for determining the coefficients of the polynomial. We do this for three cases corresponding to the three sides of the triangle.

First, we consider the case where the s axis is parallel to side $\overline{V_1V_2}$, as shown in figure A-1(c). The coordinate transformation between the u-v system and the s-t system is expressed by

$$\begin{aligned} u &= [(\sin\theta_{uv})(s - s_0) - (\cos\theta_{uv})(t - t_0)] / (L_u \sin\theta_{uv}) , \\ v &= (t - t_0) / (L_v \sin\theta_{uv}) , \end{aligned} \tag{A-10}$$

where L_u, L_v, and θ_{uv} are constants given in (A-9). Partial derivatives with respect to s and t are expressed by

$$\begin{aligned} \frac{\partial}{\partial s} &= \frac{1}{L_u} \frac{\partial}{\partial u} , \\ \frac{\partial}{\partial t} &= -\frac{\cos\theta_{uv}}{L_u \sin\theta_{uv}} \frac{\partial}{\partial u} + \frac{1}{L_v \sin\theta_{uv}} \frac{\partial}{\partial v} , \end{aligned} \tag{A-11}$$

respectively. From (A-2), (A-7), and (A-11), we obtain

$$L_u p_{41} - 5 L_v \cos\theta_{uv} p_{50} = 0 . \qquad \text{(A-12)}$$

Next, we consider the case where the s axis is parallel to side $\overline{V_1 V_3}$, as shown in figure A-1(d). The coordinate transformation is expressed by

$$\begin{aligned} u &= -(t - t_0) / (L_u \sin\theta_{uv}) , \\ v &= [(\sin\theta_{uv})(s - s_0) + (\cos\theta_{uv})(t - t_0)] / (L_v \sin\theta_{uv}) . \end{aligned} \qquad \text{(A-13)}$$

Partial derivatives are expressed by

$$\begin{aligned} \frac{\partial}{\partial s} &= \frac{1}{L_v} \frac{\partial}{\partial v} , \\ \frac{\partial}{\partial t} &= -\frac{1}{L_u \sin\theta_{uv}} \frac{\partial}{\partial u} + \frac{\cos\theta_{uv}}{L_v \sin\theta_{uv}} \frac{\partial}{\partial v} . \end{aligned} \qquad \text{(A-14)}$$

Then, from (A-2), (A-7), and (A-14), we obtain

$$L_v p_{14} - 5 L_u \cos\theta_{uv} p_{05} = 0 . \qquad \text{(A-15)}$$

Next, we consider the third case where the s axis is parallel to side $\overline{V_2 V_3}$, as shown in figure A-1(e). The coordinate transformation is expressed by

$$\begin{aligned} u &= A(s - s_0) + B(t - t_0) , \\ v &= C(s - s_0) + D(t - t_0) , \end{aligned} \qquad \text{(A-16)}$$

where

$$\begin{aligned} A &= \sin(\theta_{uv} - \theta_{us}) / (L_u \sin\theta_{uv}) , \\ B &= -\cos(\theta_{uv} - \theta_{us}) / (L_u \sin\theta_{uv}) , \\ C &= \sin\theta_{us} / (L_v \sin\theta_{uv}) , \\ D &= \cos\theta_{us} / (L_v \sin\theta_{uv}) , \end{aligned} \qquad \text{(A-17)}$$

$$\theta_{us} = \tan^{-1}[(d-c)/(b-a)] - \tan^{-1}(c/a) .$$

The θ_{us} constant is the angle between the s and the u axes. The a, b, c, and d constants are given in (A-4), and L_u, L_v, and θ_{uv} are given in (A-9). Partial derivatives with respect to s and t are expressed by

$$\frac{\partial}{\partial s} = A\frac{\partial}{\partial u} + C\frac{\partial}{\partial v} ,$$
$$\frac{\partial}{\partial t} = B\frac{\partial}{\partial u} + D\frac{\partial}{\partial v} . \qquad \text{(A-18)}$$

From (A-2), (A-7), and (A-18), we obtain

$$5A^4 B p_{50} + A^3(4BC + AD)p_{41} + A^2 C(3BC + 2AD)p_{32}$$
$$+ AC^2(2BC + 3AD)p_{23} + C^3(BC + 4AD)p_{14} + 5C^4 D p_{05} = 0 . \qquad \text{(A-19)}$$

Equations (A-12), (A-15), and (A-19) are the results of implementation of the third assumption (A-2) in the u-v coordinate system. They are used for determining the coefficients of the polynomial (A-7).

Determination of the Coefficients of the Polynomial.

Obviously, we can determine the coefficients of the lower-power terms by letting u = 0 and v= 0 and by inserting the values of z, z_u, z_v, z_{uu}, z_{uv}, and z_{vv} at V_1 (i.e., u = 0 and v = 0) in (A-7) and (A-8). The results are

$$p_{00} = z(0,0) ,$$
$$p_{10} = z_u(0,0) ,$$
$$p_{01} = z_v(0,0) ,$$
$$p_{20} = z_{uu}(0,0)/2 , \qquad \text{(A-20)}$$

$$p_{11} = z_{uv}(0,0) ,$$

$$p_{02} = z_{vv}(0,0)/2 .$$

Next, letting u = 1 and v = 0 and inserting the values of z, z_u, and z_{uu} at V_2 (i.e., u = 1 and v = 0) in (A-7) and the first and the third equations in (A-8), we obtain the following three equations:

$$\begin{aligned} p_{30} + \quad p_{40} + \quad p_{50} &= z(1,0) - p_{00} - p_{10} - p_{20} , \\ 3p_{30} + \; 4p_{40} + \; 5p_{50} &= z_u(1,0) - p_{10} - 2p_{20} , \\ 6p_{30} + 12p_{40} + 20p_{50} &= z_{uu}(1,0) - 2p_{20} . \end{aligned}$$

Solving these equations with respect to p_{30}, p_{40}, and p_{50}, we obtain

$$\begin{aligned} p_{30} &= [20z(1,0) - 8z_u(1,0) + z_{uu}(1,0) - 20p_{00} - 12p_{10} - 6p_{20}]/2, \\ p_{40} &= -15z(1,0) + 7z_u(1,0) - z_{uu}(1,0) + 15p_{00} + 8p_{10} + 3p_{20} , \\ p_{50} &= [12z(1,0) - 6z_u(1,0) + z_{uu}(1,0) - 12p_{00} - 6p_{10} - 2p_{20}]/2 . \end{aligned} \qquad \text{(A-21)}$$

Since p_{00}, p_{10}, and p_{20} are already determined by (A-20), we can calculate p_{30}, p_{40}, and p_{50} from (A-21).

Similarly, using the values of z, z_v, and z_{vv} at V_3 (i.e., u = 0 and v = 1) and working with (A-7) and the second and the last equations in (A-8), we obtain

$$\begin{aligned} p_{03} &= [20z(0,1) - 8z_v(0,1) + z_{vv}(0,1) - 20p_{00} - 12p_{01} - 6p_{02}]/2, \\ p_{04} &= -15z(0,1) + 7z_v(0,1) - z_{vv}(0,1) + 15p_{00} + 8p_{01} + 3p_{02} , \\ p_{05} &= [12z(0,1) - 6z_v(0,1) + z_{vv}(0,1) - 12p_{00} - 6p_{01} - 2p_{02}]/2 . \end{aligned} \qquad \text{(A-22)}$$

With p_{50} and p_{05} determined, we can determine p_{41} and p_{14} from (A-12) and (A-15), respectively. The results are

$$p_{41} = \frac{5 L_v \cos \theta_{uv}}{L_u} p_{50} ,$$

$$p_{14} = \frac{5 L_u \cos \theta_{uv}}{L_v} p_{05} . \qquad \text{(A-23)}$$

Next, we use the values of z_v and z_{uv} at V_2 (i.e., u = 1 and v = 0) with the second and the fourth equations in (A-8) and obtain

$$p_{21} + p_{31} = z_v(1,0) - p_{01} - p_{11} - p_{41} ,$$

$$2 p_{21} + 3 p_{31} = z_{uv}(1,0) - p_{11} - 4 p_{41} .$$

Solving these equations, we obtain

$$\begin{aligned} p_{21} &= 3 z_v(1,0) - z_{uv}(1,0) - 3 p_{01} - 2 p_{11} + p_{41} , \\ p_{31} &= -2 z_v(1,0) + z_{uv}(1,0) + 2 p_{01} + p_{11} - 2 p_{41} . \end{aligned} \qquad \text{(A-24)}$$

Similarly, using the values of z_u and z_{uv} at V_3 (i.e., u = 0 and v = 1) with the first and the fourth equations in (A-8), we obtain

$$\begin{aligned} p_{12} &= 3 z_u(0,1) - z_{uv}(0,1) - 3 p_{10} - 2 p_{11} + p_{14} , \\ p_{13} &= -2 z_u(0,1) + z_{uv}(0,1) + 2 p_{10} + p_{11} - 2 p_{14} . \end{aligned} \qquad \text{(A-25)}$$

Equation (A-19) is rewritten as

$$g_1 p_{32} + g_2 p_{23} = h_1 , \qquad \text{(A-26)}$$

where

$$\begin{aligned} g_1 &= A^2 C (3 BC + 2 AD) , \\ g_2 &= A C^2 (2 BC + 3 AD) , \end{aligned} \qquad \text{(A-27)}$$

$$h_1 = -5A^4 B p_{50} - A^3(4BC + AD) p_{41}$$

$$- C^3(BC + 4AD) p_{14} - 5C^4 D p_{05} ,$$

with A, B, C, and D defined by (A-17). From the value of z_{vv} at V_2 and the last equation in (A-8), we obtain

$$p_{22} + p_{32} = h_2 , \tag{A-28}$$

where

$$h_2 = (1/2) z_{vv}(1,0) - p_{02} - p_{12} . \tag{A-29}$$

Similarly, from the value of z_{uu} at V_3 and the third equation in (A-8), we obtain

$$p_{22} + p_{23} = h_3 , \tag{A-30}$$

where

$$h_3 = (1/2) z_{uu}(0,1) - p_{20} - p_{21} . \tag{A-31}$$

Solving (A-26), (A-28), and (A-30) with respect to p_{22}, p_{32}, and p_{23}, we obtain

$$\begin{aligned} p_{22} &= (g_1 h_2 + g_2 h_3 - h_1)/(g_1 + g_2) , \\ p_{32} &= h_2 - p_{22} , \\ p_{23} &= h_3 - p_{22} , \end{aligned} \tag{A-32}$$

with g_1, g_2, h_1, h_2, and h_3 given by (A-27), (A-29), and (A-31).

Step-by-Step Description of the Procedure.

In summary, the coefficients of the polynomial are determined by the following steps:

(i) Determine a, b, c, and d (coefficients for coordinate transformation) from (A-4).

(ii) Calculate partial derivatives z_u, z_v, z_{uu}, z_{uv}, and z_{vv} from (A-6).

(iii) Calculate L_u, L_v, and θ_{uv} (constants associated with the u-v coordinate system) from (A-9).

(iv) Calculate A, B, C, and D (coefficients for another coordinate transformation) from (A-17).

(v) Determine 18 coefficients of the polynomial from (A-20), (A-21), (A-22), (A-23), (A-24), and (A-25) -- in this order.

(vi) Calculate g_1, g_2, h_1, h_2, and h_3 from (A-27), (A-29), and (A-31).

(vii) Determine the remaining three coefficients from (A-32).

For a given point (x,y) in the triangle, one can interpolate the z value by the following steps:

(i) Transform x and y to u and v by (A-5) with necessary coefficients given by (A-4).

(ii) Evaluate the polynomial for z(u, v) given in (A-7).

Although some equations look complicated, the procedure described here is straightforward. It can easily be implemented as a computer subroutine.

APPENDIX B

COMPUTER SUBPROGRAM PACKAGE

User information and Fortran listings of the IDBVIP/IDSFFT subprogram package are given in this appendix. This package implements the method of bivariate interpolation and smooth surface fitting for irregularly distributed data points, described in section 2 of this report. It is written in ANSI Standard Fortran (ANSI, 1966).

The package consists of a block-data subprogram and the following six subroutines; i.e., IDBVIP, IDGEOM, IDLCTN, IDPDRV, IDPTIP, and IDSFFT. Two subroutines, IDBVIP and IDSFFT, are the master subroutines of the package, and each interfaces with the user. The remaining four subroutines are common supporting subroutines called by IDBVIP and IDSFFT. The IDBVIP subroutine performs bivariate interpolation for irregularly distributed data points; it estimates the z values at the specified points in the x-y plane. The IDSFFT subroutine performs smooth surface fitting; it estimates the z values at the specified rectangular grid points in the x-y plane and generates a doubly-dimensioned array containing these estimated values.

