GEOPHYSICS, VOL. 39, NO. 1 (FEBRUARY 1974), P. 39–48, 3 FIGS., 3 TABLES

MACHINE CONTOURING USING MINIMUM CURVATURE†

IAN C. BRIGGS*

Machine contouring must not introduce information which is not present in the data. The one-dimensional spline fit has well defined smoothness properties. These are duplicated for two-dimensional interpolation in this paper, by solving the corresponding differential equation. Finite difference equations are deduced from a principle of minimum total curvature, and an iterative method of solution is outlined. Observations do not have to lie on a regular grid. Gravity and aeromagnetic surveys provide examples which compare favorably with the work of draftsmen.

INTRODUCTION

Contour maps are useful in the evaluation and interpretation of geophysical data. With the rapid increase in the rate of acquisition of data, a computer is an attractive means of producing contour maps.

Although errors occur in most geophysical observations, contour maps are usually drawn so that the imaginary surface on which the contours lie passes exactly through the observations. The problem of interpolation is then either: (a) to define a continuous function of the two space variables, which takes the values of the observations at the required, perhaps random, positions; or (b) to define a set of values at the points of a regular grid, so that a grid point value tends to an observational value if the position of the observation tends to the grid point. A solution to (a) gives a solution to (b), but a solution to (b) may not give a solution to (a). The solution to (b) is the one most commonly used as an input to a program which draws contour lines.

Methods for the production of contour maps have been published by Crain & Bhattacharyya (1967), Smith (1968), Cole (1968), Pelto et al (1968), and McIntyre et al (1968). These methods are variations of either weighting or function fitting or both, and give a solution to problem (a) and, hence, (b). Crain (1970) has provided a review of these methods.

This article describes a method for finding a solution to problem (b) without first finding a solution to problem (a). The solution also happens to be the smoothest. This attribute gives confidence in the use of the method and explains the quality of the resulting contour maps.

The problem of interpolation in one dimension has led to the piecewise polynomial fit, or spline (Ahlberg et al, 1967). A continuous function is found for all values of the independent variable. This method has been extended to two dimensions (De Boor, 1962), and used by Bhattachryya (1969) to give a solution to problem (a).

However, if the observation points in two dimensions are randomly situated, the fitting of piecewise two-dimensional polynomials to polygons seems difficult, although it is possible if the set of polygons are topologically equivalent to a rectangular grid (Hessing et al, 1972).

The optimum properties of the spline fit can be obtained in both one and two dimensions by solving the differential equation equivalent to a third-order spline. This is the equation which describes the displacement of a thin sheet in one or two dimensions under the influence of point forces. The 'boundary conditions' are not only at the ends or boundary, but within the region of interest. The solution is forced to take up the value of the observation at the point of observation, in one or two dimensions. The equation is

† Manuscript received by the Editor February 21, 1973; revised manuscript received September 10, 1973.

* Bureau of Mineral Resources, Canberra, A.C.T., Australia.

solved numerically, and thus gives a solution to problem (b).

The smoothness properties follow from the method of deducing the difference equations, and the quality of the resulting contour map is thus determined. The solution of the set of difference equations is a time-consuming process, but the iteration times on the computer have been reduced and can be reduced still further.

THE METHOD

The differential equation

The thin metallic strip or sheet is bent by forces acting at points so that the displacement at these points is equal to the observation to be satisfied. Let u be the displacement, x, y the space variables, and let forces f_n act at (x_n, y_n), $n=1, \cdots, N$, where the observations are w_n, then (Love, 1926)

$$\begin{aligned} \frac{d^4u}{dx^4} &= f_n,\ x = x_n, \\ &= 0 \text{ otherwise,} \end{aligned} \tag{1}$$

in one dimension and

$$\begin{aligned} \frac{\partial^4 u}{\partial x^4} + 2\frac{\partial^4 u}{\partial x^2 \partial y^2} + \frac{\partial^4 u}{\partial y^4} & \\ &= f_n,\ x = x_n,\ y = y_n, \\ &= 0 \text{ otherwise,} \end{aligned} \tag{2}$$

in two dimensions. The units are dimensionless. A condition on the solution is that $u(x_n)=w_n$ or $u(x_n, y_n)=w_n$. In one dimension, u, $\partial u/\partial x$, and $\partial^2 u/\partial x^2$, the curvature, are continuous across the point where the force is acting, but $\partial^3 u/\partial x^3$ is discontinuous across such a point and the value of the discontinuity is equal to the force acting at that point (Love, 1926). A solution in one dimension is given by a third-order polynomial

$$u = a_0 + a_1 x + a_2 x^2 + a_3 x^3,$$

for each segment between the points where the forces are acting. The coefficients $a_0, \cdots, a_3$ are found by using the continuity conditions above. This solution is a cubic spline.

In two dimensions the solution of equation (2) is to be used in place of the two-dimensional, third-order piecewise polynomial fit.

Boundary conditions

The most suitable condition for the ends of the strip or edge of the thin sheet is that of freedom. For a strip, the region between the end and the extreme observation will have a linear form, and for a sheet, the area between the edge and the observations will tend to a plane as the sheet becomes larger.

For one and two dimensions, at the ends or edge, the force is zero, and the bending moment about a tangential line is zero. For one dimension, these conditions give

$$\frac{\partial^3 u}{\partial x^3} = 0, \tag{3}$$

and

$$\frac{\partial^2 u}{\partial x^2} = 0, \text{ respectively.} \tag{4}$$

For two dimensions, they give

$$\frac{\partial}{\partial x}\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) = 0, \tag{5}$$

where the normal to the edge is in the x-direction, and give (4) also. The condition that

$$u(x_n) = w_n,$$

or

$$u(x_n, y_n) = w_n, \tag{6}$$

is also a "boundary" condition.

Equation (1) with boundary conditions (3), (4), and (6) or equation (2) with boundary conditions (5), (4), and (6) are solved numerically.

Finite difference equations

Equation (2) can be derived from the principle of minimum curvature. Difference equations can be formed from equation (2) using Taylor's theorem (Young, 1962) or directly from the principle of minimum curvature. The boundary equations are more easily deduced by the latter means. Equations to be used when an observation does not lie on a grid point are more easily deduced by the former.

Consider the total squared curvature

$$C(u) = \iint\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right)^2 dxdy. \tag{7}$$

It must be shown that if a function $u(x, y)$ makes C an extremum, then it obeys equation (2), and also that if a function u obeys equation (2) then it minimizes C. Let $u(x, y)$ be a function on a region in R^2 with boundary B. Let u make C an extremum. Let $z(x, y)$ be a function of $u(x, y)$ and an arbitrary function g, with

$$g = 0 \quad \text{and} \quad \frac{\partial g}{\partial n} = 0 \text{ on } B,$$

where $\partial/\partial n$ denotes a derivative along the normal to B,

$$z(x, y) = u(x, y) + \epsilon g(x, y),$$

where ϵ is a real number.

Then

$$\left.\frac{\partial C(z)}{\partial \epsilon}\right|_{\epsilon=0} = 0,$$

and this must hold for all functions $g(x, y)$. Writing ∇^2 for

$$\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$$

$$C(z) = \iint (\nabla^2 u)^2 dxdy + 2\epsilon \iint \nabla^2 u \nabla^2 g dxdy + \epsilon^2 \iint \nabla^2 g dxdy,$$

and

$$\left.\frac{\partial C(z)}{\partial \epsilon}\right|_{\epsilon=0} = 2 \iint \nabla^2 u \nabla^2 g dxdy.$$

Using Green's theorem, (Courant and Hilbert, 1953) the right-hand side gives

$$2\left(\iint g \nabla^2(\nabla^2 u) dxdy + \int_B \nabla^2 u \frac{\partial g}{\partial n} dl - \int_B g \frac{\partial}{\partial n} (\nabla^2 u) dl \right).$$

The last two integrals vanish and leave

$$\iint g \nabla^2(\nabla^2 u) dxdy = 0.$$

Since this must hold for all g, $\nabla^2(\nabla^2 u) = 0$. Conversely, if u obeys equation (2), and if z is any other function on R^2, with $z = u$, and $\partial z/\partial n = \partial u/\partial n$ on B, we can show that $C(u) \leq C(z)$. Consider

$$C(z) - C(u) = \iint [(\nabla^2 z)^2 - (\nabla^2 u)^2] dxdy.$$

The right-hand side gives

$$\iint (\nabla^2 z - \nabla^2 u)^2 dxdy + 2 \iint \nabla^2 u (\nabla^2 z - \nabla^2 u) dxdy.$$

The last term gives upon the use of Green's theorem

$$2\left(\iint (z - u) \nabla^2(\nabla^2 u) dxdy + \int_B \nabla^2 u \frac{\partial}{\partial n} (z - u) dl - \int_B (z - u) \frac{\partial}{\partial n} (\nabla^2 u) dl \right).$$

The integrals are zero; $C(u)$ is then always less than or equal to $C(z)$.

The principle of minimum curvature is used to deduce the normal difference equations. The total squared curvature (7) is constructed directly in terms of elements of the set of grid point values

$$u_{i,j} \equiv u(x_i, y_j),$$
$$x_i = (i - 1)h,\ y_j = (j - 1)h,$$
$$i = 1, \cdots, I,\ j = 1, \cdots, J,$$

where h is the grid spacing. The discrete total squared curvature is

$$C = \sum_{i=1}^{I} \sum_{j=1}^{J} (C_{i,j})^2, \tag{8}$$

where $C_{i,j}$ is the curvature at (x_i, y_j). $C_{i,j}$ is a function of $u_{i,j}$ and some neighboring grid values; the exact set depends on the accuracy with which the curvature is to be represented.

To minimize the sum C, the functions

$$\frac{\partial C}{\partial u_{i,j}}, \quad i = 1, \cdots, I;\ j = 1, \cdots, J, \tag{9}$$

are set equal to zero (Stiefel, 1963). The resulting equations determine a set of relations between neighboring grid-point values, one relation for each grid point.

In one dimension the simplest approximation to the curvature at x_i is

$$(u_{i+1} + u_{i-1} - 2u_i)/h^2,$$

and in two dimensions at (x_i, y_i), it is

$$C_{i,j} = (u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - 4u_{i,j})/h^2. \qquad (10)$$

Along edges and rows one from the edge, and near corners, different expressions for the curvature are used. For example, at an edge $j=1$,

$$C_{i,j} = (u_{i+1,j} + u_{i-1,j} - 2u_{i,j})/h^2. \qquad (11)$$

These special cases are also included in the total for C. Away from the edges, (10) shows that a grid point value $u_{i,j}$ occurs in the expressions for,

$$C_{i,j},\ C_{i+1,j},\ C_{i-1,j},\ C_{i,j+1} \quad \text{and} \quad C_{i,j-1}.$$

Thus, only these need to be considered when equation (10) is used. Using (8), (9), and (10) the common difference equation for the biharmonic equation results:

$$\begin{aligned} &u_{i+2,j} + u_{i,j+2} + u_{i-2,j} + u_{i,j-2} \\ &+ 2(u_{i+1,j+1} + u_{i-1,j+1} + u_{i+1,j-1} + u_{i-1,j-1}) \\ &- 8(u_{i+1,j} + u_{i-1,j} + u_{i,j-1} + u_{i,j+1}) \qquad (12) \\ &+ 20u_{i,j} = 0. \end{aligned}$$

For the edge $j=1$, the difference equation is

$$\begin{aligned} u_{i-2,j} + u_{i+2,j} + u_{i,j+2} + u_{i-1,j+1} + u_{i+1,j+1} \\ - 4(u_{i-1,j} + u_{i,j+1} + u_{i+1,j}) \qquad (13) \\ + 7u_{i,j} = 0. \end{aligned}$$

A complete set is given in Appendix B.

The point boundary conditions (6) are used by setting $u_{i,j}=w_n$ wherever $u_{i,j}$ occurs in the set of linear equations, and by removing those equations which correspond to these fixed grid points.

Observation not on a grid point

If an observation does not coincide with a grid point another difference equation is required for grid points which are the vertices of the grid square in which the observation falls. The observation point becomes part of the grid.

The equation used is a special case of a general method for using a random grid for the numerical solution of differential equations. The general method is used by letting one grid point, the observation, be on an irregular grid; the remaining neighbors are on the regular grid in the difference equation relating a grid-point value to its neighbors.

Equation (2) is equivalent to

$$\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}\right)\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) = 0.$$

If

$$C_{ij} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \quad \text{at } (x_i, y_j), \qquad (14)$$

equation (14) gives the difference equation (Young, 1962),

$$\begin{aligned} C_{i+1,j} + C_{i-1,j} + C_{i,j+1} + C_{i,j-1} \\ - 4C_{i,j} = 0. \end{aligned} \qquad (15)$$

If equation (10) is used in equation (15), equation (12) results. However, we need an expression for $C_{i,j}$ which uses values of u at discrete points not lying on a regular grid.