The package includes three common blocks; i.e., IDGM, IDNN, and IDPI. Including these common areas, the package occupies approximately 3200 locations on the CDC-6600 computer.

When the user wishes to call either IDBVIP or IDSFFT subroutine repeatedly with identical data as parts of input data in two consecutive calls, he can save computation times considerably by specifying an appropriate mode of computation. (This mode is specified with the MD parameter in the call statements to be described later.)

User information on IDBVIP and that of IDSFFT will follow. This information is followed by Fortran listings of the seven subprograms -- six subroutines listed in alphabetical order, followed by the block-data subprogram.

The IDBVIP Subroutine.

This subroutine performs bivariate interpolation when the projections of the data points in the x-y plane are irregularly distributed in the plane.

This subroutine is called by the following statement:

```
CALL IDBVIP(MD,NDP,XD,YD,ZD,WK,NIP,XI,YI,ZI)
```

In this call statement, the input parameters are

MD = mode of computation (must be 1, 2, or 3),
= 1 for new XD-YD,
= 2 for old XD-YD, new XI-YI,
= 3 for old XD-YD, old XI-YI,

NDP = number of data points (must be 4 or greater),

XD = array of dimension NDP containing the x coordinates of the data points,

YD = array of dimension NDP containing the y coordinates of the data points,

ZD = array of dimension NDP containing the z coordinates of the data points,

WK = array of dimension (2*NDP+NNP+5)*NDP+NIP to be used internally as a work area,

NIP = number of points to be interpolated at (must be 1 or greater),

XI = array of dimension NIP containing the x coordinates of the points to be interpolated at,

YI = array of dimension NIP containing the y coordinates of the points to be interpolated at,

where NNP is the number of additional data points used for estimating partial derivatives at each data point. The output parameter is

ZI = array of dimension NIP, where the z coordinates of the interpolated points will be stored.

The LUN constant in the data initialization statement is the logical unit number of the standard output unit and is, therefore, system dependent. The user must enter an appropriate number into LUN before compiling this subroutine.

The value of NNP must be given through the IDNN common block. NNP must be 2 or greater, but smaller than NDP. In the subprogram package listed below, it is set to 4. The user can change it by declaring

```
COMMON/IDNN/NNP
```

in his calling program and by assigning a number of his choice to NNP with an arithmetic assignment statement before the call to IDBVIP.

The call to this subroutine with MD = 2 must be preceded by another call to this subroutine with the same NDP value and with the same contents of the XD and YD arrays. The call with MD = 3 must be preceded by another call with the same NDP and NIP values and with the same contents of the XD, YD, XI, and YI arrays. Between the call with MD = 2 or 3 and its preceding call, the WK array should not be disturbed.

Table B-1 (p. 32) shows the approximate computation times required on the CDC-6600 computer.

The IDSFFT Subroutine.

This subroutine performs smooth surface fitting when the projections of the data points in the x-y plane are irregularly distributed in the plane.

This subroutine is called by the following statement:

CALL IDSFFT (MD, NDP, XD, YD, ZD, WK, NXI, NYI, XI, YI, ZI)

In this call statement, the input parameters are

MD = mode of computation (must be 1, 2, or 3),
= 1 for new XD-YD,
= 2 for old XD-YD, new XI-YI,
= 3 for old XD-YD, old XI-YI,

NDP = number of data points (must be 4 or greater),

XD = array of dimension NDP containing the x coordinates of the data points,

YD = array of dimension NDP containing the y coordinates of the data points,

ZD = array of dimension NDP containing the z coordinates of the data points,

WK = array of dimension (2 * NDP + NNP + 5) * NDP + NXI * NYI to be used internally as a work area,

NXI = number of output grid points in the x coordinate (must be 1 or greater),

NYI = number of output grid points in the y coordinate (must be 1 or greater),

XI = array of dimension NXI containing the x coordinates of the output grid points,

YI = array of dimension NYI containing the y coordinates of the output grid points,

where NNP is the number of additional data points used for estimating partial derivatives at each data point. The output parameter is

ZI = doubly-dimensioned array of dimension (NXI, NYI), where the interpolated z values at the output grid points will be stored.

The LUN constant in the data initialization statement is the logical unit number of the standard output unit and is, therefore, system dependent. The user must enter an appropriate number into LUN before compiling this subroutine.

The value of NNP must be given through the IDNN common block. NNP must be 2 or greater, but smaller than NDP. In the subprogram package listed below, it is set to 4. The user can change it by declaring

COMMON/IDNN/NNP

in his calling program and by assigning a number of his choice to NNP with an arithmetic assignment statement before the call to this subroutine.

The call to this subroutine with MD = 2 must be preceded by another call to this subroutine with the same NDP value and with the same contents of the XD and YD arrays. The call with MD = 3 must be preceded by another call with the same NDP, NXI, and NYI values and with the same contents of the XD, YD, XI, and YI arrays. Between the call with MD = 2 or 3 and its preceding call, the WK array should not be disturbed.

Table B-2 (p. 32) shows the approximate computation times required on the CDC-6600 computer.

Table B-1. Approximate computation times required for the IDBVIP subroutine on the CDC-6600 computer.

NDP	NIP	Time (seconds)		
		MD = 1	MD = 2	MD = 3
20	10	0.40	0.03	0.02
	100	0.50	0.12	0.06
	1000	1.4	1.0	0.35
30	10	1.3	0.04	0.03
	100	1.5	0.16	0.07
	1000	2.7	1.4	0.50
50	10	6.6	0.05	0.04
	100	6.8	0.24	0.10
	1000	8.8	2.2	0.70

Table B-2. Approximate computation times required for the IDSFFT subroutine on the CDC-6600 computer.

NDP	NXI * NYI	Time (seconds)		
		MD = 1	MD = 2	MD = 3
20	11 * 11	0.50	0.12	0.07
	33 * 33	1.1	0.70	0.40
	101 * 101	5.8	5.4	3.4
30	11 * 11	1.5	0.16	0.08
	33 * 33	2.1	0.85	0.41
	101 * 101	7.3	6.0	3.5
50	11 * 11	6.8	0.22	0.11
	33 * 33	7.8	1.2	0.50
	101 * 101	14.0	7.3	3.7

```
      SUBROUTINE  IDBVIP(MD,NDP,XD,YD,ZD,WK,NIP,XI,YI,ZI)                IBI 001
C THIS SUBROUTINE PERFORMS BIVARIATE INTERPOLATION WHEN THE PRO-          IBI 002
C JECTIONS OF THE DATA POINTS IN THE X-Y PLANE ARE IRREGULARLY            IBI 003
C DISTRIBUTED IN THE PLANE.                                               IBI 004
C THE INPUT PARAMETERS ARE                                                IBI 005
C     MD  = MODE OF COMPUTATION (MUST BE 1, 2, OR 3),                     IBI 006
C         = 1 FOR NEW XD-YD,                                              IBI 007
C         = 2 FOR OLD XD-YD, NEW XI-YI,                                   IBI 008
C         = 3 FOR OLD XD-YD, OLD XI-YI,                                   IBI 009
C     NDP = NUMBER OF DATA POINTS (MUST BE 4 OR GREATER),                 IBI 010
C     XD  = ARRAY OF DIMENSION NDP STORING THE X COORDINATES              IBI 011
C           OF THE DATA POINTS,                                           IBI 012
C     YD  = ARRAY OF DIMENSION NDP STORING THE Y COORDINATES              IBI 013
C           OF THE DATA POINTS,                                           IBI 014
C     ZD  = ARRAY OF DIMENSION NDP STORING THE Z COORDINATES              IBI 015
C           OF THE DATA POINTS,                                           IBI 016
C     WK  = ARRAY OF DIMENSION (2*NDP+NNP+5)*NDP+NIP                      IBI 017
C           TO BE USED AS A WORK AREA,                                    IBI 018
C     NIP = NUMBER OF INTERPOLATED POINTS                                 IBI 019
C           (MUST BE 1 OR GREATER),                                       IBI 020
C     XI  = ARRAY OF DIMENSION NIP STORING THE X COORDINATES              IBI 021
C           OF THE INTERPOLATED POINTS,                                   IBI 022
C     YI  = ARRAY OF DIMENSION NIP STORING THE Y COORDINATES              IBI 023
C           OF THE INTERPOLATED POINTS,                                   IBI 024
C WHERE NNP IS THE NUMBER OF ADDITIONAL DATA POINTS USED FOR              IBI 025
C ESTIMATING PARTIAL DERIVATIVES AT EACH DATA POINT.  THE VALUE           IBI 026
C OF NNP MUST BE GIVEN THROUGH THE IDNN COMMON.  NNP MUST BE 2            IBI 027
C OR GREATER, BUT SMALLER THAN NDP.                                       IBI 028
C THE OUTPUT PARAMETER IS                                                 IBI 029
C     ZI  = ARRAY OF DIMENSION NIP, WHERE THE Z COORDINATES               IBI 030
C           OF THE INTERPOLATED POINTS ARE TO BE DISPLAYED.               IBI 031
C THE LUN CONSTANT IN THE DATA INITIALIZATION STATEMENT IS THE            IBI 032
C LOGICAL UNIT NUMBER OF THE STANDARD OUTPUT UNIT AND IS,                 IBI 033
C THEREFORE, SYSTEM DEPENDENT.                                            IBI 034
C DECLARATION STATEMENTS                                                  IBI 035
      DIMENSION   XD(10),YD(10),ZD(10),WK(1000),                          IBI 036
     1            XI(10),YI(10),ZI(10)                                    IBI 037
      COMMON/IDNN/NNP                                                     IBI 038
      COMMON/IDGM/NDPC,NNPC,NT,NL                                         IBI 039
      COMMON/IDPI/NCF,ICF                                                 IBI 040
      EQUIVALENCE (FNDPO,NDPO),(FNDPPV,NDPPV),                            IBI 041
     1            (FNNPO,NNPO),(FNNPPV,NNPPV),                            IBI 042
     2            (FNIPO,NIPO),(FNIPPV,NIPPV),                            IBI 043
     3            (FNT,NT),(FNL,NL)                                       IBI 044
      DATA  LUN/6/                                                        IBI 045
C SETTING OF SOME INPUT PARAMETERS TO LOCAL VARIABLES.  (ALL MD)          IBI 046
   10 MDO=MD                                                              IBI 047
      NDPO=NDP                                                            IBI 048
      NDPC=NDPO                                                           IBI 049
      NIPO=NIP                                                            IBI 050
      NNPO=NNP                                                            IBI 051
      NNPC=NNPO                                                           IBI 052
C ERROR CHECK.  (ALL MD)                                                  IBI 053
   20 IF(MDO.LT.1.OR.MDO.GT.3)      GO TO 90                             IBI 054
      IF(NDPO.LT.4)         GO TO 90                                      IBI 055
      IF(NIPO.LT.1)         GO TO 90                                      IBI 056
      IF(NNPO.LT.2.OR.NNPO.GE.NDPO) GO TO 90                              IBI 057
      IF(MDO.NE.1)          GO TO 22                                      IBI 058
   21 WK(1)=FNDPO                                                         IBI 059
      WK(2)=FNNPO                                                         IBI 060
      GO TO 24                                                            IBI 061
   22 FNDPPV=WK(1)                                                        IBI 062
      FNNPPV=WK(2)                                                        IBI 063
      IF(NDPO.NE.NDPPV)    GO TO 90                                       IBI 064
      IF(NNPO.NE.NNPPV)    GO TO 90                                       IBI 065
```

```
      IF(MD0.NE.3)         GO TO 24                                    IBI 066
   23 FNIPPV=WK(3)                                                     IBI 067
      IF(NIP0.NE.NIPPV)    GO TO 90                                    IBI 068
      GO TO 30                                                         IBI 069
   24 WK(3)=FNIP0                                                      IBI 070
C ALLOCATION OF STORAGE AREAS IN THE WK ARRAY.  (ALL MD)               IBI 071
   30 NDNDM1=NDP0*(NDP0-1)                                             IBI 072
      IWIPT=7                                                          IBI 073
      IWIPL=IWIPT+NDNDM1                                               IBI 074
      IWIPN=IWIPL+NDNDM1                                               IBI 075
      IWPD =IWIPN+NDP0*NNP0                                            IBI 076
      IWIT =IWPD +NDP0*5                                               IBI 077
C DIVIDES THE X-Y PLANE INTO A NUMBER OF TRIANGLES AND                 IBI 078
C DETERMINES NNP POINTS NEAREST EACH DATA POINT.  (MD=1)               IBI 079
   40 IF(MD.GT.1)     GO TO 42                                         IBI 080
   41 CALL IDGEOM(XD,YD,WK(IWIPT),WK(IWIPL),WK(IWIPN))                 IBI 081
      WK(5)=FNT                                                        IBI 082
      WK(6)=FNL                                                        IBI 083
      GO TO 50                                                         IBI 084
   42 FNT=WK(5)                                                        IBI 085
      FNL=WK(6)                                                        IBI 086
C ESTIMATES PARTIAL DERIVATIVES AT ALL DATA POINTS.  (ALL MD)          IBI 087
   50 CALL IDPDRV(XD,YD,ZD,WK(IWIPN),WK(IWPD))                         IBI 088
C LOCATES ALL INTERPOLATED POINTS.  (MD=1,2)                           IBI 089
   60 IF(MD0.EQ.3)    GO TO 70                                         IBI 090
      JWIT=IWIT-1                                                      IBI 091
      DO 61  IIP=1,NIP0                                                IBI 092
        JWIT=JWIT+1                                                    IBI 093
        CALL IDLCTN(XD,YD,WK(IWIPT),WK(IWIPL),                         IBI 094
     1              XI(IIP),YI(IIP),WK(JWIT))                          IBI 095
   61 CONTINUE                                                         IBI 096
C INTERPOLATION OF THE ZI VALUES.  (ALL MD)                            IBI 097
   70 NCF=0                                                            IBI 098
      ICF=0                                                            IBI 099
      JWIT=IWIT-1                                                      IBI 100
      DO 71  IIP=1,NIP0                                                IBI 101
        JWIT=JWIT+1                                                    IBI 102
        CALL IDPTIP(XD,YD,ZD,WK(IWIPT),WK(IWIPL),WK(IWPD),             IBI 103
     1              WK(JWIT),XI(IIP),YI(IIP),ZI(IIP))                  IBI 104
   71 CONTINUE                                                         IBI 105
C NORMAL EXIT                                                          IBI 106
   80 RETURN                                                           IBI 107
C ERROR EXIT                                                           IBI 108
   90 WRITE (LUN,2090) MD0,NDP0,NIP0,NNP0                              IBI 109
      RETURN                                                           IBI 110
C FORMAT STATEMENT FOR ERROR MESSAGE                                   IBI 111
 2090 FORMAT(1X/41H ***   IMPROPER INPUT PARAMETER VALUE(S)./           IBI 112
     1   7H    MD =,I4,10X,5HNDP =,I6,10X,5HNIP =,I6,                 IBI 113
     2   10X,5HNNP =,I6/                                             IBI 114
     3   35H ERROR DETECTED IN ROUTINE   IDBVIP/)                    IBI 115
      END                                                              IBI 116


      SUBROUTINE  IDGEOM(XD,YD,IPT,IPL,IPN)                            IGM 001
C THIS SUBROUTINE DIVIDES THE X-Y PLANE INTO A NUMBER OF               IGM 002
C TRIANGULAR AREAS ACCORDING TO GIVEN DATA POINTS IN THE PLANE,        IGM 003
C DETERMINES LINE SEGMENTS THAT FORM THE BORDER OF DATA AREA,          IGM 004
C DETERMINES THE TRIANGLE NUMBERS CORRESPONDING TO THE BORDER          IGM 005
C LINE SEGMENTS, AND SELECTS SEVERAL DATA POINTS THAT ARE              IGM 006
C NEAREST TO EACH OF THE DATA POINTS.                                  IGM 007
C AT COMPLETION, POINT NUMBERS OF THE VERTEXES OF EACH TRIANGLE        IGM 008
C ARE LISTED COUNTER-CLOCKWISE.  POINT NUMBERS OF THE END POINTS       IGM 009
C OF EACH BORDER LINE SEGMENT ARE LISTED COUNTER-CLOCKWISE,            IGM 010
C LISTING ORDER OF THE LINE SEGMENTS BEING COUNTER-CLOCKWISE.          IGM 011
```

```
C THE INPUT PARAMETERS ARE                                              IGM 012
C     XD,YD = ARRAYS STORING THE X AND Y COORDINATES, RESP.,            IGM 013
C             OF DATA POINTS.                                           IGM 014
C THE OUTPUT PARAMETERS ARE                                             IGM 015
C     IPT = ARRAY OF DIMENSION 3*NT, WHERE THE POINT NUMBERS            IGM 016
C           OF THE VERTEXES OF THE (IT)TH TRIANGLE ARE TO BE            IGM 017
C           DISPLAYED AS THE (3*IT-2)ND, (3*IT-1)ST, AND                IGM 018
C           (3*IT)TH ELEMENTS, IT=1,2,...,NT,                           IGM 019
C     IPL = ARRAY OF DIMENSION 3*NL, WHERE THE POINT NUMBERS            IGM 020
C           OF THE END POINTS OF THE (IL)TH BORDER LINE                 IGM 021
C           SEGMENT AND ITS RESPECTIVE TRIANGLE NUMBER ARE              IGM 022
C           TO BE DISPLAYED AS THE (3*IL-2)ND, (3*IL-1)ST,              IGM 023
C           AND (3*IL)TH ELEMENTS, IL=1,2,..., NL,                      IGM 024
C     IPN = ARRAY OF DIMENSION NDP*NNP, WHERE THE POINT                 IGM 025
C           NUMBERS OF NNP DATA POINTS NEAREST TO EACH OF               IGM 026
C           THE DATA POINTS ARE TO BE DISPLAYED,                        IGM 027
C WHERE NDP IS THE TOTAL NUMBER OF DATA POINTS, NNP IS THE              IGM 028
C NUMBER OF DATA POINTS NEAREST TO EACH DATA POINT, NL IS               IGM 029
C THE NUMBER OF BORDER LINE SEGMENTS, AND NT IS THE NUMBER              IGM 030
C OF TRIANGLES.  NDP AND NNP ARE GIVEN TO THIS SUBROUTINE               IGM 031
C THROUGH THE IDGM COMMON.  NL AND NT ARE CALCULATED BY THIS            IGM 032
C SUBROUTINE AND ARE LEFT IN THE IDGM COMMON AT COMPLETION.             IGM 033
C DECLARATION STATEMENTS                                                IGM 034
      DIMENSION   XD(10),YD(10),IPT(100),IPL(100),IPN(50)               IGM 035
      COMMON/IDGM/NDP,NNP,NT,NL                                         IGM 036
      EQUIVALENCE (DSQ1,IDSQ1),(DSQ2,IDSQ2),(DSQM,IDSQM)                IGM 037
C PRELIMINARY PROCESSING                                                IGM 038
   10 NDP0=NDP                                                          IGM 039
      NDPM1=NDP0-1                                                      IGM 040
      NNP0=NNP                                                          IGM 041
      NNPM1=NNP0-1                                                      IGM 042
C DETERMINES THE NEAREST NNP POINTS.                                    IGM 043
   20 DO 29  IP1=1,NDP0                                                 IGM 044
        X1=XD(IP1)                                                      IGM 045
        Y1=YD(IP1)                                                      IGM 046
        J1MX=IP1*NNP0                                                   IGM 047
        J1MN=J1MX-NNPM1                                                 IGM 048
        DO 28  J1=J1MN,J1MX                                             IGM 049
          J2MX=J1-1                                                     IGM 050
          IDMN=0                                                        IGM 051
          DO 27  IP2=1,NDP0                                             IGM 052
            IF(IP2.EQ.IP1)          GO TO 27                            IGM 053
            IF(J1.GT.J1MN)          GO TO 22                            IGM 054
   21       DSQ1=(XD(IP2)-X1)**2+(YD(IP2)-Y1)**2                        IGM 055
            IPT(IP2)=IDSQ1                                              IGM 056
            GO TO 23                                                    IGM 057
   22       IDSQ1=IPT(IP2)                                              IGM 058
   23       IF(IDMN.EQ.0)           GO TO 24                            IGM 059
            IF(DSQ1.GE.DSQMN)       GO TO 27                            IGM 060
   24       IF(J1MN.GT.J2MX)        GO TO 26                            IGM 061
            DO 25  J2=J1MN,J2MX                                         IGM 062
              IF(IP2.EQ.IPN(J2))    GO TO 27                            IGM 063
   25       CONTINUE                                                    IGM 064
   26       DSQMN=DSQ1                                                  IGM 065
            IDMN=IP2                                                    IGM 066
   27     CONTINUE                                                     IGM 067
          IPN(J1)=IDMN                                                  IGM 068
   28   CONTINUF                                                       IGM 069
   29 CONTINUE                                                          IGM 070
C LISTS ALL THE POSSIBLE LINE SEGMENTS IN THE IPL ARRAY,                IGM 071
C CALCULATES THE SQUARES OF THE LINE SEGMENT LENGTHS, AND STORE         IGM 072
C THEM IN THE IPT ARRAY.                                                IGM 073
   30 IL=0                                                              IGM 074
      DO 32  IP1=1,NDPM1                                                IGM 075
        X1=XD(IP1)                                                      IGM 076
```

```
      Y1=YD(IP1)                                                        IGM 077
      IP1P1=IP1+1                                                       IGM 078
      DO 31  IP2=IP1P1,NDP0                                             IGM 079
        IL=IL+1                                                         IGM 080
        ILT2=IL+IL                                                      IGM 081
        IPL(ILT2-1)=IP1                                                 IGM 082
        IPL(ILT2)  =IP2                                                 IGM 083
        DSQ1=(XD(IP2)-X1)**2+(YD(IP2)-Y1)**2                            IGM 084
        IPT(IL)=IDSQ1                                                   IGM 085
   31   CONTINUE                                                        IGM 086
   32 CONTINUE                                                          IGM 087
      NL0=IL                                                            IGM 088
C SORTS THE IPL AND IPT ARRAYS IN ASCENDING ORDER OF THE LINE           IGM 089
C SEGMENT LENGTH (DISTANCE).                                            IGM 090
   35 NLM1=NL0-1                                                        IGM 091
      DO 37  IL1=1,NLM1                                                 IGM 092
        IDSQ1=IPT(IL1)                                                  IGM 093
        ILM=IL1                                                         IGM 094
        DSQM=DSQ1                                                       IGM 095
        IL2MN=IL1+1                                                     IGM 096
        DO 36  IL2=IL2MN,NL0                                            IGM 097
          IDSQ2=IPT(IL2)                                                IGM 098
          IF(DSQ2.GE.DSQM)      GO TO 36                                IGM 099
          ILM=IL2                                                       IGM 100
          DSQM=DSQ2                                                     IGM 101
   36   CONTINUE                                                        IGM 102
        IPT(ILM)=IDSQ1                                                  IGM 103
        IPT(IL1)=IDSQM                                                  IGM 104