Let u be a continuous function on the real two-dimensional space R^2 and let (x_0, y_0) be in R^2. If the set of points

$$\{x_0 + \xi_k,\ y_0 + \eta_k\}, \qquad k = 1, \cdots, 5$$

are also in R^2, then for sufficiently small ξ_k, η_k and if u has sufficiently many derivatives,

$$u_k \equiv u(x_0 + \xi_k,\ y_0 + \eta_k), \qquad k = 1, \cdots, 5,$$

is approximated by

$$u_0 + \xi_k \frac{\partial u}{\partial x}\bigg|_0 + \eta_k \frac{\partial u}{\partial y}\bigg|_0 + \tfrac{1}{2}\xi_k^2 \frac{\partial^2 u}{\partial x^2}\bigg|_0 + \xi_k\eta_k \frac{\partial^2 u}{\partial x \partial y}\bigg|_0 + \tfrac{1}{2}\eta_k^2 \frac{\partial^2 u}{\partial y^2}. \qquad (16)$$

To find an expression for

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \qquad \text{at } (x_0, y_0),$$

both sides of equations (16) are multiplied by a real number b_k and a sum is made over k, so that

$$
\begin{aligned}
\sum_{k=1}^{5} b_k u_k = u_0 \sum b_k &+ \left.\frac{\partial u}{\partial x}\right|_0 \sum b_k \xi_k \\
&+ \left.\frac{\partial u}{\partial y}\right|_0 \sum b_k \eta_k \\
&+ \frac{1}{2} \left.\frac{\partial^2 u}{\partial x^2}\right|_0 \sum b_k \xi_k^2 \qquad (17) \\
&+ \left.\frac{\partial^2 u}{\partial x \partial y}\right|_0 \sum b_k \xi_k \eta_k \\
&+ \frac{1}{2} \left.\frac{\partial^2 u}{\partial y^2}\right|_0 \sum b_k \eta_k^2 .
\end{aligned}
$$

If the b_k are chosen such that

$$
\begin{gathered}
\sum b_k \xi_k = 0, \qquad \sum b_k \eta_k = 0, \\
\sum b_k \xi_k^2 = 2, \qquad \sum b_k \xi_k \eta_k = 0, \\
\sum b_k \eta_k^2 = 2,
\end{gathered}
\qquad (18)
$$

then

$$
\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \qquad \text{at } (x_0, y_0)
$$

is approximated by

$$
\sum_{k=1}^{5} b_k u_k - u_0 \sum_{k=1}^{5} b_k . \qquad (19)
$$

The matrix

$$
\begin{Bmatrix}
\xi_1 & \xi_2 & \xi_3 & \xi_4 & \xi_5 \\
\eta_1 & \eta_2 & \eta_3 & \eta_4 & \eta_5 \\
\xi_1^2 & \xi_2^2 & \xi_3^2 & \xi_4^2 & \xi_5^2 \\
\xi_1\eta_1 & \xi_2\eta_2 & \xi_3\eta_3 & \xi_4\eta_4 & \xi_5\eta_5 \\
\eta_1^2 & \eta_2^2 & \eta_3^2 & \eta_4^2 & \eta_5^2
\end{Bmatrix}
\qquad (20)
$$

must be nonsingular, for the b_k to exist. For the present purpose, where one u_k lies off the regular grid, and the remaining four lie on the regular grid, with

$$
\xi, \eta = h, 0, -h,
$$

a suitable set is

$$
(h, -h), (0, -h), (-h, 0), (-h, h), (\xi_5, \eta_5),
$$

with $\xi_5 > 0$ and $\eta_5 > 0$. Thus, for an expression for the curvature at (x_i, y_j), we can use

$$
C_{i,j} = \sum_{k=1}^{4} b_k u_k - u_{i,j} \sum_{k=1}^{5} b_k + b_5 w_n, \qquad (21)
$$

where $\{u_k\}$ is

$$
u_{i+1,j-1},\ u_{i,j-1},\ u_{i-1,j},\ u_{i-1,j+1},
$$

and w_n is the nearby observation value.

Equation (21) can be used in (15) to give a linear equation relating a grid point to neighboring grid points and an observation. This is used in place of equation (12).

Iteration matrix

The set of linear algebraic equations (12), (13), and others are best solved iteratively (Young, 1962). Given an approximate set of $u_{i,j}$, a new set is obtained by making $u_{i,j}$ the subject of equations (12) and (13) and others. For example, (13) gives

$$
\begin{aligned}
u_{i,j}^{p+1} = [4(&u_{i-1,j}^{p} + u_{i,j+1}^{p} + u_{i+1,j}^{p}) \\
&- (u_{i-2,j}^{p} + u_{i+2,j}^{p} + u_{i,j+2}^{p} \qquad (22) \\
&\quad + u_{i-1,j+1}^{p} + u_{i+1,j+1}^{p})]/7,
\end{aligned}
$$

where the index p indicates the pth iteration. Starting values must be given, and one suitable method is to use the value of the nearest observation or a weighted sum of neighboring observations.

Iteration matrices which give faster rates of convergence than that defined by (22) are known (Young, 1962; Parter, 1959), but are not described here. The proof of the existence of a solution to the linear equations is omitted (Stiefel, 1963).

Smoothness properties

The measure of smoothness, $C = \sum (c_{i,j})^2$ is a function of h and the precision of the approximation for $C_{i,j}$. Because the linear equations are deduced from the principle of minimum C, for a given h and for a given definition of curvature, the resulting grid-point surface is smoother than, or as smooth as, any other grid-point surface. Two contour maps produced by different means but using the same data, can be compared for smoothness by digitizing the map, if necessary, and calculating the total curvature C. The map with the lower value of C is usually the more acceptable, and delineates trends more clearly.

Nothing will be said here about the convergence of the grid point values or of C, as the grid spacing tends to zero. However, for a given grid spacing, the method gives the smoothest possible contour map, and it can be used with some confidence as a representation of the given data.

Drawing lines

There are many different methods of drawing the contours once the grid (Crain, 1970) surface has been found. The method used in the examples involves a four-point cubic interpolation between grid points to find contour cuts, and then a cubic spline to join the cuts. The observations art not used. This is the weak link in the present scheme. Improvements can be made by using the observations or by using two-dimensional cubic interpolation over a grid square. The overall success of the application of minimum total curvature warrants the undertaking of further work in the improvement of details.

Table 1. The smoothest set of discrete values fixed at $i=3, 5, 8$.

i	V_i
1	-5.62
2	1.69
3	9.00
4	16.31
5	25.00
6	36.46
7	49.77
8	64.00
9	78.23
10	92.46

EXAMPLES

For each map the time for one iteration for one grid point was approximately 0.4 msec using a CDC 3600 computer. Up to 260,000 grid points have been used to contour 60,000 observations at one time. To provide edge matching when an entire survey cannot be contoured at once, data beyond the area to be contoured are used.

The iterations were discontinued when all significant relocation of contour lines had taken place.

Test cases

Two simple test examples are: (1) a one-dimensional set of data taken to lie on a straight line; and (2) a set of data points (at least four are necessary) taken to lie on a plane. In case (1) the free grid points tend to values lying on the same straight line, and in case (2) the free grid points tend to values lying in the same plane.

Table 1 gives the values of a one-dimensional set of grid points which minimize the total curvature. Grid points at $i=3, 5, 8$ are fixed and the imaginary forces required to bend the spline act at these points. The difference equations used are given in Appendix A.

Table 2 gives the values of a two-dimensional set of grid points fixed at (7, 3), (8, 5), (5, 5), (8, 8), and (4, 8). This set of grid points minimizes the sum of the point curvatures defined by equation (10).

Table 3 gives the values of a two-dimensional set of grid points fixed at infinity and at $x=.2$, $y=.3$ where the observation is $(.2)^2+(.3)^2=.13$.

Table 2. The smoothest set of grid valves fixed at (7, 3), (8, 5), (5, 5), (8, 8), (4, 8).

i/j	1	2	3	4	5	6	7	8	9	10
1	-99.34	-89.96	-80.30	-70.10	-59.19	-47.48	-35.01	-21.93	-8.44	5.25
2	-84.07	-75.42	-66.30	-56.53	-45.95	-34.46	-22.12	-9.12	4.35	18.14
3	-69.07	-61.31	-52.89	-43.67	-33.48	-22.17	-9.86	3.21	16.80	30.84
4	-54.66	-47.83	-40.14	-31.56	-21.83	-10.64	1.74	15.00	28.79	43.14
5	-41.19	-35.18	-28.14	-20.19	-11.00	0.13	12.61	26.05	40.14	54.87
6	-29.03	-23.59	-16.97	-9.55	-0.68	10.25	22.80	36.46	50.85	65.94
7	-18.57	-13.42	-7.00	-0.14	8.40	19.37	32.15	46.16	60.85	76.25
8	-9.89	-5.04	0.86	7.61	16.00	27.31	40.50	55.00	70.01	85.74
9	-2.59	2.03	7.55	14.29	22.95	34.23	47.63	62.51	78.20	94.48
10	4.00	8.15	13.01	19.37	28.03	39.44	53.34	69.00	85.67	102.78

Table 3. The smoothest set of grid valves fixed at infinity and $x=.2$, $y=.3$, with $x=0$, $y=0$ at (3, 3).

i \ j	1	2	3	4	5
1	8.00	5.00	4.00	5.00	8.00
2	5.00	2.00	1.00	2.00	5.00
3	4.00	1.00	0.00	1.00	4.00
4	5.00	2.00	1.00	2.00	5.00
5	8.00	5.00	4.00	5.00	8.00

The condition at infinity is simulated by setting grid values at (x, y) to x^2+y^2 beyond a limit, and by not using the boundary difference equations. This table shows the results of using equation (21) for the case where an observation does not fall on a grid point. The grid points used are $(-1, 1)$, $(-1, 0)$, $(0, -1)$, $(1, -1)$, and the observation at $(.2, .3)$. The matrix (20) is

$$\begin{pmatrix} -1 & -1 & 0 & 1 & .2 \\ 1 & 0 & -1 & -1 & .3 \\ 1 & 1 & 0 & 1 & .04 \\ -1 & 0 & 0 & -1 & .06 \\ 1 & 0 & 1 & 1 & .09 \end{pmatrix}$$

and the resulting coefficients b_k, $k=1, \cdots, 5$ are .68, .60, .73, .48, and 2.67.

These are used in equations (21) and (15) to give a value for the grid point at $x=0$, $y=0$.

These and other higher-order surfaces test the method in general and the difference equations in particular. The illustrated examples use real data.

Almost uniform data

Figure 1 is the resulting contour map for gravimetric data sampled in mgal on a nominal 11 km

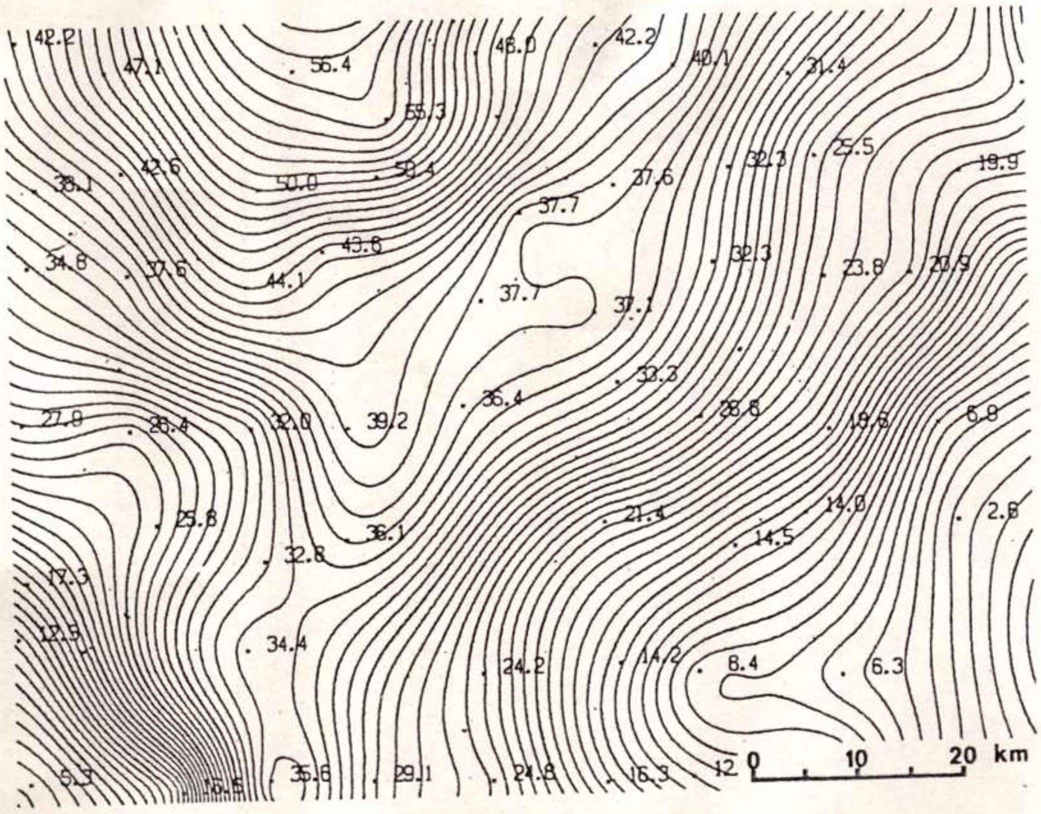


FIG. 1. Gravity data contoured at 1-mgal intervals using a grid spacing of 1.85 km.

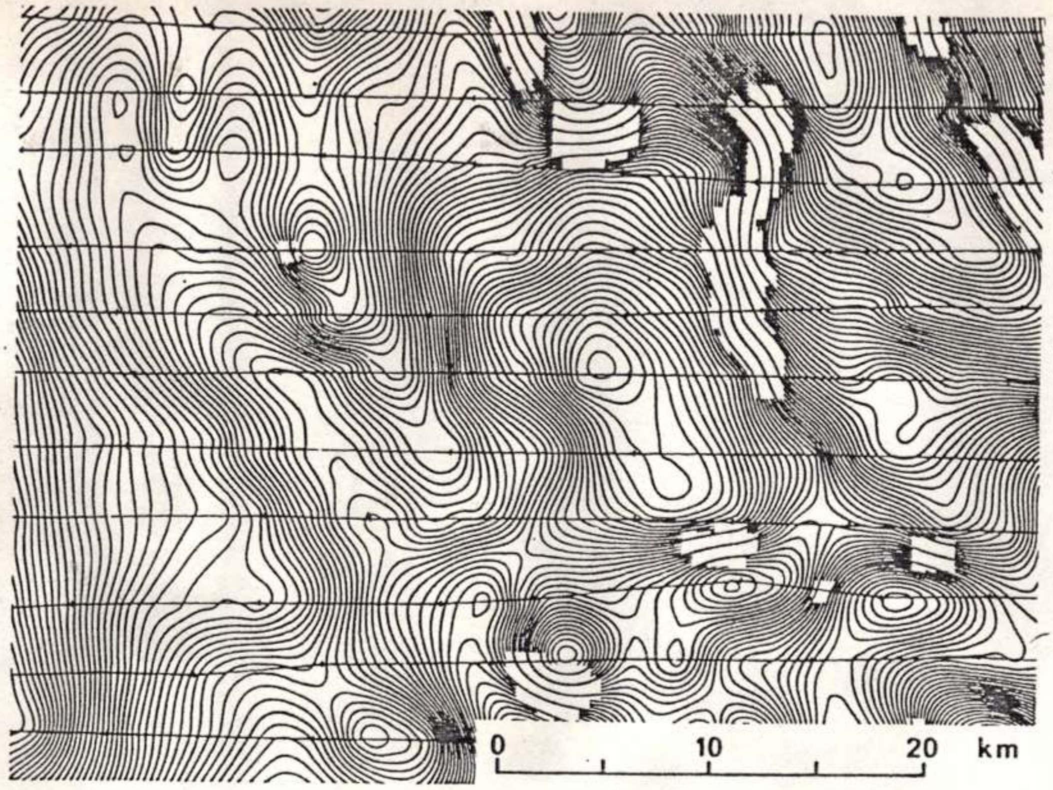


FIG. 2. Aeromagnetic data contoured at 10-gamma intervals using a grid spacing of 0.5 km.

network. The grid spacing was 1.85 km and the number of iterations was 90. The number of grid points was 1500. The smoothness of the interpolating grid surface is apparent. The shapes of contours for data for this type generally agree with those of draftsmen. Differences occur when the interpolating grid surface lies outside the range of a closed group of observations.

The deficiency in the line-drawing routine shows itself when the 50-mgal contour does not pass exactly through a 50-mgal observation.