        IL1T2=IL1+IL1                                                   IGM 105
        ILMT2=ILM+ILM                                                   IGM 106
        ITS=IPL(IL1T2-1)                                                IGM 107
        IPL(IL1T2-1)=IPL(ILMT2-1)                                       IGM 108
        IPL(ILMT2-1)=ITS                                                IGM 109
        ITS=IPL(IL1T2)                                                  IGM 110
        IPL(IL1T2)=IPL(ILMT2)                                           IGM 111
        IPL(ILMT2)=ITS                                                  IGM 112
   37 CONTINUE                                                          IGM 113
C ELIMINATES LINE SEGMENTS THAT CROSS OR LIE OVER SHORTER ONE.          IGM 114
   40 IL0=1                                                             IGM 115
      DO 46  IL1=2,NL0                                                  IGM 116
        IL1T2=IL1+IL1                                                   IGM 117
        IP1=IPL(IL1T2-1)                                                IGM 118
        IP2=IPL(IL1T2)                                                  IGM 119
        X1=XD(IP1)                                                      IGM 120
        X2=XD(IP2)                                                      IGM 121
        Y1=YD(IP1)                                                      IGM 122
        Y2=YD(IP2)                                                      IGM 123
        DX21=X2-X1                                                      IGM 124
        DY21=Y2-Y1                                                      IGM 125
        DO 45  IL2=1,IL0                                                IGM 126
          IL2T2=IL2+IL2                                                 IGM 127
          IP3=IPL(IL2T2-1)                                              IGM 128
          IP4=IPL(IL2T2)                                                IGM 129
          X3=XD(IP3)                                                    IGM 130
          X4=XD(IP4)                                                    IGM 131
          Y3=YD(IP3)                                                    IGM 132
          Y4=YD(IP4)                                                    IGM 133
          DX43=X4-X3                                                    IGM 134
          DX42=X4-X2                                                    IGM 135
          DX41=X4-X1                                                    IGM 136
          DX32=X3-X2                                                    IGM 137
          DX31=X3-X1                                                    IGM 138
          DY43=Y4-Y3                                                    IGM 139
          DY42=Y4-Y2                                                    IGM 140
          DY41=Y4-Y1                                                    IGM 141
```

```
          DY32=Y3-Y2                                                    IGM 142
          DY31=Y3-Y1                                                    IGM 143
          IF(IP3.NE.IP1)  GO TO 41                                      IGM 144
          IF(DY41*DX21-DX41*DY21.NE.0.0) GO TO 45                       IGM 145
          IF(DX41*DX21+DY41*DY21)         45,45,46                      IGM 146
   41     IF(IP4.NE.IP1)  GO TO 42                                      IGM 147
          IF(DY31*DX21-DX31*DY21.NE.0.0) GO TO 45                       IGM 148
          IF(DX31*DX21+DY31*DY21)         45,45,46                      IGM 149
   42     IF(IP3.NE.IP2)  GO TO 43                                      IGM 150
          IF(DY42*DX21-DX42*DY21.NE.0.0) GO TO 45                       IGM 151
          IF(DX42*DX21+DY42*DY21)         46,45,45                      IGM 152
   43     IF(IP4.NE.IP2)  GO TO 44                                      IGM 153
          IF(DY32*DX21-DX32*DY21.NE.0.0) GO TO 45                       IGM 154
          IF(DX32*DX21+DY32*DY21)         46,45,45                      IGM 155
   44     IF((DY31*DX21-DX31*DY21)*(DY41*DX21-DX41*DY21).GE.0.0)        IGM 156
     1                                    GO TO 45                      IGM 157
          IF((DY31*DX43-DX31*DY43)*(DY32*DX43-DX32*DY43).LT.0.0)        IGM 158
     1                                    GO TO 46                      IGM 159
   45   CONTINUE                                                        IGM 160
        IL0=IL0+1                                                       IGM 161
        IL0T2=IL0+IL0                                                   IGM 162
        IPL(IL0T2-1)=IP1                                                IGM 163
        IPL(IL0T2)  =IP2                                                IGM 164
   46 CONTINUE                                                          IGM 165
      NL0=IL0                                                           IGM 166
C RE-SORTS THE IPL ARRAY IN ASCENDING ORDER OF ITS ELEMENTS.            IGM 167
   50 NLT2=NL0+NL0                                                      IGM 168
      NLM1T2=NLT2-2                                                     IGM 169
      DO 54  IL1T2=2,NLM1T2,2                                           IGM 170
        ILMT2=IL1T2                                                     IGM 171
        IPM1=IPL(ILMT2-1)                                               IGM 172
        IPM2=IPL(ILMT2)                                                 IGM 173
        IL2T2M=IL1T2+2                                                  IGM 174
        DO 53  IL2T2=IL2T2M,NLT2,2                                      IGM 175
          IP21=IPL(IL2T2-1)                                             IGM 176
          IP22=IPL(IL2T2)                                               IGM 177
          IF(IPM1-IP21)   53,51,52                                      IGM 178
   51     IF(IPM2-IP22)   53,53,52                                      IGM 179
   52     ILMT2=IL2T2                                                   IGM 180
          IPM1=IP21                                                     IGM 181
          IPM2=IP22                                                     IGM 182
   53   CONTINUE                                                        IGM 183
        IPL(ILMT2-1)=IPL(IL1T2-1)                                       IGM 184
        IPL(ILMT2)  =IPL(IL1T2)                                         IGM 185
        IPL(IL1T2-1)=IPM1                                               IGM 186
        IPL(IL1T2)  =IPM2                                               IGM 187
   54 CONTINUE                                                          IGM 188
C DETERMINES TRIANGLES.                                                 IGM 189
   60 IT=0                                                              IGM 190
      NLM1=NL0-1                                                        IGM 191
      NLM2=NL0-2                                                        IGM 192
      DO 67  IL1=1,NLM2                                                 IGM 193
        IL1T2=IL1+IL1                                                   IGM 194
        IP1=IPL(IL1T2-1)                                                IGM 195
        IP2=IPL(IL1T2)                                                  IGM 196
        IL1P1=IL1+1                                                     IGM 197
        DO 66  IL2=IL1P1,NLM1                                           IGM 198
          IL2T2=IL2+IL2                                                 IGM 199
          IF(IPL(IL2T2-1).NE.IP1)   GO TO 67                            IGM 200
          IP3=IPL(IL2T2)                                                IGM 201
          IL2P1=IL2+1                                                   IGM 202
          DO 62  IL3=IL2P1,NL0                                          IGM 203
            IL3T2=IL3+IL3                                               IGM 204
            IF(IPL(IL3T2-1)-IP2)    62,61,66                            IGM 205
   61       IF(IPL(IL3T2)  -IP3)    62,63,66                            IGM 206
```

```
   62      CONTINUE                                                        IGM 207
           GO TO 66                                                        IGM 208
   63      IPT1=IP1                                                        IGM 209
           IPT2=IP2                                                        IGM 210
           IPT3=IP3                                                        IGM 211
           IF((YD(IPT3)-YD(IPT1))*(XD(IPT2)-XD(IPT1))-                     IGM 212
          1    (XD(IPT3)-XD(IPT1))*(YD(IPT2)-YD(IPT1)).GE.0.0)            IGM 213
          2                                                 GO TO 64      IGM 214
           ITS=IPT2                                                        IGM 215
           IPT2=IPT3                                                       IGM 216
           IPT3=ITS                                                        IGM 217
   64      X1=XD(IPT1)                                                     IGM 218
           X2=XD(IPT2)                                                     IGM 219
           X3=XD(IPT3)                                                     IGM 220
           Y1=YD(IPT1)                                                     IGM 221
           Y2=YD(IPT2)                                                     IGM 222
           Y3=YD(IPT3)                                                     IGM 223
           DX32=X3-X2                                                      IGM 224
           DX21=X2-X1                                                      IGM 225
           DX13=X1-X3                                                      IGM 226
           DY32=Y3-Y2                                                      IGM 227
           DY21=Y2-Y1                                                      IGM 228
           DY13=Y1-Y3                                                      IGM 229
           DO 65  IP0=1,NDP0                                               IGM 230
             IF(IP0.EQ.IPT1.OR.IP0.EQ.IPT2.OR.IP0.EQ.IPT3)                 IGM 231
          1                                                 GO TO 65      IGM 232
             X0=XD(IP0)                                                    IGM 233
             Y0=YD(IP0)                                                    IGM 234
             IF((Y0-Y1)*DX21-(X0-X1)*DY21.LT.0.0)   GO TO 65               IGM 235
             IF((Y0-Y2)*DX32-(X0-X2)*DY32.LT.0.0)   GO TO 65               IGM 236
             IF((Y0-Y3)*DX13-(X0-X3)*DY13.GE.0.0)   GO TO 66               IGM 237
   65      CONTINUE                                                        IGM 238
           IT=IT+1                                                         IGM 239
           ITT3=IT*3                                                       IGM 240
           IPT(ITT3-2)=IPT1                                                IGM 241
           IPT(ITT3-1)=IPT2                                                IGM 242
           IPT(ITT3)  =IPT3                                                IGM 243
   66    CONTINUE                                                          IGM 244
   67 CONTINUE                                                           IGM 245
      NT0=IT                                                             IGM 246
      NT=NT0                                                             IGM 247
C SELECTS AND SORTS LINE SEGMENTS THAT FORM THE BORDER.                   IGM 248
   70 IL0=0                                                              IGM 249
      DO 75  IL1=1,NL0                                                   IGM 250
        IL1T2=IL1+IL1                                                    IGM 251
        IP1=IPL(IL1T2-1)                                                 IGM 252
        IP2=IPL(IL1T2)                                                   IGM 253
        X1=XD(IP1)                                                       IGM 254
        Y1=YD(IP1)                                                       IGM 255
        X2=XD(IP2)                                                       IGM 256
        Y2=YD(IP2)                                                       IGM 257
        DX21=X2-X1                                                       IGM 258
        DY21=Y2-Y1                                                       IGM 259
        DO 71  IP0=1,NDP0                                                IGM 260
          IF(IP0.EQ.IP1.OR.IP0.EQ.IP2)   GO TO 71                        IGM 261
          S=(YD(IP0)-Y1)*DX21-(XD(IP0)-X1)*DY21                          IGM 262
          IF(S.NE.0.0)    GO TO 72                                       IGM 263
   71   CONTINUE                                                         IGM 264
   72   IP0MN=IP0+1                                                      IGM 265
        DO 73  IP0=IP0MN,NDP0                                            IGM 266
          IF(IP0.EQ.IP1.OR.IP0.EQ.IP2)   GO TO 73                        IGM 267
          IF(((YD(IP0)-Y1)*DX21-(XD(IP0)-X1)*DY21)*S.LT.0.0)             IGM 268
         1                                                  GO TO 75     IGM 269
   73   CONTINUE                                                         IGM 270
```

```
      ILO=ILO+1                                                          IGM 271
      ILOT2=ILO+ILO                                                      IGM 272
      IF(S.LT.0.0) GO TO 74                                              IGM 273
      IPL(ILOT2-1)=IP1                                                   IGM 274
      IPL(ILOT2)  =IP2                                                   IGM 275
      GO TO 75                                                           IGM 276
   74 IPL(ILOT2-1)=IP2                                                   IGM 277
      IPL(ILOT2)  =IP1                                                   IGM 278
   75 CONTINUE                                                           IGM 279
      NLO=ILO                                                            IGM 280
      NLM1=NLO-1                                                         IGM 281
      DO 79  IL1=2,NLM1                                                  IGM 282
        IL1T2=IL1+IL1                                                    IGM 283
        IP2=IPL(IL1T2-2)                                                 IGM 284
        IF(IPL(IL1T2-1).EQ.IP2)      GO TO 79                            IGM 285
        IL1P1=IL1+1                                                      IGM 286
        DO 77  IL2=IL1P1,NLO                                             IGM 287
          IL2T2=IL2+IL2                                                  IGM 288
          IF(IPL(IL2T2-1).EQ.IP2)    GO TO 78                            IGM 289
   77   CONTINUE                                                         IGM 290
   78   IP1=IPL(IL1T2-1)                                                 IGM 291
        IP2=IPL(IL1T2)                                                   IGM 292
        IPL(IL1T2-1)=IPL(IL2T2-1)                                        IGM 293
        IPL(IL1T2)  =IPL(IL2T2)                                          IGM 294
        IPL(IL2T2-1)=IP1                                                 IGM 295
        IPL(IL2T2)  =IP2                                                 IGM 296
   79 CONTINUE                                                           IGM 297
      NL=NLO                                                             IGM 298
C FINDS OUT TRIANGLES CORRESPONDING TO THE BORDER LINE                   IGM 299
C SEGMENTS.                                                              IGM 300
   80 NLP1=NLO+1                                                         IGM 301
      DO 83  ILR=1,NLO                                                   IGM 302
        IL=NLP1-ILR                                                      IGM 303
        ILT2=IL+IL                                                       IGM 304
        ILT3=ILT2+IL                                                     IGM 305
        IPL1=IPL(ILT2-1)                                                 IGM 306
        IPL2=IPL(ILT2)                                                   IGM 307
        DO 81  IT=1,NTO                                                  IGM 308
          ITT3=IT*3                                                      IGM 309
          IPT1=IPT(ITT3-2)                                               IGM 310
          IPT2=IPT(ITT3-1)                                               IGM 311
          IPT3=IPT(ITT3)                                                 IGM 312
          IF(IPL1.NE.IPT1.AND.IPL1.NE.IPT2.AND.IPL1.NE.IPT3)             IGM 313
     1                                                  GO TO 81         IGM 314
          IF(IPL2.EQ.IPT1.OR.IPL2.EQ.IPT2.OR.IPL2.EQ.IPT3)               IGM 315
     1                                                  GO TO 82         IGM 316
   81   CONTINUE                                                         IGM 317
   82   IPL(ILT3-2)=IPL1                                                 IGM 318
        IPL(ILT3-1)=IPL2                                                 IGM 319
        IPL(ILT3)  =IT                                                   IGM 320
   83 CONTINUE                                                           IGM 321
      RETURN                                                             IGM 322
      END                                                                IGM 323


      SUBROUTINE  IDLCTN(XD,YD,IPT,IPL,XII,YII,ITI)                      ILC 001
C THIS SUBROUTINE LOCATES A POINT, I.E., DETERMINES WHAT                 ILC 002