Line data

A more difficult set of data to contour is one whose density of sampling is not isotropic. The data for the total intensity aeromagnetic maps of Figure 2 and Figure 3 were taken at 0.8 km intervals along flight-lines nominally 3.2 km apart and at a height of 650 m above ground. The grid spacing used in the contouring was 0.5 km and the number of iterations was 60. The number of grid points is 6000 in Figure 2 and 16,500 in Figure 3.

The general smoothness is satisfactory although a relatively rough profile along a line has an effect on the contours close to the flight-line, and this may not be desirable. The interpolation is not a product of one-dimensional splines. A section across the flight-lines is not the result of a one-dimensional spline. This may be a drawback in some cases, but enables trends not lying at right-angles to the flight path to be displayed.

CONCLUSION

The principle of minimum total curvature provides a method of two-dimensional interpolation which allows a computer to draw reasonable maps of geophysical data. The results are not always as a draftsman would have them, but are an adequate substitute in most cases.

ACKNOWLEDGMENT

The author wishes to thank the Director of the Bureau of Mineral Resources for permission to publish this paper.

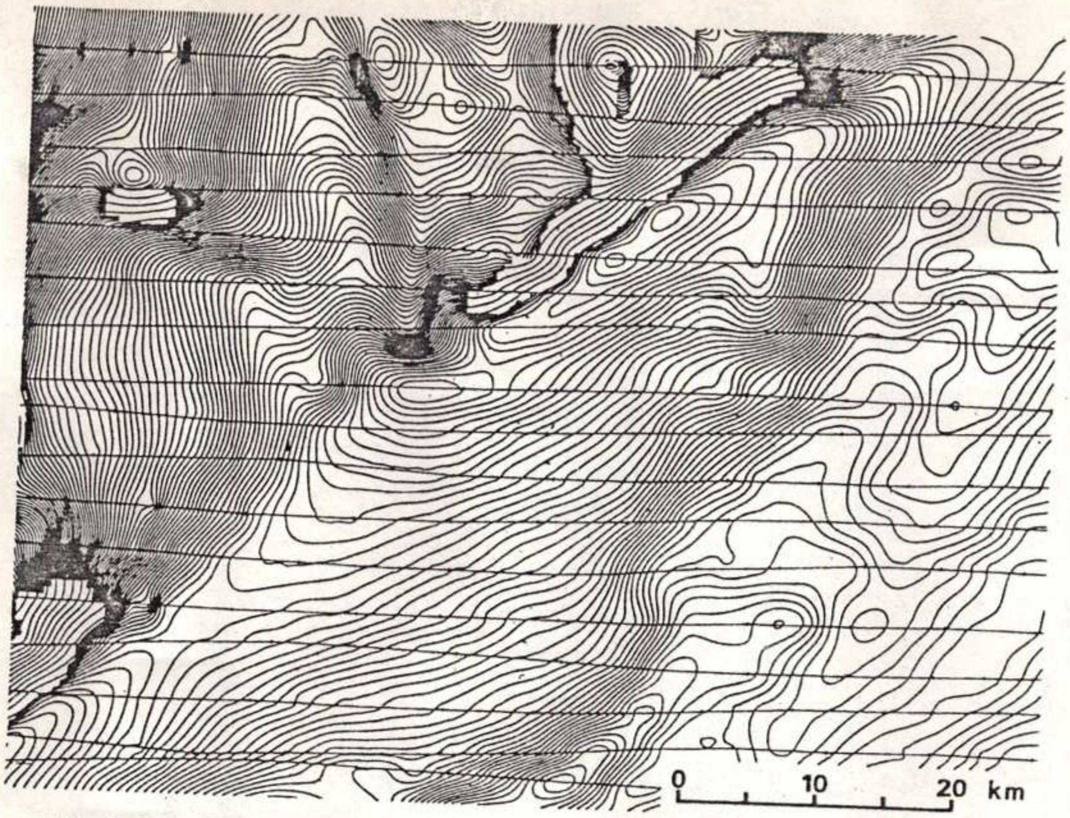


FIG. 3. Aeromagnetic data contoured at 10-gamma intervals using a grid spacing of 0.5 km.

REFERENCES

Ahlberg, J. H., Nilson, E. N., and Walsh, J. L., 1967, The theory of splines and their applications: New York, Academic Press.

Bhattacharyya, B. K., 1969, Bicubic spline interpolation as a method for treatment of potential field data: Geophysics, v. 34, p. 402–423.

Cole, A. J., 1968, Algorithm for the production of contour maps from scattered data: Nature, v. 220, p. 92–94.

Courant, R., and Hilbert, D., 1953, Methods of mathematical physics: New York, Interscience Publishers, Inc.

Crain, I. K., 1970, Computer interpolation and contouring of two-dimensional data: A review: Geoexplor., v. 8, p. 71–86.

Crain, I. K., and Bhattacharyya, B. K., 1967, Treatment of non-equispaced two-dimensional data with a digital computer: Geoexplor., v. 5, p. 173–194.

De Boor, 1962, Bicubic spline interpolation: J. Math. and Phys., v. 41, p. 212–218.

Hessing, R. C., Lee, H. K., Pierce, A., and Powers, E. N., 1972, Automatic contouring using bicubic functions: Geophysics, v. 37 p. 669–674.

Love, A. E. H., 1926, The mathematical theory of elasticity: Cambridge, University Press.

McIntyre, D. B., Pollard, D. D., and Smith, R., 1968, Computer programs for automatic contouring: Kansas Geol. Surv., Comput. Contr. no. 28.

Parter, S. V., 1959, On two-line iterative methods for the Laplace and biharmonic difference equations: Num. Math., v. 1, p. 240–252.

Pelto, Chester R., Elkins, Thomas A., and Boyd, H. A., 1968, Automatic contouring of irregularly spaced data: Geophysics, v. 33, p. 424–430.

Smith, F. G., 1968, Three computer programs for contouring map data: Can. J. Earth Sci., v. 5, p. 324–327.

Stiefel, E. L., 1963, An introduction to numerical mathematics: New York, Academic Press.

Young, D., 1962, The numerical solution of elliptic and parabolic partial differential equations, *in* Survey of numerical analysis, edited by Todd, J.: New York, McGraw-Hill Book Co., Inc.

APPENDIX A

A set of difference equations for one-dimensional interpolation is given. The curvature used is

$$C_i = (u_{i+1} + u_{i-1} - 2u_i)/h^2.$$

Normal

Away from the ends, use

$$u_{i-2} + u_{i+2} - 4(u_{i-1} + u_{i+1}) + 6u_i = 0.$$

End

At the end $i=1$, use

GEOPHYSICS, VOL. 37, NO. 4 (AUGUST 1972), P. 669–674, 4 FIGS.

AUTOMATIC CONTOURING USING BICUBIC FUNCTIONS†

R. C. HESSING,* HENRY K. LEE,* ALAN PIERCE,* AND ELDON N. POWERS*

A method is described for using a digital computer to construct contour maps automatically. Contour lines produced by this method have correct relations to given discrete data points regardless of the spatial distribution of these points. The computer-generated maps are comparable to those drawn manually.

The region to be contoured is divided into quadrilaterals whose vertices include the data points. After supplying values at each of the remaining vertices by using a surface-fitting technique, bicubic functions are constructed on each quadrilateral to form a smooth surface through the data points. Points on a contour line are obtained from these surfaces by solving the resulting cubic equations.

The bicubic functions may be used for other calculations consistent with the contour maps, such as interpolation of equally spaced values, calculation of cross-sections, and volume calculations.

INTRODUCTION

Contour-map displays of data are widely used in many engineering and scientific applications. In the petroleum industry, contour mapping by digital computer has been used as an effective tool to analyze geological and geophysical data and to depict subsurface structures. Thus, computer mapping is playing a vital role in petroleum exploration and production.

Several articles have recently appeared on automatic contouring (Morse, 1969; Cottafava and Le Moli, 1969; Pelto et al, 1968). Most automatic contouring schemes generate a regular grid prior to contouring. The original data points which do not lie on the grid are neglected after the grid has been constructed. Thus, contouring by this scheme works fairly well if the data are uniformly spaced. For irregularly spaced data, the grid approach may be unable to honor every data point, and the contour lines will fail to represent a surface which contains the original data points. This paper describes a method that honors every data point by constructing a smooth surface through the original data. The proposed method makes use of multivariable curve interpolation by Ferguson (1964).

DESCRIPTION OF METHOD

Let D be any set of points $d_k=(x_k, y_k, z_k)$, $k=1, 2, \cdots, K$, obtained from a continuous function of two variables. In order to use the method suggested by Ferguson (1964), D must be a subset of an array $\{P_{n,m}\}$, $n=1, 2, \cdots, N$, $m=1, 2, \cdots, M$, arranged so that the structure obtained by connecting adjacent points by straight-line segments is topologically equivalent to an $N \times M$ planar rectangular grid.

The array $\{P_{n,m}\}$ is constructed by dividing the set D into "vertical" subsets $V_1, V_2, \cdots, V_M$ which satisfy the following conditions:

1) If d_a is a member of V_m and d_b is a member of V_{m+1}, then $x_a < x_b$.
2) If d_a and d_b are members of the same vertical subset V_m, the line connecting (x_a, y_a) and (x_b, y_b) must form an angle with the y axis that is less than $\pi/4$.
3) Each V_m contains at least one member of D.

† Manuscript received by the Editor October 30, 1971; revised manuscript received January 12, 1972.
* Amoco Production Company, Tulsa, Oklahoma 74102.

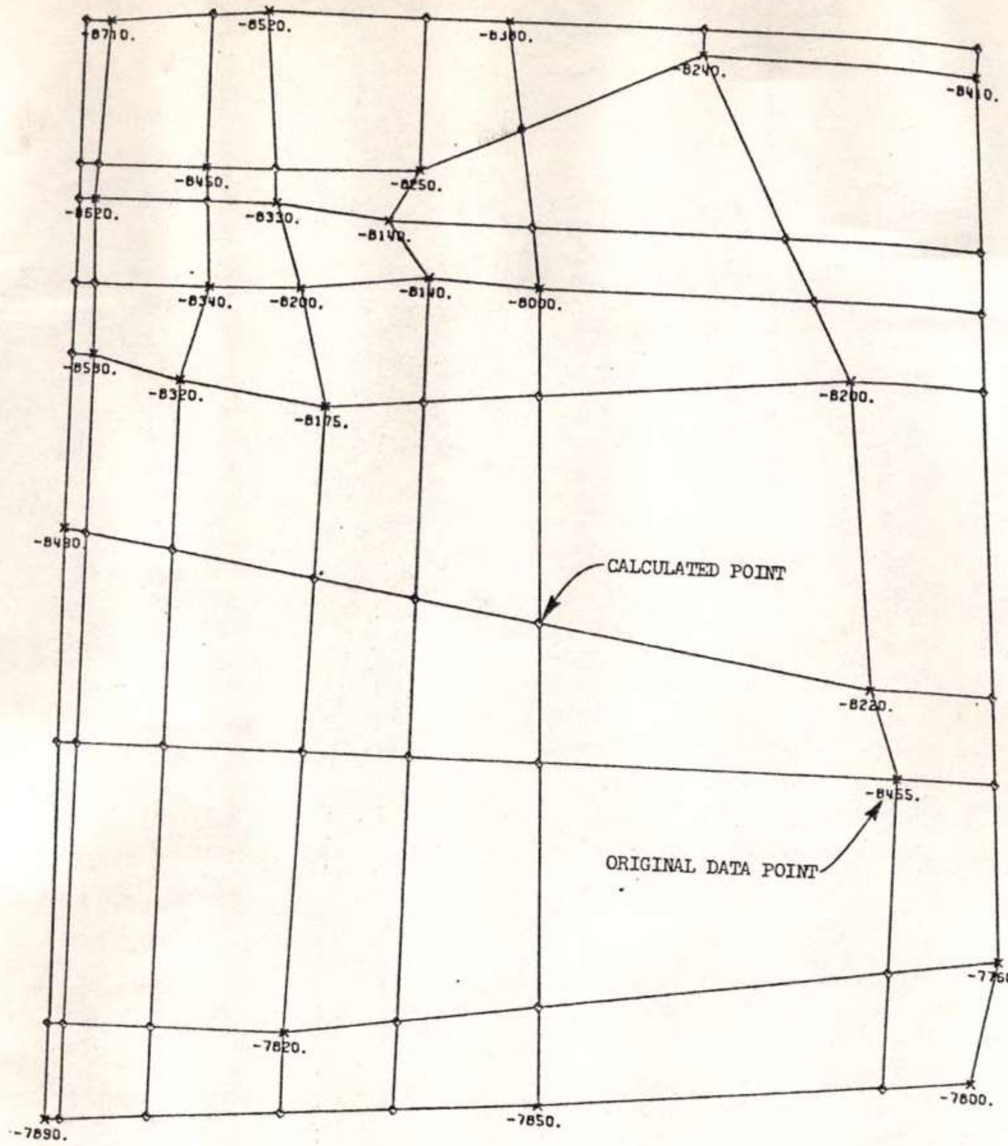


FIG. 1. Planar "grid" of sample data case.

4) If d_a is a member of V_1 or V_M it must also be on the boundary of the convex hull of D.

Similar conditions are used to divide the set D into "horizontal" subsets $H_1, H_2, \cdots, H_N$:

1) If d_a is a member of H_n and d_b is a member of H_{n+1}, then $y_a < y_b$.
2) If d_a and d_b are members of the same horizontal subset H_n, the line connecting (x_a, y_a) and (x_b, y_b) must form an angle with the x axis that is less than $\pi/4$.
3) Each H_n contains at least one member of D.
4) If d_a is a member of H_1 or H_N it must also be on the boundary of the convex hull of D.

For reasons of economy the number of these vertical and horizontal sets should be kept to a minimum; the number of such sets does not significantly affect the appearance of the map.

Now $I_{n,m} = H_n \cap V_m$ contains at most one element. If $I_{n,m}$ is nonempty, call the element it contains $P_{n,m}$. If $I_{n,m}$ is empty, a point must be constructed that will satisfy requirements analogous to conditions 1) and 2) for V_m and H_n.

The x, y coordinates of a suitable point $P_{n,m}$ where $I_{n,m}$ is empty can easily be found by the solution of two linear equations involving line segments through data points in V_m and H_n. A sample data case depiciting the x, y locations of the $\{P_{n,m}\}$ array is shown in Figure 1.

At each point in the array $\{P_{n,m}\}$ which is not a data point, a z value is obtained by fitting a polynomial to the surrounding data points. A weighted least-squares fit is used. The weight assigned to each of the contributing data points is based on its distance from the point being generated and its position in relation to the other data points. The reciprocal of the square of the distance gives greater importance to nearby points, while relatively isolated data points are given more credence by diminishing the effect of clusters. After all of the nondata points in the array $\{P_{n,m}\}$ have been assigned values, they have the same status as the original data points. Determination of the weights is described in some detail below. A recent report by Shepard (1968) presents another approach to this type of interpolation problem.

Choices are first made for several fixed numbers whose use is described below: the number $4L$ of subdivisions of the circle; the number $K+1$ of nonzero levels of intermediate weights; the sequence $W_0 > W_1 >, \cdots, > W_K > W_{K+1} = W_{K+2} =, \cdots, = 0$ of such weights.

Now let $(x_{n,m}, y_{n,m})$ be a point at which a value is to be supplied and let (a_i, b_i, c_i), $i = 1, 2, \cdots, I$, be the data points. Let

$$d_i^2 = (a_i - x_{n,m})^2 + (b_i - y_{n,m})^2 \text{ and}$$
$$\theta_i = \tan^{-1}[(b_i - y_{n,m})/(a_i - x_{n,m})].$$

Suppose $1 \leq i \leq I$. For each l, $0 \leq l \leq L-1$, consider the four sectors

$$\pi l/2L \leq \theta < \pi/2 + \pi l/2L,$$
$$\pi/2 + \pi l/2L \leq \theta < \pi + \pi l/2L,$$
$$\pi + \pi l/2L \leq \theta < 3\pi/2 + \pi l/2L, \text{ and}$$
$$3\pi/2 + \pi l/2L \leq \theta < 0 \text{ or } 0 \leq \theta < \pi l/2L,$$

and let $J_{i,l}$ be the number of data points (a_j, b_j, c_j) such that θ_j and θ_i are in the same sector and $d_j^2 < d_i^2$. The weight of the point (a_i, b_i, c_i) is then

$$(1/d_i^2) \sum_{l=0}^{L-1} W_{J_{i,l}}.$$

It can now be seen that the dependence of the weights on L, K, and $W_0, W_1, \cdots, W_K$ is quite complex. Over a period of time several values were tried for each of the parameters. Obviously computations increase with increasing L and K, yet L and K must be sufficiently large (and $W_0, W_1, \cdots, W_K$ decreasing sharply enough) to provide a smooth transition when the process is repeated at nearby points. Empirical evidence has demonstrated that $W_0 = 1$, $W_1 = 1/2, \cdots, W_K = W_{K-1}/2$ and $L = 10$, $K = 4$ are reasonable values for the parameters.

Now a smooth surface (i.e., one with continous first-order partial derivatives) through the points of the array is constructed as a composite of surfaces, one surface for each quadrilateral determined by this array. For each point $P_{n,m} = (x_{n,m}, y_{n,m}, z_{n,m})$ in the array (which is neither the top of a column nor the right of a row) we follow Ferguson (1964) to construct a function

$$S_{n,m}(u, v) = [X_{n,m}(u, v), Y_{n,m}(u, v), Z_{n,m}(u, v)]$$

with the parameters restricted to the unit square $0 \leq u, v \leq 1$ and with bicubic components

$$X_{n,m}(u, v) = \sum_{p=0}^{3} \sum_{q=0}^{3} a_{p,q}^{n,m} u^p v^q, \qquad (1)$$

$$Y_{n,m}(u, v) = \sum_{p=0}^{3} \sum_{q=0}^{3} b_{p,q}^{n,m} u^p v^q, \qquad (2)$$

and

$$Z_{n,m}(u, v) = \sum_{p=0}^{3} \sum_{q=0}^{3} c_{p,q}^{n,m} u^p v^q \qquad (3)$$

(see also Coons, 1967).

To determine the coefficients of these bicubics, we require first that

$$S_{n,m}(u, v) = \begin{cases} P_{n,m} & \text{if } u = v = 0, \\ P_{n,m+1} & \text{if } u = 1, v = 0, \\ P_{n+1,m} & \text{if } u = 0, v = 1, \\ P_{n+1,m+1} & \text{if } u = v = 1, \end{cases} \qquad (4)$$

so that the surface $S_{n,m}$ contains the points $P_{n,m}$, $P_{n,m+1}$, $P_{n+1,m}$, and $P_{n+1,m+1}$. This surface is defined over a region which is approximately the quadrilateral determined by these points or, more precisely, over the region to be bounded by the curves $[X_{n,m}(0, v), Y_{n,m}(0, v)]$, $[X_{n,m}(u, 0), Y_{n,m}(u, 0)]$, $[X_{n,m}(1, v), Y_{n,m}(1, v)]$, and $[X_{n,m}(u, 1), Y_{n,m}(u, 1)]$.

If we have previously calculated prescribed derivatives $R_{n,m}$ and $T_{n,m}$ at the points $P_{n,m}$, we may require also that

$$\frac{\partial S_{n,m}}{\partial u}(u, v) = \begin{cases} R_{n,m} & \text{if } u = v = 0 \\ R_{n,m+1} & \text{if } u = 1, v = 0 \\ R_{n+1,m} & \text{if } u = 0, v = 1 \\ R_{n+1,m+1} & \text{if } u = v = 1 \end{cases} \tag{5}$$

and

$$\frac{\partial S_{n,m}}{\partial v}(u, v) = \begin{cases} T_{n,m} & \text{if } u = v = 0 \\ T_{n,m+1} & \text{if } u = 1, v = 0 \\ T_{n+1,m} & \text{if } u = 0, v = 1 \\ T_{n+1,m+1} & \text{if } u = v = 1. \end{cases} \tag{6}$$

Together with the condition that $\partial^2 S_{n,m}/\partial u \partial v$ vanish at (0, 0), (0, 1), (1, 0), and (1, 1), equations (4), (5), and (6) determine the coefficients of the bicubics in equations (1), (2), and (3) in terms of the arrays $\{P_{n,m}\}$, $\{R_{n,m}\}$, and $\{T_{n,m}\}$.

It can be shown (Ferguson, 1964) that the composite surface constructed in this manner is continuous, i.e.,

$$S_{n,m}(u, v) = \begin{cases} S_{n,m-1}(1, v) & \text{if } u = 0 \\ S_{n,m+1}(0, v) & \text{if } u = 1 \\ S_{n-1,m}(u, 1) & \text{if } v = 0 \\ S_{n+1,m}(u, 0) & \text{if } v = 1, \end{cases} \tag{7}$$

and smooth:

$$\frac{\partial S_{n,m}}{\partial u}(u, v) = \begin{cases} \dfrac{\partial S_{n,m-1}}{\partial u}(1, v) & \text{if } u = 0 \\ \dfrac{\partial S_{n,m+1}}{\partial u}(0, v) & \text{if } u = 1 \\ \dfrac{\partial S_{n-1,m}}{\partial u}(u, 1) & \text{if } v = 0 \\ \dfrac{\partial S_{n+1,m}}{\partial u}(u, 0) & \text{if } v = 1, \end{cases} \tag{8}$$

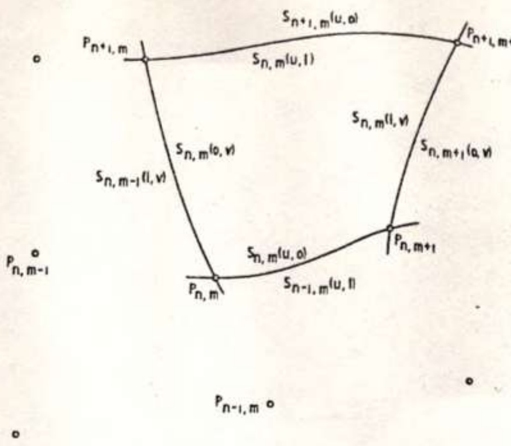


FIG. 2. An element of the composite surface.

and

$$\frac{\partial S_{n,m}}{\partial v}(u, v) = \begin{cases} \dfrac{\partial S_{n,m-1}}{\partial v}(1, v) & \text{if } u = 0 \\ \dfrac{\partial S_{n,m+1}}{\partial v}(0, v) & \text{if } u = 1 \\ \dfrac{\partial S_{n-1,m}}{\partial v}(u, 1) & \text{if } v = 0 \\ \dfrac{\partial S_{n+1,m}}{\partial v}(u, 0) & \text{if } v = 1 \end{cases} \tag{9}$$

(see Figure 2).

The derivatives $R_{n,m}$ and $T_{n,m}$ are calculated from the points near $P_{n,m}$ in order to avoid the lengthy spline-type calculation of Ferguson (1964). We let

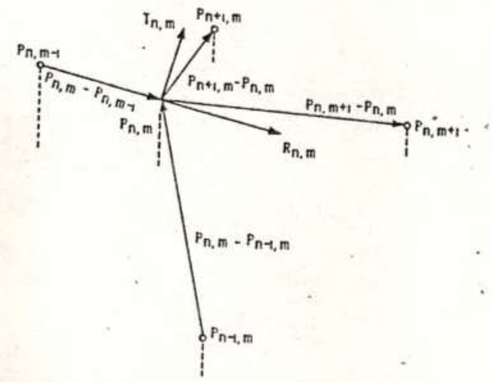


FIG. 3. Derivative calculation.

$$D_1^2 = (x_{n,m} - x_{n,m-1})^2 + (y_{n,m} - y_{n,m-1})^2,$$
$$D_2^2 = (x_{n,m+1} - x_{n,m})^2 + (y_{n,m+1} - y_{n,m})^2,$$

and set

$$E = (1/D_1^2)(P_{n,m} - P_{n,m-1}) + (1/D_2^2)(P_{n,m+1} - P_{n,m}).$$

Thus $E=(E_1, E_2, E_3)$ is a weighted sum of $P_{n,m} - P_{n,m-1}$ and $P_{n,m+1} - P_{n,m}$, greater weight being given to the vector connecting the closer point.

We finally set

$$R_{n,m} = (\min (x_{n,m} - x_{n,m-1}, x_{n,m+1} - x_{n,m})) \cdot (1/E_1)E.$$

Thus the x coordinate of $R_{n,m}$ is $\min (x_{n,m} - x_{n,m-1}, x_{n,m+1} - x_{n,m})$ which insures (as the method of Ferguson does not) that $X_{n,m}(u, 0)$ is monotonically increasing.

The vector $T_{n,m}$ is calculated in the same fashion, using instead the points $P_{n+1,m}$ and $P_{n-1,m}$ and interchanging the roles of x and y (see Figure 3).

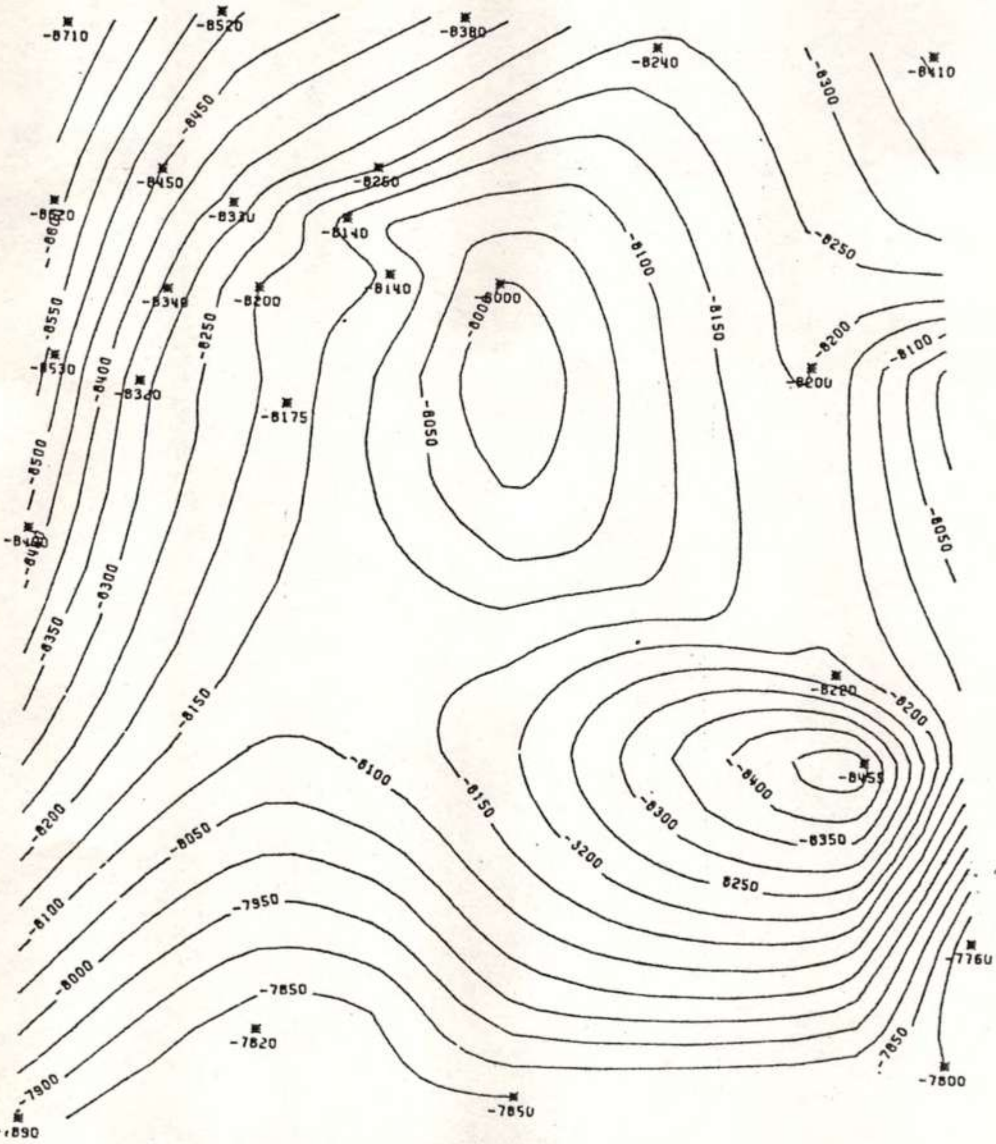


FIG. 4. Contour map of sample data case.

Once the coefficients of the bicubic function have been determined, the interpolation is essentially complete and the composite surface is available for many applications—contouring, determination of a finer grid, calculation of derivatives, approximation of double integrals, etc.

Typically, automatic contouring methods solve for z at given coordinates, and the location of the contour height is found by straight line interpolation. Linear interpolation in the true sense of the word is not used in this particular method. Points on the intersection of the plane $z=H$ with the surface $S_{n,m}$ are found for a given contour height H. Given a value of v, $0 \leq v \leq 1$, the cubic equation

$$\sum_{p=0}^{3}\left(\sum_{q=0}^{3} c_{p,q}^{n,m} v^{q}\right) u^{p} = H$$

is solved for u. For each solution u between 0 and 1, x and y are calculated from equations (1) and (2). The contour line is approximated by the line segments between these points. Usually five solutions are adequate to yield a smooth contour inside the quadrilateral. If the contour is not sufficiently smooth, more solutions are obtained. The roles of u and v may be reversed depending on the entry and exit positions of the contour.