C TRIANGLE A GIVEN POINT (XII,YII) BELONGS TO.  WHEN THE GIVEN           ILC 003
C POINT DOES NOT LIE INSIDE THE DATA AREA, THIS SUBROUTINE               ILC 004
C DETERMINES THE BORDER LINE SEGMENT IN THE AREA ABOVE WHICH THE         ILC 005
C POINT LIES, OR TWO BORDER LINE SEGMENTS BETWEEN TWO AREAS              ILC 006
C ABOVE WHICH THE POINT LIES.                                            ILC 007
C THE INPUT PARAMETERS ARE                                               ILC 008
```

```
C     XD,YD = ARRAYS STORING THE X AND Y COORDINATES, RESP.,          ILC 009
C             OF DATA POINTS,                                         ILC 010
C     IPT = ARRAY STORING THE POINT NUMBERS OF THE VERTEXES           ILC 011
C           OF THE TRIANGLES,                                         ILC 012
C     IPL = ARRAY STORING THE POINT NUMBERS OF THE END                 ILC 013
C           POINTS OF THE BORDER LINE SEGMENTS AND THEIR              ILC 014
C           RESPECTIVE TRIANGLE NUMBERS,                              ILC 015
C     XII,YII = X AND Y COORDINATES, RESP., OF                        ILC 016
C               INTERPOLATED POINT.                                   ILC 017
C THE OUTPUT PARAMETER IS                                             ILC 018
C     ITI = TRIANGLE NUMBER, WHEN THE POINT IS INSIDE THE             ILC 019
C           DATA AREA, OR                                             ILC 020
C           TWO BORDER LINE SEGMENT NUMBERS, IL1 AND IL2,             ILC 021
C           CODED TO IL1*(NT+NL)+IL2, WHEN THE POINT IS               ILC 022
C           OUTSIDE THE DATA AREA, WHERE NT IS THE NUMBER OF          ILC 023
C           TRIANGLES AND NL, THAT OF BORDER LINE SEGMENTS.           ILC 024
C DECLARATION STATEMENTS                                              ILC 025
      DIMENSION   XD(10),YD(10),IPT(100),IPL(100)                     ILC 026
      COMMON/IDGM/NDP,NNP,NT,NL                                       ILC 027
      DATA  NTPV/0/,NLPV/0/                                           ILC 028
C PRELIMINARY PROCESSING                                              ILC 029
   10 NT0=NT                                                          ILC 030
      NL0=NL                                                          ILC 031
      NTL=NT0+NL0                                                     ILC 032
      X0=XII                                                          ILC 033
      Y0=YII                                                          ILC 034
C CHECK IF IN THE SAME TRIANGLE AS PREVIOUS                           ILC 035
   20 IF(NT0.NE.NTPV)      GO TO 35                                   ILC 036
      IF(NL0.NE.NLPV)      GO TO 35                                   ILC 037
      IT0=ITIPV                                                       ILC 038
      IF(IT0.GT.NT0)       GO TO 25                                   ILC 039
      IT0T3=IT0*3                                                     ILC 040
      IP1=IPT(IT0T3-2)                                                ILC 041
      IP2=IPT(IT0T3-1)                                                ILC 042
      IP3=IPT(IT0T3)                                                  ILC 043
      X1=XD(IP1)                                                      ILC 044
      X2=XD(IP2)                                                      ILC 045
      X3=XD(IP3)                                                      ILC 046
      Y1=YD(IP1)                                                      ILC 047
      Y2=YD(IP2)                                                      ILC 048
      Y3=YD(IP3)                                                      ILC 049
      IF((Y0-Y1)*(X2-X1)-(X0-X1)*(Y2-Y1))      50,21,21               ILC 050
   21 IF((Y0-Y2)*(X3-X2)-(X0-X2)*(Y3-Y2))      50,22,22               ILC 051
   22 IF((Y0-Y3)*(X1-X3)-(X0-X3)*(Y1-Y3))      50,80,80               ILC 052
C CHECK IF ON THE SAME BORDER LINE SEGMENT                            ILC 053
   25 IL1=IT0/NTL                                                     ILC 054
      IL2=IT0-IL1*NTL                                                 ILC 055
      IL1T3=IL1*3                                                     ILC 056
      IP1=IPL(IL1T3-2)                                                ILC 057
      IP2=IPL(IL1T3-1)                                                ILC 058
      X1=XD(IP1)                                                      ILC 059
      X2=XD(IP2)                                                      ILC 060
      Y1=YD(IP1)                                                      ILC 061
      Y2=YD(IP2)                                                      ILC 062
      DX02=X0-X2                                                      ILC 063
      DY02=Y0-Y2                                                      ILC 064
      DX21=X2-X1                                                      ILC 065
      DY21=Y2-Y1                                                      ILC 066
      CS0221=DX02*DX21+DY02*DY21                                      ILC 067
      IF(IL2.NE.IL1)       GO TO 30                                   ILC 068
      IF(CS0221)      26,26,50                                        ILC 069
   26 DX01=X0-X1                                                      ILC 070
      DY01=Y0-Y1                                                      ILC 071
      IF(DY01*DX21-DX01*DY21)  27,27,50                               ILC 072
   27 IF(DX01*DX21+DY01*DY21)  50,80,80                               ILC 073
```

```
C CHECK IF BETWEEN THE SAME TWO BORDER LINE SEGMENTS                    ILC 074
   30 IF(CS0221)      50,31,31                                          ILC 075
   31 IL2T3=IL2*3                                                       ILC 076
      IP3=IPL(IL2T3-1)                                                  ILC 077
      X3=XD(IP3)                                                        ILC 078
      Y3=YD(IP3)                                                        ILC 079
      DX32=X3-X2                                                        ILC 080
      DY32=Y3-Y2                                                        ILC 081
      IF(DX02*DX32+DY02*DY32)  80,80,50                                 ILC 082
C WHEN CALLED WITH A NEW SET OF NT AND NL                               ILC 083
   35 NTPV=NT0                                                          ILC 084
      NLPV=NL0                                                          ILC 085
      ITIPV=0                                                           ILC 086
C LOCATION INSIDE THE DATA AREA                                         ILC 087
   50 IT0T3=0                                                           ILC 088
      DO 69  IT0=1,NT0                                                  ILC 089
        IT0T3=IT0T3+3                                                   ILC 090
        IF(IT0.EQ.ITIPV)  GO TO 69                                      ILC 091
        IP1=IPT(IT0T3-2)                                                ILC 092
        IP2=IPT(IT0T3-1)                                                ILC 093
        IP3=IPT(IT0T3)                                                  ILC 094
        X1=XD(IP1)                                                      ILC 095
        X2=XD(IP2)                                                      ILC 096
        X3=XD(IP3)                                                      ILC 097
        IF(X0-X1)     53,55,51                                          ILC 098
   51   IF(X0-X2)     55,55,52                                          ILC 099
   52   IF(X0-X3)     55,55,69                                          ILC 100
   53   IF(X0-X2)     54,55,55                                          ILC 101
   54   IF(X0-X3)     69,55,55                                          ILC 102
   55   Y1=YD(IP1)                                                       ILC 103
        Y2=YD(IP2)                                                      ILC 104
        Y3=YD(IP3)                                                      ILC 105
        IF(Y0-Y1)     58,60,56                                          ILC 106
   56   IF(Y0-Y2)     60,60,57                                          ILC 107
   57   IF(Y0-Y3)     60,60,69                                          ILC 108
   58   IF(Y0-Y2)     59,60,60                                          ILC 109
   59   IF(Y0-Y3)     69,60,60                                          ILC 110
   60   IF((Y0-Y1)*(X2-X1)-(X0-X1)*(Y2-Y1))   69,61,61                  ILC 111
   61   IF((Y0-Y2)*(X3-X2)-(X0-X2)*(Y3-Y2))   69,62,62                  ILC 112
   62   IF((Y0-Y3)*(X1-X3)-(X0-X3)*(Y1-Y3))   69,80,80                  ILC 113
   69 CONTINUE                                                          ILC 114
C LOCATION OUTSIDE THE DATA AREA                                        ILC 115
   70 NL0T3=NL0*3                                                       ILC 116
      IP1=IPL(NL0T3-2)                                                  ILC 117
      IP2=IPL(NL0T3-1)                                                  ILC 118
      X1=XD(IP1)                                                        ILC 119
      Y1=YD(IP1)                                                        ILC 120
      X2=XD(IP2)                                                        ILC 121
      Y2=YD(IP2)                                                        ILC 122
      DX02=X0-X2                                                        ILC 123
      DY02=Y0-Y2                                                        ILC 124
      DX21=X2-X1                                                        ILC 125
      DY21=Y2-Y1                                                        ILC 126
      CS0221=DX02*DX21+DY02*DY21                                        ILC 127
      DO 74  IL0=1,NL0                                                  ILC 128
        X1=X2                                                           ILC 129
        Y1=Y2                                                           ILC 130
        DX01=DX02                                                       ILC 131
        DY01=DY02                                                       ILC 132
        IP2=IPL(3*IL0-1)                                                ILC 133
        X2=XD(IP2)                                                      ILC 134
        Y2=YD(IP2)                                                      ILC 135
        DX02=X0-X2                                                      ILC 136
        DY02=Y0-Y2                                                      ILC 137
```

```
      DX21=X2-X1                                                        ILC 138
      DY21=Y2-Y1                                                        ILC 139
      CSPV=CS0221                                                       ILC 140
      CS0221=DX02*DX21+DY02*DY21                                        ILC 141
      IF(CS0221)    71,71,74                                            ILC 142
   71 IF(DX01*DX21+DY01*DY21)      73,72,72                             ILC 143
   72 IF(DY01*DX21-DX01*DY21)      76,76,74                             ILC 144
   73 IF(CSPV)      74,74,75                                            ILC 145
   74 CONTINUE                                                          ILC 146
      ILO=1                                                             ILC 147
   75 ITO=ILO-1                                                         ILC 148
      IF(ITO.EQ.0)    ITO=NLO                                           ILC 149
      GO TO 77                                                          ILC 150
   76 ITO=ILO                                                           ILC 151
   77 ITO=ITO*NTL+ILO                                                   ILC 152
C NORMAL EXIT                                                           ILC 153
   80 ITI=ITO                                                           ILC 154
      ITIPV=ITO                                                         ILC 155
      RETURN                                                            ILC 156
      END                                                               ILC 157


      SUBROUTINE  IDPDRV(XD,YD,ZD,IPN,PD)                               IPD 001
C THIS SUBROUTINE ESTIMATES PARTIAL DERIVATIVES OF THE FIRST AND        IPD 002
C SECOND ORDER AT THE DATA POINTS.                                      IPD 003
C THE INPUT PARAMETERS ARE                                              IPD 004
C     XD,YD,ZD = ARRAYS STORING THE X, Y, AND Z COORDINATES,            IPD 005
C                RESP., OF DATA POINTS,                                 IPD 006
C     IPN = ARRAY STORING THE POINT NUMBERS OF NNP DATA                 IPD 007
C           POINTS NEAREST TO EACH OF THE DATA POINTS,                  IPD 008
C WHERE NNP IS THE NUMBER OF DATA POINTS USED FOR ESTIMATION            IPD 009
C OF PARTIAL DERIVATIVES AT EACH DATA POINT.  NNP IS GIVEN              IPD 010
C THROUGH THE IDGM COMMON.                                              IPD 011
C THE OUTPUT PARAMETER IS                                               IPD 012
C     PD  = ARRAY OF DIMENSION 5*NDP, WHERE THE ESTIMATED               IPD 013
C           ZX, ZY, ZXX, ZXY, AND ZYY VALUES AT THE DATA                IPD 014
C           POINTS ARE TO BE DISPLAYED,                                 IPD 015
C WHERE NDP IS THE TOTAL NUMBER OF DATA POINTS.  NDP IS GIVEN           IPD 016
C THROUGH THE IDGM COMMON.                                              IPD 017
C DECLARATION STATEMENTS                                                IPD 018
      DIMENSION   XD(10),YD(10),ZD(10),IPN(100),PD(50)                  IPD 019
      COMMON/IDGM/NDP,NNP,NT,NL                                         IPD 020
      REAL        NMX,NMY,NMZ,NMXX,NMXY,NMYX,NMYY                       IPD 021
C PRELIMINARY PROCESSING                                                IPD 022
   10 NDPO=NDP                                                          IPD 023
      NNPO=NNP                                                          IPD 024
      NNPM1=NNPO-1                                                      IPD 025
C ESTIMATION OF ZX AND ZY                                               IPD 026
   20 JPDO=-5                                                           IPD 027
      JIPNO=-NNPO                                                       IPD 028
      DO 24  IPO=1,NDPO                                                 IPD 029
        JPDO=JPDO+5                                                     IPD 030
        XO=XD(IPO)                                                      IPD 031
        YO=YD(IPO)                                                      IPD 032
        ZO=ZD(IPO)                                                      IPD 033
        NMX=0.0                                                         IPD 034
        NMY=0.0                                                         IPD 035
        NMZ=0.0                                                         IPD 036
        JIPNO=JIPNO+NNPO                                                IPD 037
        DO 23  IN1=1,NNPM1                                              IPD 038
          JIPN=JIPNO+IN1                                                IPD 039
          IPI=IPN(JIPN)                                                 IPD 040
          DX1=XD(IPI)-XO                                                IPD 041
```

```
            DY1=YD(IPI)-Y0                                                  IPD 042
            DZ1=ZD(IPI)-Z0                                                  IPD 043
            IN2MN=IN1+1                                                     IPD 044
            DO 22  IN2=IN2MN,NNP0                                           IPD 045
              JIPN=JIPN0+IN2                                                IPD 046
              IPI=IPN(JIPN)                                                 IPD 047
              DX2=XD(IPI)-X0                                                IPD 048
              DY2=YD(IPI)-Y0                                                IPD 049
              DZ2=ZD(IPI)-Z0                                                IPD 050
              DNMX=DY1*DZ2-DZ1*DY2                                          IPD 051
              DNMY=DZ1*DX2-DX1*DZ2                                          IPD 052
              DNMZ=DX1*DY2-DY1*DX2                                          IPD 053
              IF(DNMZ.GE.0.0)    GO TO 21                                   IPD 054
              DNMX=-DNMX                                                    IPD 055
              DNMY=-DNMY                                                    IPD 056
              DNMZ=-DNMZ                                                    IPD 057
   21         NMX=NMX+DNMX                                                  IPD 058
              NMY=NMY+DNMY                                                  IPD 059
              NMZ=NMZ+DNMZ                                                  IPD 060
   22       CONTINUE                                                        IPD 061
   23     CONTINUE                                                          IPD 062
          PD(JPD0+1)=-NMX/NMZ                                               IPD 063
          PD(JPD0+2)=-NMY/NMZ                                               IPD 064
   24 CONTINUE                                                              IPD 065
C ESTIMATION OF ZXX, ZXY, AND ZYY                                           IPD 066
   30 JPD0=-5                                                               IPD 067
      JIPN0=-NNP0                                                           IPD 068