CONCLUSIONS

The procedure described automatically produces, as noted, sufficiently smooth contour lines which are correctly related to the data points. A contour map generated by this method from the sample data case is shown in Figure 4. Maps produced are comparable to those drawn manually. Also, the use made of bicubic functions gives this method additional advantages.

Most automatic contouring methods require extensive interpolation directly from the set of data points to form a closely spaced grid. Here such direct interpolation is necessary only for the positions in the array $\{P_{n,m}\}$ which are not occupied by data points. In some cases this interpolation is omitted entirely [i.e., if the original data are arranged in a rectangular grid or otherwise meet the requirements in Ferguson (1964)]. Even so, some benefit could be obtained by developing a more efficient interpolation method and/or finding new methods of constructing the array $\{P_{n,m}\}$ in order to reduce the number of positions not occupied by data points.

Once, however, this preliminary interpolation has been performed and the bicubic functions constructed, the determination of contour lines can be followed or replaced by further interpolation. Indeed, these functions are available for many applications such as calculation of derivatives or approximation of double integrals—at little additional cost and consistent with the contour map.

ACKNOWLEDGMENTS

The authors wish to acknowledge the efforts and achievements of J. G. Steward and K. R. Driessel, Amoco Production Company, in the early development and analysis of automatic contouring methods. We are grateful also to L. F. Kemp, Amoco Production Company, for his assistance, and to many other staff members of Amoco Production Company for their valuable comments and suggestions.

REFERENCES

Coons, S. A., 1967, Surfaces for computer-aided design of space forms: Project MAC Tech. Rep. MAC-TR-41, MIT.

Cottafava, G., and Le Moli, G., 1969, Automatic contour map: Comm. ACM, v. 12, p. 386–391.

Ferguson, J., 1964, Multivariable curve interpolation: J. ACM, v. 11, p. 221–228.

Morse, S. P., 1969, Concepts of use in contour map processing: Comm. ACM, v. 12, p. 147–169.

Pelto, C. R., Elkins, T. A., and Boyd, H. A., 1968, Automatic contouring of irregularly spaced data: Geophysics, v. 33, p. 424–430.

Shepard, D., 1968, A two-dimensional interpolation function for irregularly-spaced data: Proc.—1968 ACM Nat. Conf., p. 517–524.

S4070

CONTOURING AND THE CONTOUR MAP: A NEW PERSPECTIVE*

BY

A. E. WREN**

Abstract

Wren, A. E., 1975, Contouring and the Contour Map: A New Perspective, Geophysical Prospecting 23, 1-17.

With few exceptions, traditional approaches to contouring have been too subjective. Contouring and contour maps are too often discussed in terms more appropriate to art than to science. With hand contouring there is some justification for this attitude; with machine (i.e. programmed) contouring there is none.

Hand contouring is highly susceptible to interpretive judgement and the interpreter is not bound by rigid mathematical constraints. Hence, in allowing for the interpreter's "freedom of expression" it may be difficult to evaluate hand contouring in a tot[...] analytical and objective manner. Machine contouring, however, is based upon mathematical formulation. It is therefore a consistent and objective procedure, ideally suited to objective definition and analysis.

It can be demonstrated that the combined process of sampling plus contouring constitutes a two-dimensional filter. The contouring component is that part which introduces "distortion" or wavenumber discrimination. An ideal contour package is one that acts as an all-pass filter where the distortion is zero.

The application of filter theory to the evaluation of a machine contour package and its performance permits description in the more convenient language and terminology of the wavenumber domain, rather than that of the space domain. A more important advantage is that the contour package can be subjected to the various standards of filter evaluation such as amplitude and phase response.

The practical application, as well as the benefit, of this approach is revealed through the comparison, in both the space and wavenumber domains, of contour maps generated from various machine contour packages.

Introduction

Earth scientists have been using contour maps for many years as a fundamental tool to represent a variety of data sets. In recent years the advent of computers has led to attempts to duplicate interpretive contouring techniques by machine methods (Walters 1969, p. 2324). Although considerable progress has been made in developing an understanding of the human logic of contouring

* Paper presented at the 42nd Annual International Meeting of the S.E.G., Anaheim, California, November 26-30, 1972.

** Amoco Canada Petroleum Company Ltd., Calgary, Alberta Canada.

and many computer programs are now available to produce "reasonable" contour maps, a technique to evaluate the reliability of the final map within the context of the data itself has not yet been developed.

This paper will discuss one measure of how faithfully the final contour map depicts the actual surface from which the sample data points have been obtained. The "actual surface" is represented by some finite number of observations from discrete data points x, y, z in the region R concerned. The contouring process then becomes the construction by some operator (human or machine) of a hypothetical surface which is admissible. It can be argued that the classification of surfaces derived from a machine contour package can only be accomplished empirically and intuitively by one who is familiar with the region, the data, and the conceptual situation, or by a machine programmed to play by a prespecified set of rules pertaining to predetermined geological models.

With few exceptions, traditional approaches to contouring have been too subjective and contouring and contour maps are usually discussed in terms more appropriate to art than to science. With hand contouring there may be some justification for this attitude; with machine, or programmed, contouring there is none. Hand contouring is highly susceptible to interpretive judgement or, as may be, to interpretive license where an interpreter is not bound by rigid mathematical constraints. Hence, in allowing for an interpreter's "freedom of expression" it is difficult to evaluate contouring in a totally analytical and objective manner. Machine contouring, however, is based on mathematical formulation and is ideally suited to objective definition and analysis.

Case History

The thesis is developed in the manner of a case history where a series of machine contoured maps from a common set of data are examined. These maps are produced from several different machine programs. Some of these are available commercially, the others are "in-house". The performance of each package is compared in both the space and wavenumber domains. A set of aeromagnetic data from Western Canada is used to illustrate the technique but it should be emphasized that any type of data set would suffice.

The flight line pattern is shown on figure 1. The mean spacing between east-west lines is one mile while the mean spacing between north-south tie-lines is approximately four miles. The field data were recorded at one sample per second which, at the survey flight speed, corresponds to a spatial horizontal sample interval of 61 m (200 feet). The limited available core of the computer necessitated resampling and this involved taking every sixth value, corresponding to a spatial interval of 366 m (1200 feet). After the necessary corrections were applied, an x, y rectangular coordinate grid system was superim-

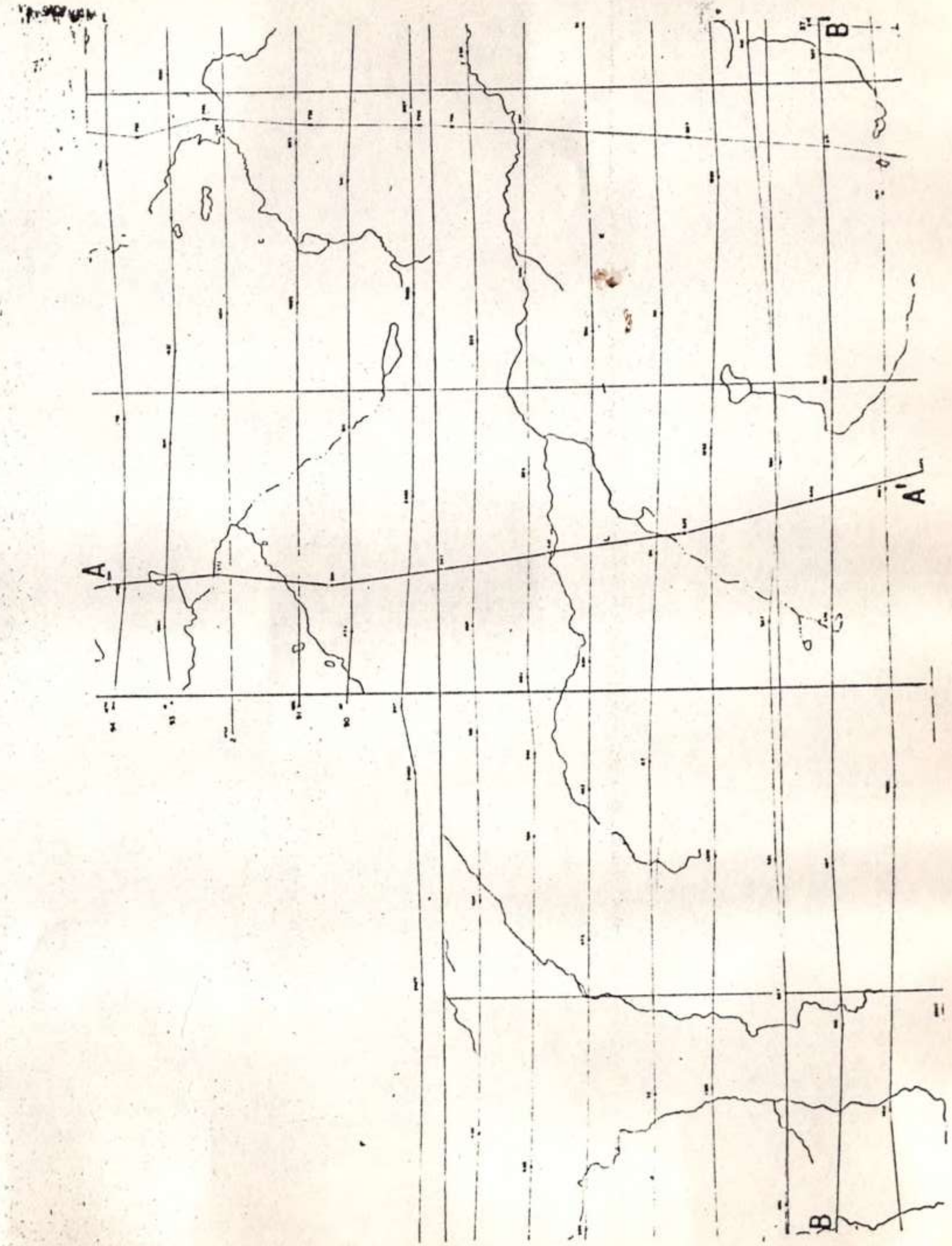


Fig. 1. Flight line map.

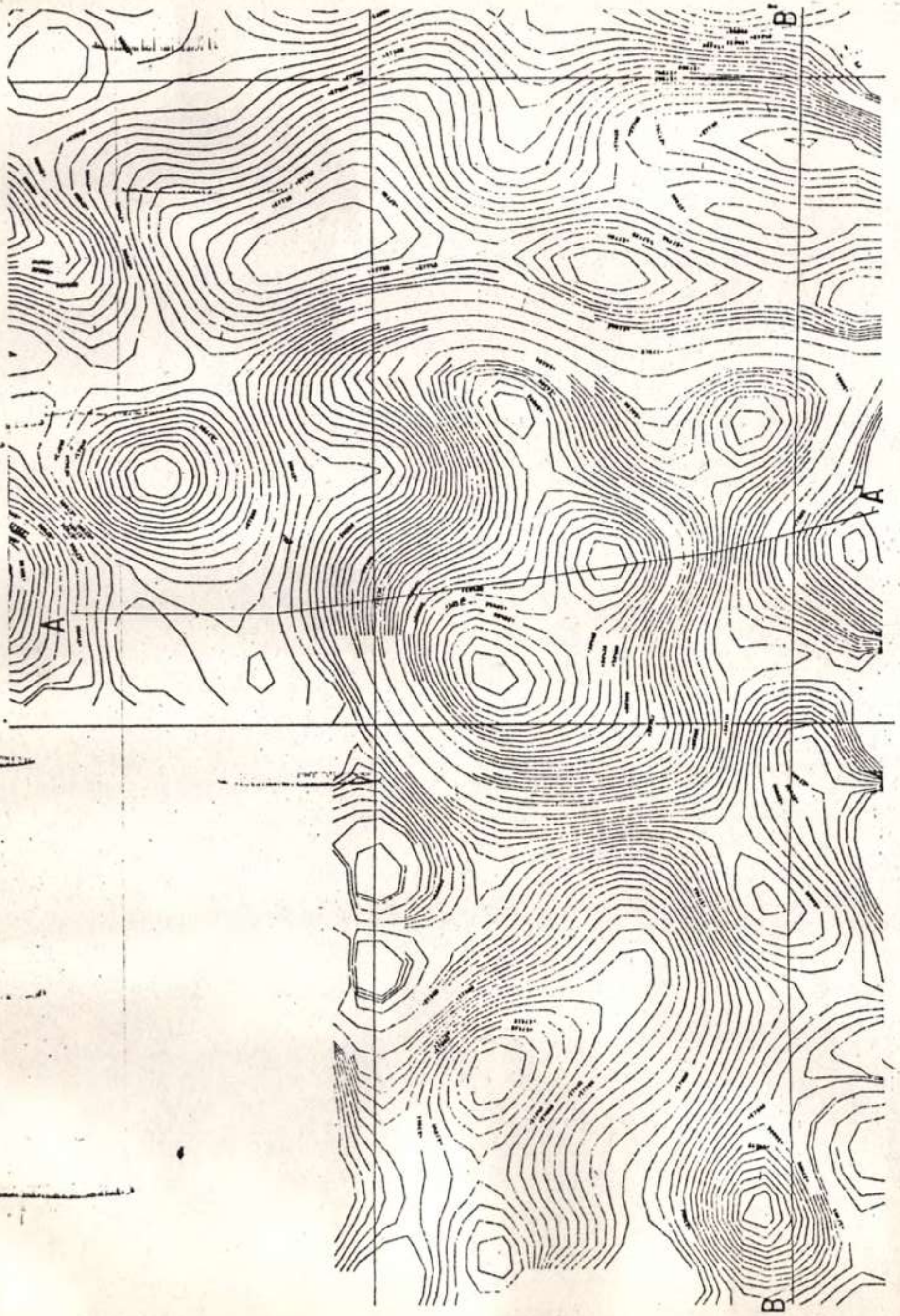


Fig. 2. Contour map output (program A).