      DO 34  IP0=1,NDP0                                                     IPD 069
        JPD0=JPD0+5                                                         IPD 070
        X0=XD(IP0)                                                          IPD 071
        Y0=YD(IP0)                                                          IPD 072
        ZX0=PD(JPD0+1)                                                      IPD 073
        ZY0=PD(JPD0+2)                                                      IPD 074
        NMXX=0.0                                                            IPD 075
        NMXY=0.0                                                            IPD 076
        NMYX=0.0                                                            IPD 077
        NMYY=0.0                                                            IPD 078
        NMZ =0.0                                                            IPD 079
        JIPN0=JIPN0+NNP0                                                    IPD 080
        DO 33  IN1=1,NNPM1                                                  IPD 081
          JIPN=JIPN0+IN1                                                    IPD 082
          IPI=IPN(JIPN)                                                     IPD 083
          DX1=XD(IPI)-X0                                                    IPD 084
          DY1=YD(IPI)-Y0                                                    IPD 085
          JPD=5*(IPI-1)                                                     IPD 086
          DZX1=PD(JPD+1)-ZX0                                                IPD 087
          DZY1=PD(JPD+2)-ZY0                                                IPD 088
          IN2MN=IN1+1                                                       IPD 089
          DO 32  IN2=IN2MN,NNP0                                             IPD 090
            JIPN=JIPN0+IN2                                                  IPD 091
            IPI=IPN(JIPN)                                                   IPD 092
            DX2=XD(IPI)-X0                                                  IPD 093
            DY2=YD(IPI)-Y0                                                  IPD 094
            JPD=5*(IPI-1)                                                   IPD 095
            DZX2=PD(JPD+1)-ZX0                                              IPD 096
            DZY2=PD(JPD+2)-ZY0                                              IPD 097
            DNMXX=DY1*DZX2-DZX1*DY2                                         IPD 098
            DNMXY=DZX1*DX2-DX1*DZX2                                         IPD 099
            DNMYX=DY1*DZY2-DZY1*DY2                                         IPD 100
            DNMYY=DZY1*DX2-DX1*DZY2                                         IPD 101
            DNMZ =DX1*DY2 -DY1*DX2                                          IPD 102
            IF(DNMZ.GE.0.0)    GO TO 31                                     IPD 103
            DNMXX=-DNMXX                                                    IPD 104
            DNMXY=-DNMXY                                                    IPD 105
```

```
            DNMYX=-DNMYX                                                    IPD 106
            DNMYY=-DNMYY                                                    IPD 107
            DNMZ =-DNMZ                                                     IPD 108
   31       NMXX=NMXX+DNMXX                                                 IPD 109
            NMXY=NMXY+DNMXY                                                 IPD 110
            NMYX=NMYX+DNMYX                                                 IPD 111
            NMYY=NMYY+DNMYY                                                 IPD 112
            NMZ =NMZ +DNMZ                                                  IPD 113
   32     CONTINUE                                                         IPD 114
   33   CONTINUE                                                           IPD 115
        PD(JPD0+3)=-NMXX/NMZ                                               IPD 116
        PD(JPD0+4)=-(NMXY+NMYX)/(2.0*NMZ)                                  IPD 117
        PD(JPD0+5)=-NMYY/NMZ                                               IPD 118
   34 CONTINUE                                                             IPD 119
      RETURN                                                               IPD 120
      END                                                                  IPD 121


      SUBROUTINE  IDPTIP(XD,YD,ZD,IPT,IPL,PDD,ITI,XII,YII,ZII)             IPI 001
C THIS SUBROUTINE PERFORMS PUNCTUAL INTERPOLATION OR EXTRAPO-              IPI 002
C LATION, I.E., DETERMINES THE Z VALUE AT A POINT.                         IPI 003
C THE INPUT PARAMETERS ARE                                                 IPI 004
C     XD,YD,ZD = ARRAYS STORING THE X, Y, AND Z COORDINATES,               IPI 005
C                RESP., OF DATA POINTS,                                    IPI 006
C     IPT = ARRAY STORING THE POINT NUMBERS OF THE VERTEXES                IPI 007
C           OF THE TRIANGLES,                                            IPI 008
C     IPL = ARRAY STORING THE POINT NUMBERS OF THE END                     IPI 009
C           POINTS OF THE BORDER LINE SEGMENTS AND THEIR                 IPI 010
C           RESPECTIVE TRIANGLE NUMBERS,                                 IPI 011
C     PDD = ARRAY STORING THE PARTIAL DERIVATIVES AT THE                   IPI 012
C           DATA POINTS,                                                 IPI 013
C     ITI = TRIANGLE NUMBER OF THE TRIANGLE IN WHICH                       IPI 014
C           THE INTERPOLATED POINT LIES,                                 IPI 015
C     XII,YII = X AND Y COORDINATES, RESP., OF                             IPI 016
C               INTERPOLATED POINT.                                        IPI 017
C THE OUTPUT PARAMETER IS                                                  IPI 018
C     ZII = INTERPOLATED Z VALUE.                                          IPI 019
C DECLARATION STATEMENTS                                                   IPI 020
      DIMENSION   XD(10),YD(10),ZD(10),IPT(100),IPL(100),PDD(50)           IPI 021
      COMMON/IDGM/NDP,NNP,NT,NL                                            IPI 022
      COMMON/IDPI/NCF,ICF                                                  IPI 023
      DIMENSION   CF0(27)                                                  IPI 024
      EQUIVALENCE (X0,CF0(1)),  (Y0,CF0(2)),  (AP,CF0(3)),                IPI 025
     1            (BP,CF0(4)),  (CP,CF0(5)),  (DP,CF0(6)),                IPI 026
     2            (P00,CF0(7)), (P10,CF0(8)), (P20,CF0(9)),               IPI 027
     3            (P30,CF0(10)),(P40,CF0(11)),(P50,CF0(12)),              IPI 028
     4            (P01,CF0(13)),(P11,CF0(14)),(P21,CF0(15)),              IPI 029
     5            (P31,CF0(16)),(P41,CF0(17)),(P02,CF0(18)),              IPI 030
     6            (P12,CF0(19)),(P22,CF0(20)),(P32,CF0(21)),              IPI 031
     7            (P03,CF0(22)),(P13,CF0(23)),(P23,CF0(24)),              IPI 032
     8            (P04,CF0(25)),(P14,CF0(26)),(P05,CF0(27))               IPI 033
      DIMENSION   CF(980)                                                  IPI 034
      DIMENSION   X(3),Y(3),Z(3),PD(15),                                   IPI 035
     1            ZU(3),ZV(3),ZUU(3),ZUV(3),ZVV(3)                        IPI 036
      EQUIVALENCE (ITO,FLITO),(ITJ,FLITJ)                                  IPI 037
      REAL        LU,LV,LUSNUV,LVSNUV                                      IPI 038
      EQUIVALENCE (P5,P05)                                                 IPI 039
      DATA  NCFMX/35/                                                      IPI 040
C SETTING OF SOME LOCAL VARIABLES.                                         IPI 041
   10 ITO=ITI                                                              IPI 042
      XIO=XII                                                              IPI 043
      YIO=YII                                                              IPI 044
      NTL=NT+NL                                                            IPI 045
```

```
C DETERMINES IF SIMPLE INTERPOLATION IS APPLICABLE.                    IPI 046
   20 IF(ITO.LE.NTL)       GO TO 30                                    IPI 047
      IL1=ITO/NTL                                                      IPI 048
      IL2=ITO-IL1*NTL                                                  IPI 049
      IL1T3=IL1*3                                                      IPI 050
      IL2T3=IL2*3                                                      IPI 051
      ITO=IPL(IL1T3)                                                   IPI 052
      IF(IL1.NE.IL2)       GO TO 40                                    IPI 053
C CALCULATION OF ZII BY SIMPLE INTERPOLATION OR EXTRAPOLATION.         IPI 054
   30 ASSIGN 31 TO LBL                                                 IPI 055
      GO TO 50                                                         IPI 056
   31 ZII=ZIO                                                          IPI 057
      RETURN                                                           IPI 058
C CALCULATION OF ZII AS A WEIGHTED MEAN OF TWO EXTRAPOLATED            IPI 059
C VALUES.                                                              IPI 060
   40 ASSIGN 41 TO LBL                                                 IPI 061
      GO TO 50                                                         IPI 062
   41 ZI1=ZIO                                                          IPI 063
      ITO=IPL(IL2T3)                                                   IPI 064
      ASSIGN 42 TO LBL                                                 IPI 065
      GO TO 50                                                         IPI 066
   42 ZI2=ZIO                                                          IPI 067
C CALCULATES THE WEIGHTING COEFFICIENTS FOR EXTRAPOLATED VALUES.       IPI 068
   45 IP1=IPL(IL1T3-2)                                                 IPI 069
      IP2=IPL(IL1T3-1)                                                 IPI 070
      IP3=IPL(IL2T3-1)                                                 IPI 071
      X1=XD(IP1)                                                       IPI 072
      Y1=YD(IP1)                                                       IPI 073
      X2=XD(IP2)                                                       IPI 074
      Y2=YD(IP2)                                                       IPI 075
      X3=XD(IP3)                                                       IPI 076
      Y3=YD(IP3)                                                       IPI 077
      DX02=XIO-X2                                                      IPI 078
      DY02=YIO-Y2                                                      IPI 079
      DX32=X3-X2                                                       IPI 080
      DY32=Y3-Y2                                                       IPI 081
      DX21=X2-X1                                                       IPI 082
      DY21=Y2-Y1                                                       IPI 083
      W1=(DX02*DX32+DY02*DY32)**2/(DX32*DX32+DY32*DY32)                IPI 084
      W2=(DX02*DX21+DY02*DY21)**2/(DX21*DX21+DY21*DY21)                IPI 085
C CALCULATES ZII AS A WEIGHTED MEAN.                                   IPI 086
   46 ZII=(W1*ZI1+W2*ZI2)/(W1+W2)                                      IPI 087
      RETURN                                                           IPI 088
C INTERNAL ROUTINE FOR PUNCTUAL INTERPOLATION.                         IPI 089
C CHECKS IF THE NECESSARY CFO VALUES ARE SAVED.                        IPI 090
   50 IF(NCF.EQ.0)         GO TO 60                                    IPI 091
      JCF=-27                                                          IPI 092
      DO 51  LCF=1,NCF                                                 IPI 093
        JCF=JCF+28                                                     IPI 094
        FLITJ=CF(JCF)                                                  IPI 095
        IF(ITO.EQ.ITJ)     GO TO 70                                    IPI 096
   51 CONTINUE                                                         IPI 097
C CALCULATION OF NEW CFO VALUES.                                       IPI 098
C DETERMINES THE COEFFICIENTS FOR THE COORDINATE SYSTEM TRANS-         IPI 099
C FORMATION FROM THE X-Y SYSTEM TO THE U-V SYSTEM, AND CALCU-          IPI 100
C LATES THE COEFFICIENTS OF THE POLYNOMIAL FOR INTERPOLATION.          IPI 101
C LOADS COORDINATE AND PARTIAL DERIVATIVE VALUES AT THE                IPI 102
C VERTEXES.                                                            IPI 103
   60 JIPT=3*(ITO-1)                                                   IPI 104
      JPD=0                                                            IPI 105
      DO 62  I=1,3                                                     IPI 106
        JIPT=JIPT+1                                                    IPI 107
        IDP=IPT(JIPT)                                                  IPI 108
        X(I)=XD(IDP)                                                   IPI 109
        Y(I)=YD(IDP)                                                   IPI 110
```

```
      Z(I)=ZD(IDP)                                                      IPI 111
      JPDD=5*(IDP-1)                                                    IPI 112
      DO 61  KPD=1,5                                                    IPI 113
        JPD=JPD+1                                                       IPI 114
        JPDD=JPDD+1                                                     IPI 115
        PD(JPD)=PDD(JPDD)                                               IPI 116
   61   CONTINUE                                                        IPI 117
   62 CONTINUE                                                          IPI 118
C DETERMINING THE COEFFICIENTS FOR THE COORDINATE SYSTEM                IPI 119
C TRANSFORMATION FROM THE X-Y SYSTEM TO THE U-V SYSTEM                  IPI 120
C AND VICE VERSA                                                        IPI 121
   63 X0=X(1)                                                           IPI 122
      Y0=Y(1)                                                           IPI 123
      A=X(2)-X0                                                         IPI 124
      B=X(3)-X0                                                         IPI 125
      C=Y(2)-Y0                                                         IPI 126
      D=Y(3)-Y0                                                         IPI 127
      AD=A*D                                                            IPI 128
      BC=B*C                                                            IPI 129
      DLT=AD-BC                                                         IPI 130
      AP= D/DLT                                                         IPI 131
      BP=-B/DLT                                                         IPI 132
      CP=-C/DLT                                                         IPI 133
      DP= A/DLT                                                         IPI 134
C CONVERSION OF THE PARTIAL DERIVATIVES AT THE VERTEXES OF THE          IPI 135
C TRIANGLE FOR THE U-V COORDINATE SYSTEM                                IPI 136
   64 AA=A*A                                                            IPI 137
      ACT2=2.0*A*C                                                      IPI 138
      CC=C*C                                                            IPI 139
      AB=A*B                                                            IPI 140
      ADBC=AD+BC                                                        IPI 141
      CD=C*D                                                            IPI 142
      BB=B*B                                                            IPI 143
      BDT2=2.0*B*D                                                      IPI 144
      DD=D*D                                                            IPI 145
      DO 65  I=1,3                                                      IPI 146
        JPD=5*I                                                         IPI 147
        ZU(I)=A*PD(JPD-4)+C*PD(JPD-3)                                   IPI 148
        ZV(I)=B*PD(JPD-4)+D*PD(JPD-3)                                   IPI 149
        ZUU(I)=AA*PD(JPD-2)+ACT2*PD(JPD-1)+CC*PD(JPD)                   IPI 150
        ZUV(I)=AB*PD(JPD-2)+ADBC*PD(JPD-1)+CD*PD(JPD)                   IPI 151
        ZVV(I)=BB*PD(JPD-2)+BDT2*PD(JPD-1)+DD*PD(JPD)                   IPI 152
   65 CONTINUE                                                          IPI 153
C CALCULATION OF THE COEFFICIENTS OF THE POLYNOMIAL                     IPI 154
   66 P00=Z(1)                                                          IPI 155
      P10=ZU(1)                                                         IPI 156
      P01=ZV(1)                                                         IPI 157
      P20=0.5*ZUU(1)                                                    IPI 158
      P11=ZUV(1)                                                        IPI 159
      P02=0.5*ZVV(1)                                                    IPI 160
      H1=Z(2)-P00-P10-P20                                               IPI 161
      H2=ZU(2)-P10-ZUU(1)                                               IPI 162
      H3=ZUU(2)-ZUU(1)                                                  IPI 163
      P30= 10.0*H1-4.0*H2+0.5*H3                                        IPI 164
      P40=-15.0*H1+7.0*H2     -H3                                       IPI 165
      P50=  6.0*H1-3.0*H2+0.5*H3                                        IPI 166
      H1=Z(3)-P00-P01-P02                                               IPI 167
      H2=ZV(3)-P01-ZVV(1)                                               IPI 168
      H3=ZVV(3)-ZVV(1)                                                  IPI 169
      P03= 10.0*H1-4.0*H2+0.5*H3                                        IPI 170
      P04=-15.0*H1+7.0*H2     -H3                                       IPI 171
      P05=  6.0*H1-3.0*H2+0.5*H3                                        IPI 172
      LU=SQRT(AA+CC)                                                    IPI 173
      LV=SQRT(BB+DD)                                                    IPI 174
      THXU=ATAN2(C,A)                                                   IPI 175
```

```
      THUV=ATAN2(D,B)-THXU                                              IPI 176
      CSUV=COS(THUV)                                                    IPI 177
      P41=5.0*LV*CSUV/LU*P50                                            IPI 178
      P14=5.0*LU*CSUV/LV*P05                                            IPI 179
      H1=ZV(2)-P01-P11-P41                                              IPI 180
      H2=ZUV(2)-P11-4.0*P41                                             IPI 181
      P21= 3.0*H1-H2                                                    IPI 182
      P31=-2.0*H1+H2                                                    IPI 183
      H1=ZU(3)-P10-P11-P14                                              IPI 184
      H2=ZUV(3)-P11-4.0*P14                                             IPI 185
      P12= 3.0*H1-H2                                                    IPI 186
      P13=-2.0*H1+H2                                                    IPI 187
      THUS=ATAN2(D*C,B-A)-THXU                                          IPI 188
      THSV=THUV-THUS                                                    IPI 189
      SNUV=SIN(THUV)                                                    IPI 190
      LUSNUV=LU*SNUV                                                    IPI 191
      LVSNUV=LV*SNUV                                                    IPI 192
      AA= SIN(THSV)/LUSNUV                                              IPI 193
      BB=-COS(THSV)/LUSNUV                                              IPI 194
      CC= SIN(THUS)/LVSNUV                                              IPI 195
      DD= COS(THUS)/LVSNUV                                              IPI 196
      AC=AA*CC                                                          IPI 197
      AD=AA*DD                                                          IPI 198
      BC=BB*CC                                                          IPI 199
      G1=AA*AC*(3.0*BC+2.0*AD)                                          IPI 200
      G2=CC*AC*(3.0*AD+2.0*BC)                                          IPI 201
      H1=-AA*AA*AA*(5.0*AA*BB*P50+(4.0*BC+AD)*P41)                      IPI 202
     1   -CC*CC*CC*(5.0*CC*DD*P05+(4.0*AD+BC)*P14)                      IPI 203
      H2=0.5*ZVV(2)-P02-P12                                             IPI 204
      H3=0.5*ZUU(3)-P20-P21                                             IPI 205
      P22=(G1*H2+G2*H3-H1)/(G1+G2)                                      IPI 206
      P32=H2-P22                                                        IPI 207
      P23=H3-P22                                                        IPI 208
C SAVES THE CF0 VALUES IN THE CF ARRAY.                                 IPI 209
   67 IF(NCF.LT.NCFMX)     NCF=NCF+1                                    IPI 210
      ICF=ICF+1                                                         IPI 211
      IF(ICF.GT.NCFMX)     ICF=1                                        IPI 212
      JCF=28*ICF-27                                                     IPI 213
      CF(JCF)=FLIT0                                                     IPI 214
      DO 68  KCF=1,27                                                   IPI 215
        JCF=JCF+1                                                       IPI 216
        CF(JCF)=CF0(KCF)                                                IPI 217
   68 CONTINUE                                                          IPI 218
      GO TO 80                                                          IPI 219
C LOADS THE CF0 VALUES FROM THE CF ARRAY.                               IPI 220
   70 DO 71  KCF=1,27                                                   IPI 221
        JCF=JCF+1                                                       IPI 222
        CF0(KCF)=CF(JCF)                                                IPI 223
   71 CONTINUE                                                          IPI 224
C TRANSFORMATION OF THE COORDINATE SYSTEM FROM X-Y TO U-V                IPI 225
   80 DX=XII-X0                                                         IPI 226
      DY=YII-Y0                                                         IPI 227
      U=AP*DX+BP*DY                                                     IPI 228
      V=CP*DX+DP*DY                                                     IPI 229
C EVALUATION OF THE POLYNOMIAL                                          IPI 230
   85 P0=P00+U*(P10+U*(P20+U*(P30+U*(P40+U*P50))))                      IPI 231
      P1=P01+U*(P11+U*(P21+U*(P31+U*P41)))                              IPI 232
      P2=P02+U*(P12+U*(P22+U*P32))                                      IPI 233
      P3=P03+U*(P13+U*P23)                                              IPI 234
      P4=P04+U*P14                                                      IPI 235
      ZI0=P0+V*(P1+V*(P2+V*(P3+V*(P4+V*P5))))                           IPI 236
      GO TO LBL, (31,41,42)                                             IPI 237
      END                                                               IPI 238
```

```
      SUBROUTINE  IDSFFT(MD,NDP,XD,YD,ZD,WK,NXI,NYI,XI,YI,ZI)          ISF 001
C THIS SUBROUTINE PERFORMS SMOOTH SURFACE FITTING WHEN THE PRO-        ISF 002
C JECTIONS OF THE DATA POINTS IN THE X-Y PLANE ARE IRREGULARLY         ISF 003
C DISTRIBUTED IN THE PLANE.                                            ISF 004
C THE INPUT PARAMETERS ARE                                             ISF 005
C      MD  = MODE OF COMPUTATION (MUST BE 1, 2, OR 3),                 ISF 006
C          = 1 FOR NEW XD-YD,                                          ISF 007
C          = 2 FOR OLD XD-YD, NEW XI-YI,                               ISF 008
C          = 3 FOR OLD XD-YD, OLD XI-YI,                               ISF 009
C      NDP = NUMBER OF DATA POINTS (MUST BE 4 OR GREATER),             ISF 010
C      XD  = ARRAY OF DIMENSION NDP STORING THE X COORDINATES          ISF 011
C            OF THE DATA POINTS,                                       ISF 012
C      YD  = ARRAY OF DIMENSION NDP STORING THE Y COORDINATES          ISF 013
C            OF THE DATA POINTS,                                       ISF 014
C      ZD  = ARRAY OF DIMENSION NDP STORING THE Z COORDINATES          ISF 015
C            OF THE DATA POINTS,                                       ISF 016
C      WK  = ARRAY OF DIMENSION (2*NDP+NNP+5)*NDP+NXI*NYI              ISF 017
C            TO BE USED AS A WORK AREA,                                ISF 018
C      NXI = NUMBER OF OUTPUT GRID POINTS IN THE X COORDINATE          ISF 019
C            (MUST BE 1 OR GREATER),                                   ISF 020
C      NYI = NUMBER OF OUTPUT GRID POINTS IN THE Y COORDINATE          ISF 021
C            (MUST BE 1 OR GREATER),                                   ISF 022
C      XI  = ARRAY OF DIMENSION NXI STORING THE X COORDINATES          ISF 023
C            OF THE OUTPUT GRID POINTS,                                ISF 024
C      YI  = ARRAY OF DIMENSION NYI STORING THE Y COORDINATES          ISF 025
C            OF THE OUTPUT GRID POINTS,                                ISF 026
C WHERE NNP IS THE NUMBER OF ADDITIONAL DATA POINTS USED FOR            ISF 027
C ESTIMATING PARTIAL DERIVATIVES AT EACH DATA POINT.  THE VALUE         ISF 028
C OF NNP MUST BE GIVEN THROUGH THE IDNN COMMON.  NNP MUST BE 2          ISF 029
C OR GREATER, BUT SMALLER THAN NDP.                                     ISF 030
C THE OUTPUT PARAMETER IS                                               ISF 031
C      ZI  = DOUBLY-DIMENSIONED ARRAY OF DIMENSION (NXI,NYI),          ISF 032
C            WHERE THE INTERPOLATED Z VALUES AT THE OUTPUT             ISF 033
C            GRID POINTS ARE TO BE DISPLAYED.                          ISF 034
C THE LUN CONSTANT IN THE DATA INITIALIZATION STATEMENT IS THE          ISF 035
C LOGICAL UNIT NUMBER OF THE STANDARD OUTPUT UNIT AND IS,               ISF 036
C THEREFORE, SYSTEM DEPENDENT.                                          ISF 037
C DECLARATION STATEMENTS                                                ISF 038
      DIMENSION   XD(10),YD(10),ZD(10),WK(1000),                        ISF 039
     1            XI(10),YI(10),ZI(100)                                 ISF 040
      COMMON/IDNN/NNP                                                   ISF 041
      COMMON/IDGM/NDPC,NNPC,NT,NL                                       ISF 042
      COMMON/IDPI/NCF,ICF                                               ISF 043
      EQUIVALENCE (FNDPO,NDPO),(FNDPPV,NDPPV),                          ISF 044
     1            (FNNPO,NNPO),(FNNPPV,NNPPV),                          ISF 045
     2            (FNXIO,NXIO),(FNXIPV,NXIPV),                          ISF 046
     3            (FNYIO,NYIO),(FNYIPV,NYIPV),                          ISF 047
     4            (FNT,NT),(FNL,NL)                                     ISF 048
      DATA  LUN/6/                                                      ISF 049
C SETTING OF SOME INPUT PARAMETERS TO LOCAL VARIABLES.  (ALL MD)        ISF 050
   10 MDO=MD                                                            ISF 051
      NDPO=NDP                                                          ISF 052
      NDPC=NDPO                                                         ISF 053
      NXIO=NXI                                                          ISF 054
      NYIO=NYI                                                          ISF 055
      NNPO=NNP                                                          ISF 056
      NNPC=NNPO                                                         ISF 057
C ERROR CHECK.  (ALL MD)                                                ISF 058
   20 IF(MDO.LT.1.OR.MDO.GT.3)           GO TO 90                     ISF 059
      IF(NDPO.LT.4)        GO TO 90                                     ISF 060
      IF(NXIO.LT.1.OR.NYIO.LT.1)    GO TO 90                          ISF 061
      IF(NNPO.LT.2.OR.NNPO.GE.NDPO) GO TO 90                          ISF 062
      IF(MDO.NE.1)         GO TO 22                                   ISF 063
   21 WK(1)=FNDPO                                                       ISF 064
      WK(2)=FNNPO                                                       ISF 065
```

```
      GO TO 24                                                          ISF 066
   22 FNDPPV=WK(1)                                                      ISF 067
      FNNPPV=WK(2)                                                      ISF 068
      IF(NDP0.NE.NDPPV)   GO TO 90                                      ISF 069
      IF(NNP0.NE.NNPPV)   GO TO 90                                      ISF 070
      IF(MD0.NE.3)        GO TO 24                                      ISF 071
   23 FNXIPV=WK(3)                                                      ISF 072
      FNYIPV=WK(4)                                                      ISF 073
      IF(NXI0.NE.NXIPV)   GO TO 90                                      ISF 074
      IF(NYI0.NE.NYIPV)   GO TO 90                                      ISF 075
      GO TO 30                                                          ISF 076
   24 WK(3)=FNXI0                                                       ISF 077
      WK(4)=FNYI0                                                       ISF 078
C ALLOCATION OF STORAGE AREAS IN THE WK ARRAY.  (ALL MD)                ISF 079
   30 NDNDM1=NDP0*(NDP0-1)                                              ISF 080
      IWIPT=7                                                           ISF 081
      IWIPL=IWIPT+NDNDM1                                                ISF 082
      IWIPN=IWIPL+NDNDM1                                                ISF 083
      IWPD =IWIPN+NDP0*NNP0                                             ISF 084
      IWIT =IWPD +NDP0*5                                                ISF 085
C DIVIDES THE X-Y PLANE INTO A NUMBER OF TRIANGLES AND                  ISF 086
C DETERMINES NNP POINTS NEAREST EACH DATA POINT.  (MD=1)                ISF 087
   40 IF(MD.GT.1)     GO TO 42                                          ISF 088
   41 CALL IDGEOM(XD,YD,WK(IWIPT),WK(IWIPL),WK(IWIPN))                 ISF 089
      WK(5)=FNT                                                         ISF 090
      WK(6)=FNL                                                         ISF 091
      GO TO 50                                                          ISF 092
   42 FNT=WK(5)                                                         ISF 093
      FNL=WK(6)                                                         ISF 094
C ESTIMATES PARTIAL DERIVATIVES AT ALL DATA POINTS.  (ALL MD)           ISF 095
   50 CALL IDPDRV(XD,YD,ZD,WK(IWIPN),WK(IWPD))                          ISF 096
C LOCATES ALL INTERPOLATED POINTS.  (MD=1,2)                            ISF 097
   60 IF(MD0.EQ.3)    GO TO 70                                          ISF 098
      IXI=0                                                             ISF 099
      JWIT=IWIT-1                                                       ISF 100
      INC=-1                                                            ISF 101
      DO 62  IYI=1,NYI0                                                 ISF 102
        INC=-INC                                                        ISF 103
        YII=YI(IYI)                                                     ISF 104
        DO 61  IXI0=1,NXI0                                              ISF 105
          IXI=IXI+INC                                                   ISF 106
          JWIT=JWIT+INC                                                 ISF 107
          CALL IDLCTN(XD,YD,WK(IWIPT),WK(IWIPL),                        ISF 108
     1                XI(IXI),YII,WK(JWIT))                             ISF 109
   61   CONTINUE                                                        ISF 110
        IXI=IXI+INC                                                     ISF 111
        JWIT=JWIT+INC+NXI0                                              ISF 112
   62 CONTINUE                                                          ISF 113
C INTERPOLATION OF THE ZI VALUES.  (ALL MD)                             ISF 114
   70 NCF=0                                                             ISF 115
      ICF=0                                                             ISF 116
      JWIT=IWIT-1                                                       ISF 117
      IXI=0                                                             ISF 118
      IZI=0                                                             ISF 119
      INC=-1                                                            ISF 120
      DO 72  IYI=1,NYI0                                                 ISF 121
        INC=-INC                                                        ISF 122
        YII=YI(IYI)                                                     ISF 123
        DO 71  IXI0=1,NXI0                                              ISF 124
          JWIT=JWIT+INC                                                 ISF 125
          IXI=IXI+INC                                                   ISF 126
          IZI=IZI+INC                                                   ISF 127
          CALL IDPTIP(XD,YD,ZD,WK(IWIPT),WK(IWIPL),WK(IWPD),            ISF 128
     1                WK(JWIT),XI(IXI),YII,ZI(IZI))                     ISF 129
   71   CONTINUE                                                        ISF 130
```

```
        JWIT=JWIT+INC+NXIO                                              ISF 131
        IXI=IXI+INC                                                     ISF 132
        IZI=IZI+INC+NXIO                                                ISF 133
   72 CONTINUE                                                          ISF 134
C NORMAL EXIT                                                           ISF 135
   80 RETURN                                                            ISF 136
C ERROR EXIT                                                            ISF 137
   90 WRITE (LUN,2090) MDO,NDPO,NXIO,NYIO,NNPO                          ISF 138
      RETURN                                                            ISF 139
C FORMAT STATEMENT FOR ERROR MESSAGE                                    ISF 140
 2090 FORMAT(1X/41H ***   IMPROPER INPUT PARAMETER VALUE(S)./           ISF 141
     1   7H    MD =,I4,10X,5HNDP =,I6,10X,5HNXI =,I6,                   ISF 142
     2   10X,5HNYI =,I6,10X,5HNNP =,I6/                                ISF 143
     3   35H ERROR DETECTED IN ROUTINE   IDSFFT/)                      ISF 144
      END                                                               ISF 145


      BLOCK DATA                                                        IBD 001
C THIS SUBPROGRAM ENTERS A NUMBER INTO THE NNP CONSTANT IN THE          IBD 002
C IDNN COMMON, WHERE NNP IS THE NUMBER OF ADDITIONAL DATA POINTS        IBD 003
C USED FOR ESTIMATING PARTIAL DERIVATIVES AT EACH DATA POINT IN         IBD 004
C THE IDBVIP/IDSFFT SUBPROGRAM PACKAGE.  NNP IS SET TO 4                IBD 005
C INITIALLY BY THIS SUBPROGRAM.                                         IBD 006
      COMMON/IDNN/NNP                                                   IBD 007
      DATA  NNP/4/                                                      IBD 008
      END                                                               IBD 009
```