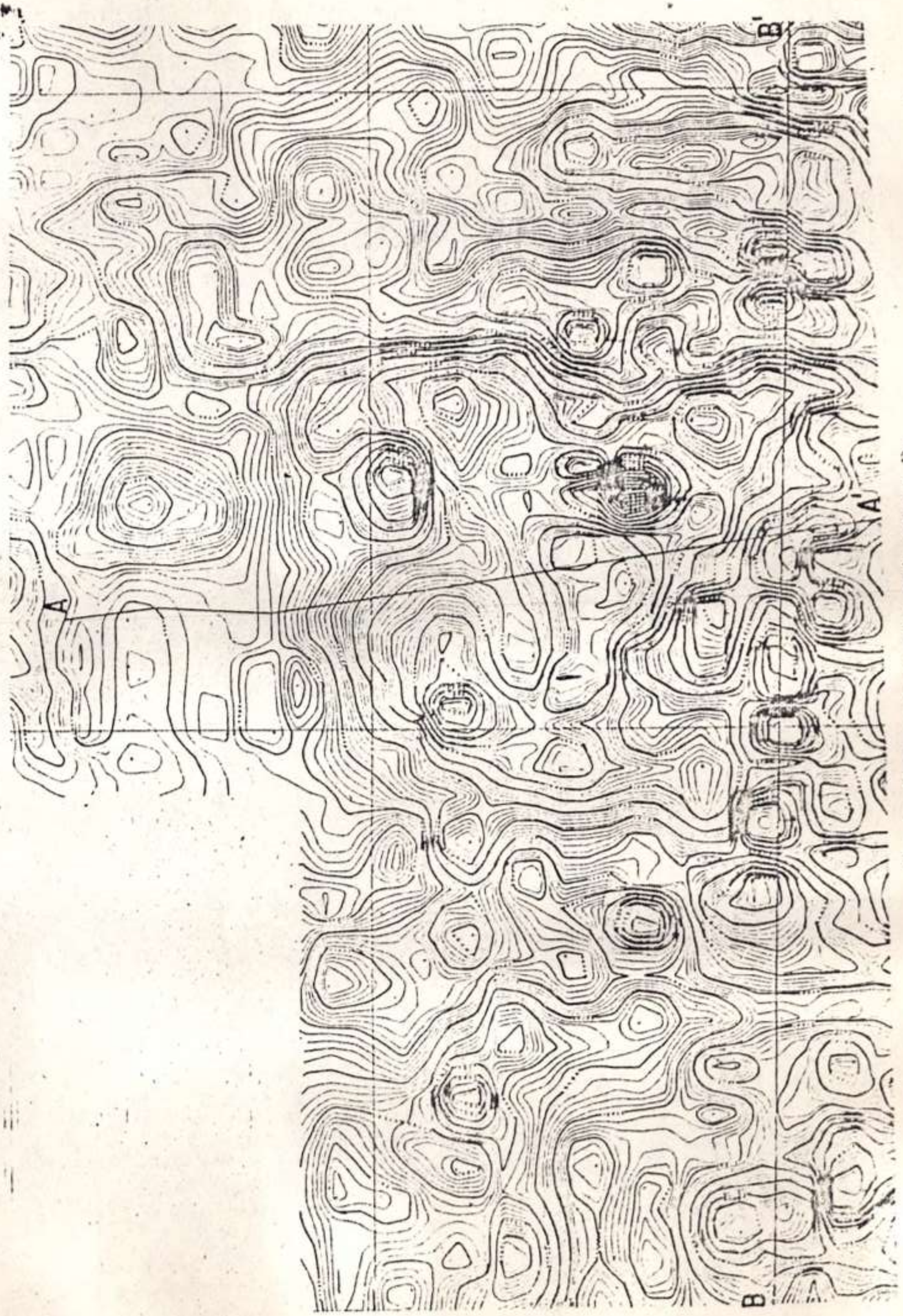


Fig. 3. Contour map output (program B)

posed on the location data. The input work tape thus lists the sampled data in x, y, z format, where z is the total magnetic intensity in gamma defined at a point in the x-y plane. The non-randomness of the spatial distribution coupled with the high density along flight lines is identical to a typical seismic sample distribution.

The Analysis

Figure 2 illustrates the first contoured map generated from the program referred to as A. Inspection of this map shows that much of the high wavenumber, or detailed information (signal and noise) apparent on the analog flight records had been eliminated in the contour process. Accordingly, the work tape was input to a second program referred to as B and the data re-contoured as shown on figure 3. The control parameters (cell size, smoothing factor, etc.) are of the same order, but visual comparisons between the two maps indicated major differences in anomaly amplitudes and wavenumber. This suggests the need for a more objective and rigorous analysis to determine the precise nature of the differences between the maps, and the true relationship of the maps to the input data.

The first step is to make the analysis comparative. This is achieved by contouring the same data set with a series of different programs (C_1, C_2, D). The results are shown on figures 4 to 6. The difference between C_1 and C_2 is one of cell size only. The second step is to determine an objective evaluation procedure.

Existing Map Analysis Techniques

Until recently, the comparison of contour maps has been visual and subjective. Now, with various quantitative techniques readily available and easily applicable to computers, more rigorous and sophisticated map comparisons can be made (Merriam and Sneath 1966, p. 1105). For example, since derivative, residual, and continuation techniques are mathematically equivalent to linear filtering in two spatial dimensions, it is possible to make quantitative evaluations of the effects of two different filter coefficient sets without resorting to empirical tests on actual data.

However, in evaluating a contour map *per se*, the optimum approach is not a comparative one. To say something is "better" is not to say that it is good or even satisfactory. The contour procedure itself should be evaluated in terms of an input-output relationship.

A New Concept of Contouring

A contour map is often thought of as a projected and/or scaled replica of a physical surface. This is not exact. The map is an approximation created by

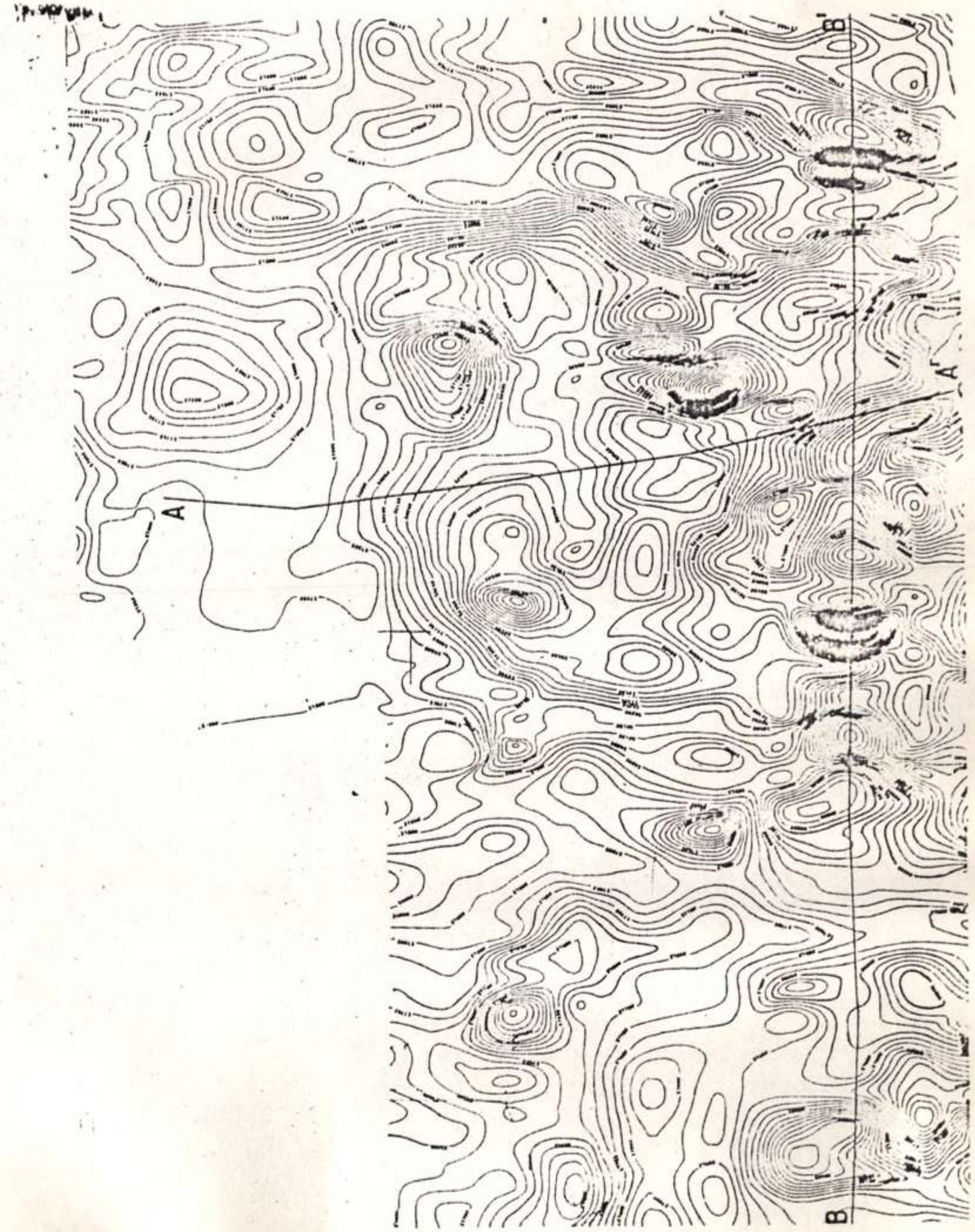


Fig. 4. Contour map output (program C_1)

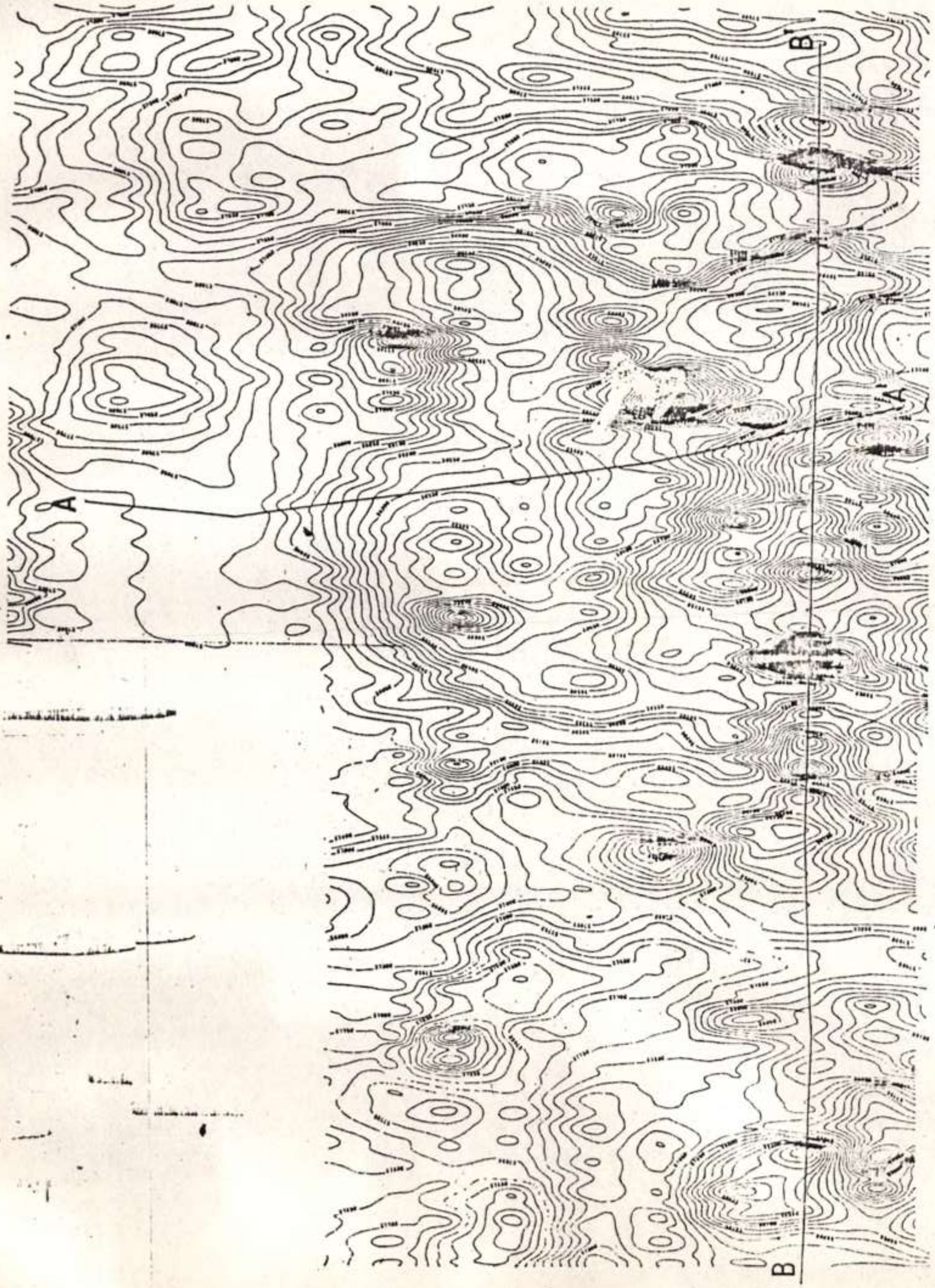


Fig. 5. Contour map output (program C_2)

Fig. 6. Contour map output (program D).

interpolation procedures from a limited set of information. The configuration of the map is therefore dependent on, first of all, the nature of the interpolation procedure and, secondly, the nature of the sampling grid. Since a given contour map represents, in most cases, an approximate or distorted version of an ideal, it can be said that the combined operation of sampling plus interpolation constitutes an act of filtering. For the present it shall be called "contour filtering".

Contouring as a Filter

Sheriff (1969, p. 261) has defined a filter as "that part of a system which discriminates against some of the information entering it". The definition given by Bracewell (1965, p. 179) is more general. He states that the term "filter" can be used to denote a system having an input and an output.

Despite having both an input and an output, contouring is not a filter in itself. The input and output are fundamentally different; the input is discrete, the output is continuous. Unless the input and output are of a similar nature, it is not meaningful to consider their relationship in terms of filter theory. This can be overcome if sampling and interpolation are considered as a unit. We then input a discrete set of data points and generate a continuous output in the form of a contour map. If the contour map is then re-sampled at the original sample input locations and if the re-sampling produces a set of points identical to the initial sampled set then the contour (interpolation) procedure has acted as an all pass filter, i.e. there is no distortion. If the converse is true, then the contouring is that part of the filter which has introduced the distortion or wavenumber discrimination.

To avoid casting doubt on the validity of such an approach it may be pointed out that geophysical data processing utilizes many procedures, which without always specifically being called so, are, in fact, filtering procedures. As discussed by Dean (1958, p. 97), operations such as upward continuations and second derivative determinations are equivalent to linear filtering in two spatial dimensions where the signal is regarded as a function of distance rather than time. Operators for such objectives as second derivatives may be analyzed and compared in the frequency domain and may be designed as band-pass, low pass or high pass filters. Also, the derivatives may be thought of as involving the convolution of data with a mathematical function or set of weights which constitutes a filter (Fuller 1967).

Description of the "Contour Filter"

It has been established that contouring may be considered in the context of a filter. One of the most significant features of a filter is that it is characterized by the relationship between its input and output, i.e. one need not be

concerned with the interior of the filter *per se*, only with its terminal properties (Papoulis 1962, pp. 81-168). The input and output may be related in several ways and, in the case of exploration maps, in either the space or wavenumber domains which are uniquely related through the Fourier transform.