REFERENCES

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

Akima, Hiroshi (1974 a), A method of bivariate interpolation and smooth surface fitting based on local procedures, Commun. ACM, 17, 18-20.

Akima, Hiroshi (1974 b), Algorithm 474, Bivariate interpolation and smooth surface fitting based on local procedures, Commun. ACM, 17, 26-31.

American National Standards Institute (ANSI) (1966), ANSI Standard Fortran, Publication X3.9-1966, ANSI, New York. Also reproduced in W. P. Heising, History and summary of FORTRAN standardization development for the ASA, Commun. ACM, 7, 590-625, Oct. 1964.

Bengtsson, Bengt-Erik, and Stig Nordbeck (1964), Construction of isarithms and isarithmic maps by computers, BIT, 4, 87-105.

Shepard, Donald (1968), A two-dimensional interpolation function for irregularly-spaced data, Proc. 1968 ACM National Conference, 517-524.

Zenisek, Alexander (1970), Interpolation polynomials on the triangle, Numerische Math., 15, 283-296.

Zlamal, Milos (1968), On the finite element method, Numerische Math., 12, 394-409.

FORM OT-29
(3-73)

U.S. DEPARTMENT OF COMMERCE
OFFICE OF TELECOMMUNICATIONS

BIBLIOGRAPHIC DATA SHEET

	1. PUBLICATION OR REPORT NO. OTR 75-70	2. Gov't Accession No.
		3. Recipient's Accession No.

4. TITLE AND SUBTITLE
A Method of Bivariate Interpolation and Smooth Surface Fitting for Values Given at Irregularly Distributed Points

5. Publication Date
August 1975

6. Performing Organization Code
OT/ITS

7. AUTHOR(S)
Hiroshi Akima

9. Project/Task/Work Unit No.

8. PERFORMING ORGANIZATION NAME AND ADDRESS
U. S. Department of Commerce
Office of Telecommunications
Institute for Telecommunication Sciences
325 Broadway, Boulder, Colorado 80302

10. Contract/Grant No.

11. Sponsoring Organization Name and Address

12. Type of Report and Period Covered

13.

14. SUPPLEMENTARY NOTES

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

A method of bivariate interpolation and smooth surface fitting is developed for z values given at points irregularly distributed in the x-y plane. The interpolating function is a fifth-degree polynomial in x and y defined in each triangular cell which has projections of three data points in the x-y plane as its vertexes. Each polynomial is determined by the given values of z and estimated values of partial derivatives at the vertexes of the triangle. Procedures for dividing the x-y plane into a number of triangles, for estimating partial derivatives at each data point, and for determining the polynomial in each triangle are described. A simple example of the application of the proposed method is shown. User information and Fortran listings are given on a computer subprogram package that implements the proposed method.

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

bivariate interpolation; interpolation; partial derivative; polynomial; smooth surface fitting.

17. AVAILABILITY STATEMENT

☒ UNLIMITED.

☐ FOR OFFICIAL DISTRIBUTION.

18. Security Class *(This report)*
Unclassified

20. Number of pages
58

19. Security Class *(This page)*
Unclassified

21. Price:

☆U.S. GOVERNMENT PRINTING OFFICE: 1975—681-352/38 REGION NO. 8