The advantage of this approach is that it permits description of the "contour filter" in the more convenient language and terminology of the wavenumber domain. A second, and more important advantage is that the contour package may be subjected to the various standards of filter evaluation, such as amplitude and phase response.

The Evaluation

Introduction

In evaluating a contour package there are obvious advantages in utilizing both the space and wavenumber domains. Ideally, this evaluation would incorporate two-dimensional Fourier analysis. As this facility was available only for gridded data at the time of the analysis, it was found expedient to utilize an available one-dimensional Fourier analysis program to analyze the filtering effects of the various contour packages on selected profiles, rather than on the entire map.

Space Domain Comparisons

The first step in the analysis is to plot selected profiles of total intensity in gamma versus horizontal distance. The flight lines chosen are indicated by A-A′ and B-B′ on figure 1. It can be seen from the map of figure 3 that these two profiles best illustrate the varied magnetic relief of the area.

Four profiles were constructed for the north-south line A-A′. The first was plotted from the original flight line data and the second from the re-sampled data. The degree to which these match is of the highest importance because with re-sampling comes the possibility of aliasing, if the sampling interval is greater than the Nyquist interval. (aliasing may also be thought of as filtering since the aliased sample set will result in a frequency output different from the frequency input). The spatial sample interval of the original data is 61 m (200 feet) and that of the re-sampled data is 366 m (1200 feet). The original data are therefore definitive of a profile whose component wavelengths are greater than 122 m (400 feet), whereas the re-sampled data are definitive only of those profiles whose component wavelengths are greater than 732 m (2400 feet). Equivalently, the aliasing spatial frequencies are .00077 cycles/meter and .000126 cycles/meter respectively. However, as shown on figure 7 the re-sampled profile is a close fit to the original profile suggesting a predominance of low wavenumber features and reducing the concern for aliasing.

The third profile on figure 7 is constructed from the map of program A (figure 2) and the fourth from the map of program B (figure 3). The profile from map B offers an excellent comparison with the resampled input. However, the profile from map A bears only a vague relationship to the other three and appears as a moving average type of profile where the spatial amplitude spectrum has been considerably distorted since the contour package A has acted as a low-pass filter.

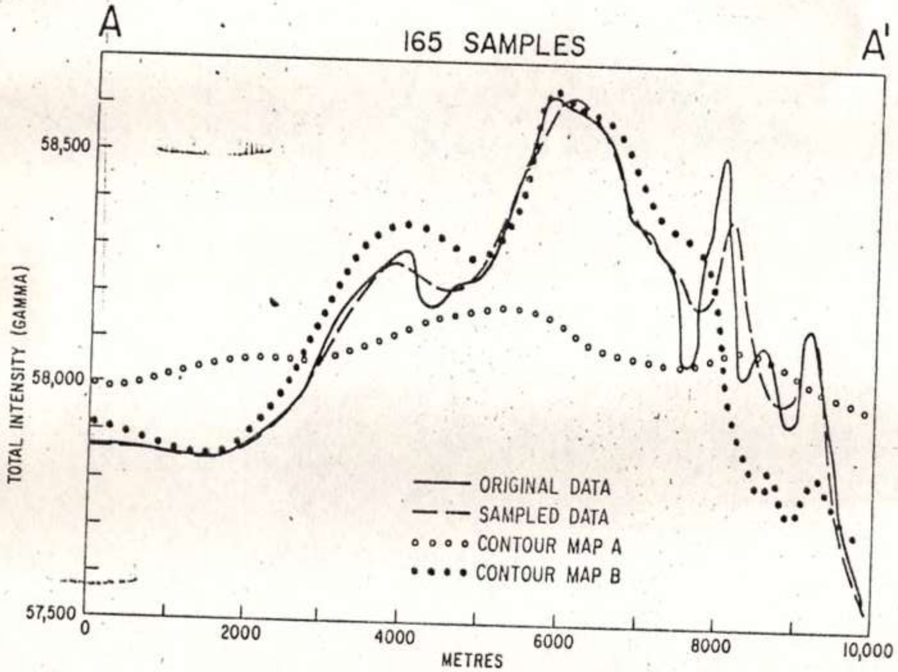


Fig. 7. Space domain profile comparisons (line A-A′).

It could be emphasized at this point that in interpreting magnetic data from the point of view of depth-to-basement determinations, the amplitudes and gradients of the flanks of an anomaly are the critical parameters. Consequently, if depth-to-basement calculations were made on map A, the calculated depths would be a factor of 4 times the true depth. The severe filtering imposed by package A has rendered the map useless for interpretation. The effects are more forcibly demonstrated in figure 8 where the magnetic relief is more eccentric. Figure 9 illustrates the comparisons of profiles B-B′ reconstructed from maps C_1, C_2 and D (figures 4-6 respectively). In each case it can be seen that the spatial amplitude spectrum of the input is well preserved by each package.

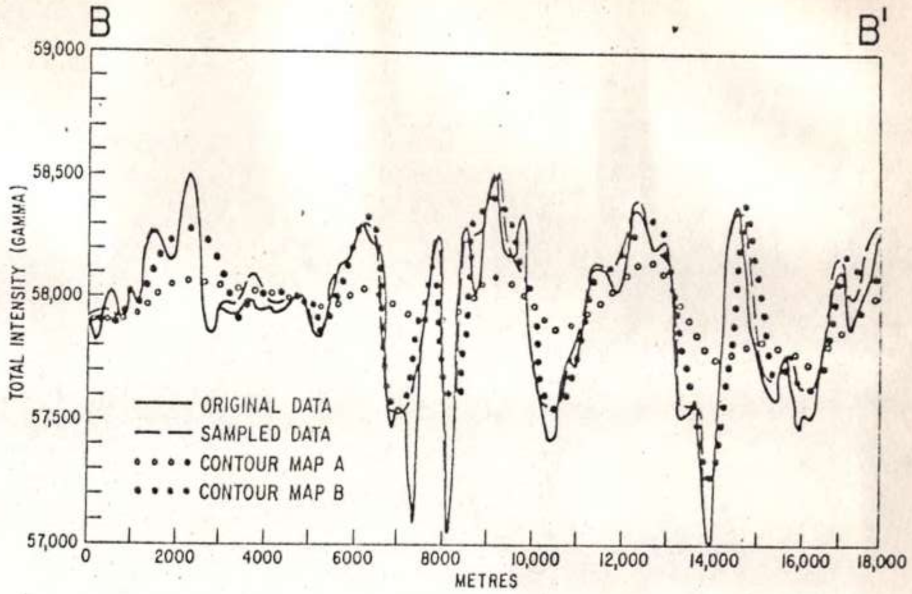


Fig. 8. Space domain profile comparisons (line B-B′).

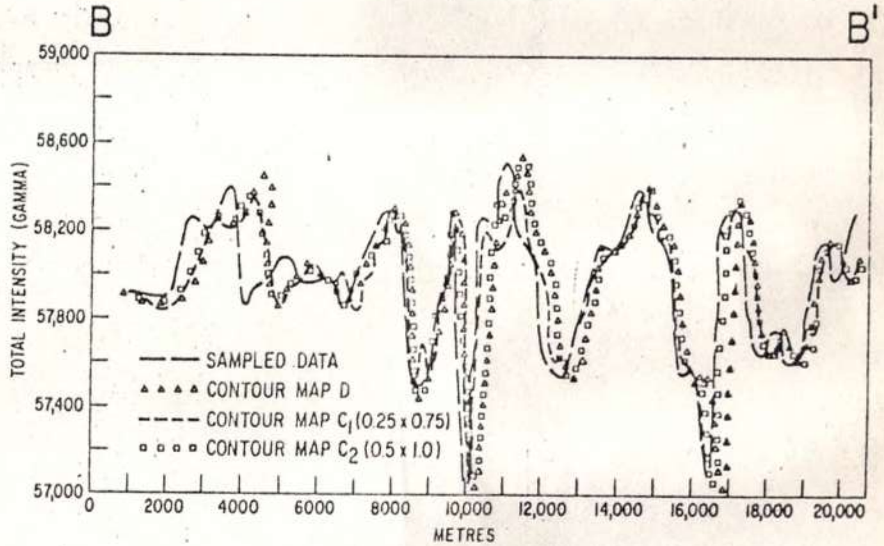


Fig. 9. Space domain profile comparisons (line B-B′).

Wavenumber Domain Analysis

The second step in the overall analysis is the determination of the various input and output wavenumber spectra. Accordingly, the analysis of the profiles was carried further by the application of a harmonic analysis program. Both amplitude and power spectra were produced for each profile. The program, unfortunately, did not have an option for phase spectra, thus these are not available. Their significance is therefore not appreciated at this time.

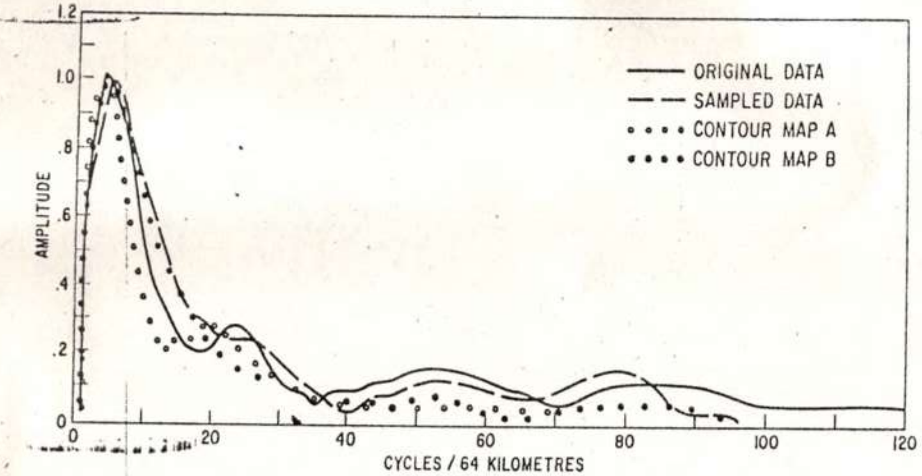


Fig. 10. Amplitude spectra (line A-A′).

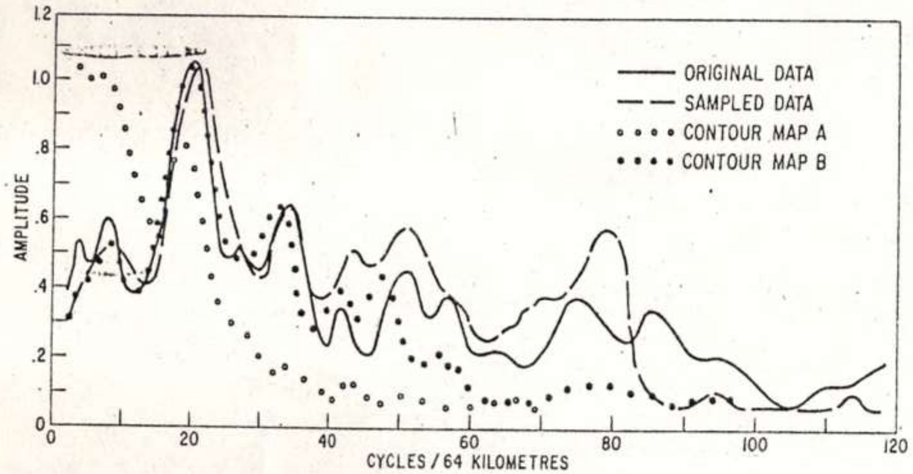


Fig. 11. Amplitude spectra (line B-B′).

Figure 10 shows the amplitude spectra for the original data, the re-sampled data and the profiles from maps A and B along line A-A′. It is therefore the wavenumber transform illustration of Figure 7. Figure 11 is the equivalent of Figure 8, along profile B-B′. Figure 12 is the amplitude spectrum equivalent of figure 9

along profile B-B′ from maps C_1, C_2 and D (figures 4-6). It is obvious that programs C_1, C_2 and D do an excellent job in maintaining the input wavenumber spectra. Any further considerations in the context of these data as to which is the "best" would involve an analysis of the phase-shift characteristics of each package, since it is difficult to separate them on the basis of their amplitude spectra alone.

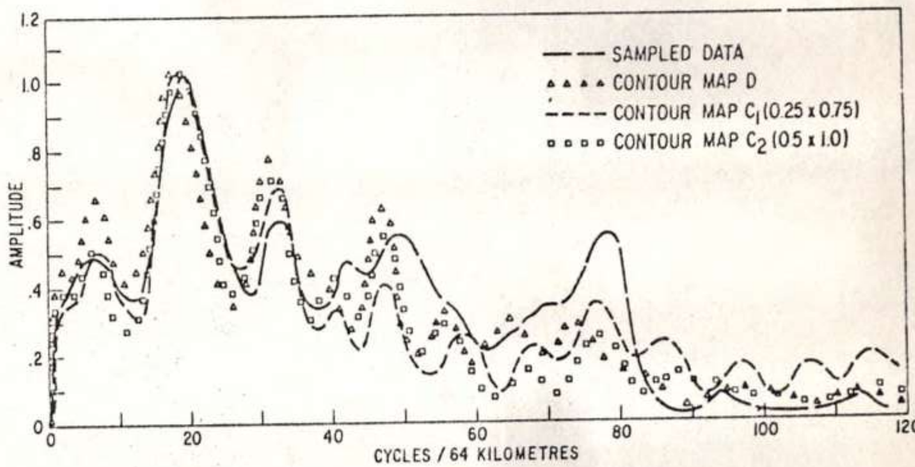


Fig. 12. Amplitude spectra (line B-B′).

Summary of the Analysis

This paper is in the nature of a reconnaissance. It sets out to emphasize more of a method than a result. Hence, the comparative aspects need not be considered in full detail. Visual inspection of map A (figure 2) inspires doubt as to its reliability. This leads to a comparison with map B (figure 3) which points out the differences but establishes nothing. An objective analysis can only be made by utilizing the input and output profiles in both the space and wavenumber domains. This analysis does not depend on having the data contoured by other packages; it is self-definitive, i.e. map A could be stamped inadequate on the basis of its amplitude characteristics in the space and wavenumber domains.

Conclusions

The quantitative comparison of a suite of contour maps produced from a common data set is a somewhat tentative objective. At the present time it is not possible to stipulate the ultimate criteria on which such comparisons might be made. However, the application of spectral analysis constitutes an objective approach, regardless how obvious it might seem in retrospect.

At this stage the following conclusions can be made:

1. For the first time a technique has been implemented which provides a quantitative assessment of the reliability of the map *in the context of the input data*.
2. It is obvious that the choice of input parameters is critical. For example, the difference in input cell size leads to the difference between maps C_1 and C_2 (figures 4, 5). The parameter determines the nature of the applied filter. Different cell dimensions will produce different filter responses. The use of the smaller cell in map C_2 has apparently forced more detail into the contours and stressed trends which are somewhat artificial.
3. The amplitude spectrum of the output data could be illustrated on the map alongside the amplitude spectrum of the input data. Ideally, these should be two-dimensional. If the frequency response of the program is then unacceptable, the input parameters can be redesigned and the map re-contoured.
4. Although this analysis has dealt with aeromagnetic data, it can be applied to contour maps generated from any data set.
5. At the present time, for most two-dimensional studies, it is a prerequisite that the data be input on a regular grid. Nonuniformly sampled data introduces the necessity for gridding. Although there are programs available which can produce gridded data from randomly distributed data, they are restricted to a fairly small matrix generation. Also, gridding and interpolation lead to filtering. Therefore, the application of these techniques prior to spectral analysis is discouraged. This stresses the need for more sophisticated programs which will perform two-dimensional harmonic analysis on irregularly spaced data directly. Such programs are now commercially available.
6. Contour filtering is, in most cases, low-pass filtering. This is due primarily to the discriminating properties of sampling.
7. Interpolation contouring is a linear filter where the wavenumber response depends on the initial data arrangement and interpolation interval in gridding.
8. A current problem in machine contouring is bias, or lack of it, in some certain direction. The only way to attack this is to perform a two-dimensional analysis. If the predominant strike is at some angle to the traverse lines, is one package preferable to another for maintaining the bias if desirable, or alternatively eliminating if undesirable?

Acknowledgements

I would like to express my thanks to Amoco Canada Petroleum Company Ltd. for permission to publish this paper. I also gratefully acknowledge the cooperation of two colleagues, Bob Smith with whom the initial study was undertaken and Eric Dahlberg for providing much stimulating discussion. Thanks are also extended to the Drafting Department at Amoco for the excellent quality of the diagrams.

References

BRACEWELL, R., 1965, The Fourier Transform and its Applications. McGraw Hill Book Co., New York.

DEAN, W. C., 1958, Frequency analysis for gravity and magnetic interpretation: Geophysics 23, 97-127.

FULLER, B. D., 1967, Two-dimensional frequency analysis and design of grid operators: Mining Geophysics, v. II, Theory, The Society of Exploration Geophysicists, 658-708.

MERRIAM, D. F. and SNEATH P. H. A., 1966, Quantitative comparison of contour maps: Jour. Geophys. Res. 71, 1105-1115.

PAPOULIS, A., 1962, The Fourier Integral and its Applications: McGraw-Hill Book Co., New York.

SHERIFF, R. E., 1969, Addendum to glossary of terms used in geophysical exploration: Geophysics 34, 255-270.

WALTERS, R. F., 1969, Contouring by machine: a user's guide: A.A.P.G. Bulletin 53, 2324-2340.

METHODS FOR CONTOURING IRREGULARLY SPACED DATA*

BY

G. BOLONDI**, F. ROCCA*** and S. ZANOLETTI**

ABSTRACT

BOLONDI, G., ROCCA, F., and ZANOLETTI, S., 1977, Methods for Contouring Irregularly Spaced Data, Geophysical Prospecting 25, 96-119.

The sampling theorem in two dimensions univocally defines a surface, provided that its values are known at points disposed on a regular lattice. If the data are irregularly spaced, the usual procedure is first to interpolate the surface on a regular grid and then to contour the interpolated data: however, the resulting surface will not necessarily assume the prescribed values on the irregular grid.

One way to obtain this result is to introduce a transformation of the coordinates such that all the original data points are transferred into part of the nodes of a regular grid. The surface is then interpolated in the points correspondent to the other crosspoints of the regular grid; the contour lines are determined in the transformed plane and then, using the inverse coordinate transformation, are transferred back to the original plane where they will certainly be congruent with the original data points.

Nonetheless, the resulting surface is very sensitive to the interpolation method used: two algorithms for that are analyzed. The first (harmonization) corresponds to the determination of the potential of an electrical field whose contour conditions are those defined by the data points. The second method consists in two dimensional statistical estimation (krigeing); in particular, the effects of different choices for the data autocovariance function are discussed.

The solutions are compared and some practical results are shown.

1. INTRODUCTION

In this paper a method for contouring irregularly spaced data will be analyzed. This problem arises often in geophysical exploration; in gravimetrics, the locations where the gravimetric anomaly has been measured are randomly scattered throughout the area to be examined; in aeromagnetics, the data are aligned on an irregular grid of the lines of flight; in seismics, the data to be contoured may derive from seismic lines and/or wells, again scattered in the plane and randomly placed, representing depths, times or velocities, and their uncertainty will be variable.

* Paper presented at 37th Meeting of the European Association of Exploration Geophysicists, June 1975, Bergen, Norway.

** AGIP — Attività Minerarie, S. Donato Milanese, Milano, Italy.

*** Politecnico di Milano, Milano, Italy.

In all cases the problem to be solved is to guarantee that the resulting surface is smooth enough, without fictitious structures, and also that the data points are honoured, exactly or approximately, depending on their nature.

The surface discussed in this paper, however, will pass strictly through the data points.

This problem has already been studied by several authors (Crain 1972, Merrill 1973); the difficulties encountered will be analyzed and the structure of a method for their solution will be outlined.

Since the problem is easily solved when the data are disposed on a regular grid, the method used will be simply that of introducing a transformation in the $x - y$ plane such that the data in the transformed plane $u - v$ have integer coordinates in the interval

$$0 < u \leqslant L, \qquad 0 < v \leqslant M.$$

In general, the number N of data points will be smaller than $L \times M$ so that to determine completely the surface in the plane $u - v$ and therefore in the plane $x - y$ their values at the other crosspoints of the grid have to be interpolated.

Two algorithms will be given for that purpose: one ideally connected to "krigeing", i.e., two dimensional estimation based on the stochastic properties of the data (Merriam 1973), the other serving to determine the electrical potential of a field whose contour conditions are those defined by the data points.

Once the surface is defined at all the crosspoints of the grid the definition is extended to all points of the plane using bicubic splines (Ahlberg, Nilson, and Walsh 1967).

A simple method for the approximate determination of the contour lines is also given. Some examples will follow.

2. GENERALITIES

Let us consider a surface regularly sampled on a grid with interval Δx and Δy in the x and y directions, respectively,

For the sampling theorem, the spectrum of the sampled surface will correspond to that of the original surface $S(f_x, f_y)$ (where f_x, f_y are spatial frequencies or wave numbers in the x and y directions, respectively), periodically repeated with centers in the points having the coordinates (see fig. 1)

$$f_x = \frac{1}{2\Delta x} \qquad f_y = \frac{1}{2\Delta y}. \tag{1}$$

If $S(f_x, f_y)$ is contained in the shaded lozenge, then we can interpolate the surface from the sampled data without errors, provided that we use filters whose Fourier transforms are (Papoulis 1968)

$$H(f_x, f_y) = 1 \qquad \text{in the lozenge}$$
$$H(f_x, f_y) = 0 \qquad \text{elsewhere} \qquad (2)$$

This means that if the highest spatial frequencies contained in the signal are low enough, a complete reconstruction is possible. This reconstruction is obtained by means of the convolution of the sampled signal with the pulse response of the filter, i.e., the Fourier anti-transform of $H(f_x, f_y)$. This pulse response decreases to zero practically in a few sample intervals.

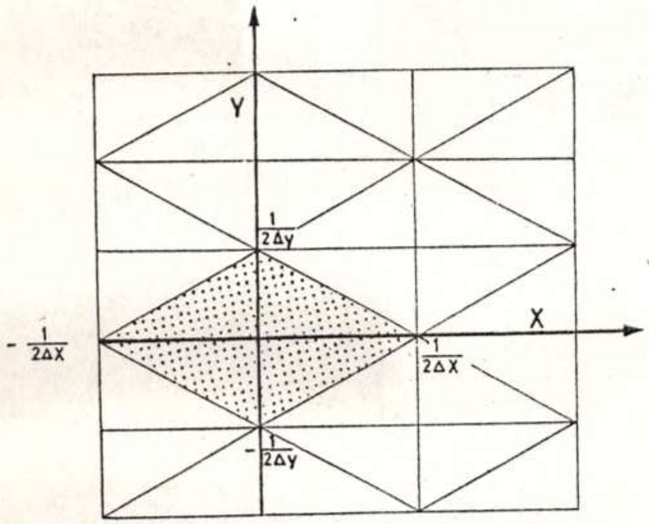


Fig. 1. Fourier Transform of a sampled surface.

This means that each point of the continuous surface is interpolated from the neighbouring points (Papoulis 1968).

In a later section we will see that it is possible to obtain a good smoothed approximation of this surface without actually using the afore-mentioned complicated interpolation procedure; it will be enough to approximate the surface by reducing it to quadrilateral patches whose vertices are the cross-points of the grid; the values of the surface inside the patches will be determined by means of bicubic spline interpolation.

On the other hand when the data are irregularly spaced, the sampling theorem does not apply; therefore, we cannot have a univocally determined surface passing through the data points.

An immediate consequence is that any interpolation method chosen has many elements that can be fixed arbitrarily so that the final choice between two "reasonable" methods has to rely on ill defined "qualities" of the resulting contour map.

In principle, the simplest method that can be adopted in the random spacing case is the one of reducing it to regular spacing, interpolating in some way the data on a regular grid. We are then reduced to the case previously analyzed.

However, noticeable difficulties appear if we try to apply this method. In fact, no interpolating method has been found, up to now, capable of ensuring that the resulting surface will pass through the irregularly spaced data. Obviously the difference in level will not be very large if the grid interval is small enough, but nonetheless differences will appear.

Therefore, a second approximation is necessary in which the data on the regular grid are corrected so that the surface will behave as desired.

Thus we see that, unless we accept some inconsistency between the original data and the interpolating surface, the method that appeared so simple may be expensive in terms of computing time, or even impossible, if the grid interval is too large.

Another method, apparently more complicated, but in fact simpler, proceeds as follows (Hessing, Lee, Pierce, and Powers 1972).

A bi-univocal correspondence between the $x - y$ plane and another plane $u - v$ is found

$$x = X(u, v) \qquad u = U(x, y)$$
$$y = Y(u, v) \qquad v = V(x, y) \qquad (3)$$

such that the data points will correspond to points having integer coordinates in the $u - v$ plane. Once this correspondence has been established we are again reduced to the regular case provided that we interpolate the surface at the points corresponding to the other crosspoints of the regular grid in the $u - v$ plane.

After interpolation we get the surface in the $u - v$ plane and applying the inverse transformation (3) we obtain the surface in the $x - y$ plane.

In conclusion, the problems to be solved are, in succession:

a) finding the transformation (3) and its inverse;
b) interpolating the surface $z(x, y)$ at all the points corresponding to the crosspoints of the regular grid in the $u - v$ plane.

3. Definition of the Correspondence Between u, v and x, y

To establish the correspondence (3) it is first necessary to define the points in the $x - y$ plane corresponding to all the crosspoints of the regular grid. To accomplish this the data are divided into $(L + M)$ sets of points corresponding to the L rows and M columns of the regular grid. The subdivision is univocally determined if the two following rules are observed:

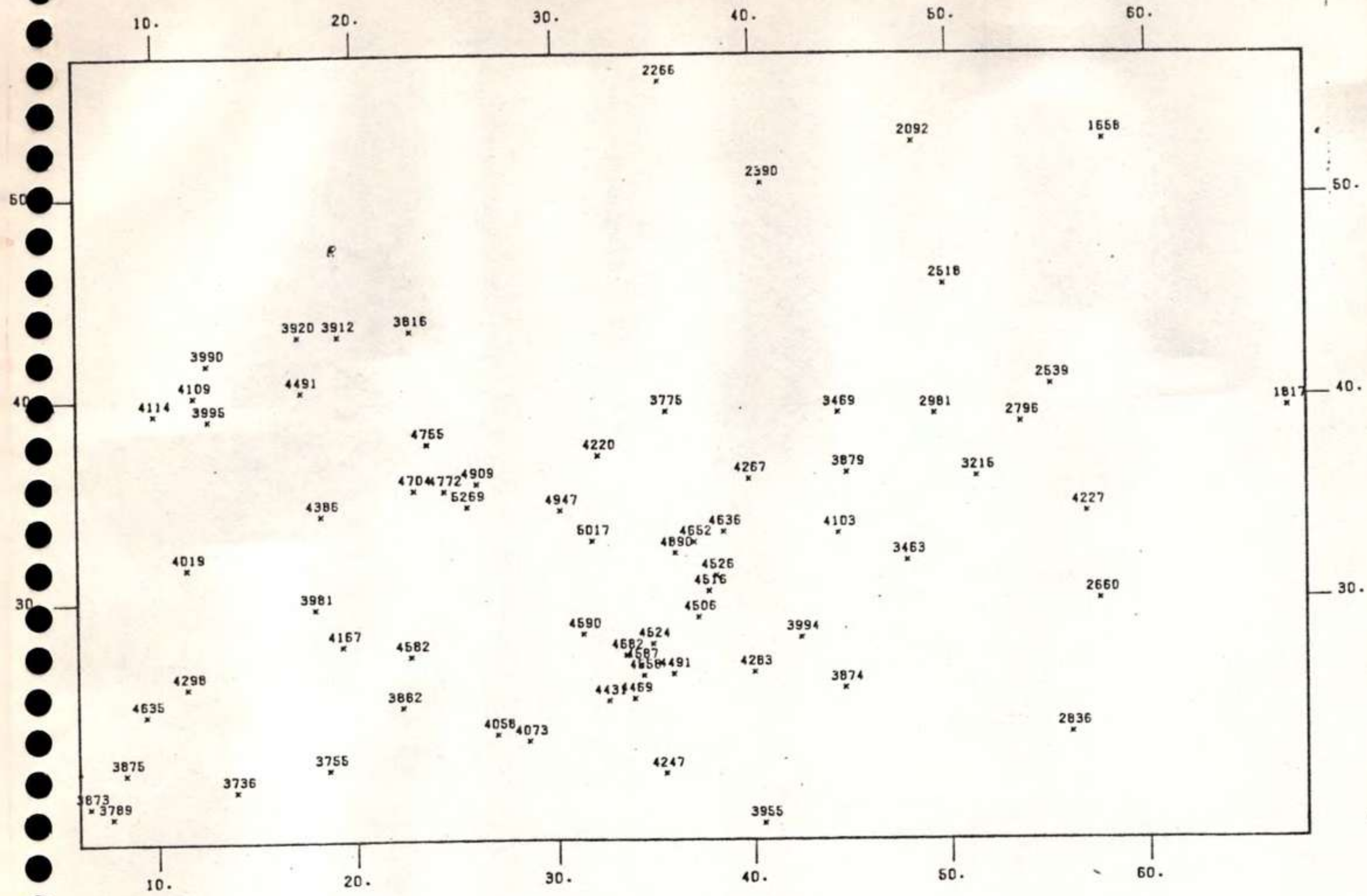


Fig. 2. Data to be contoured.

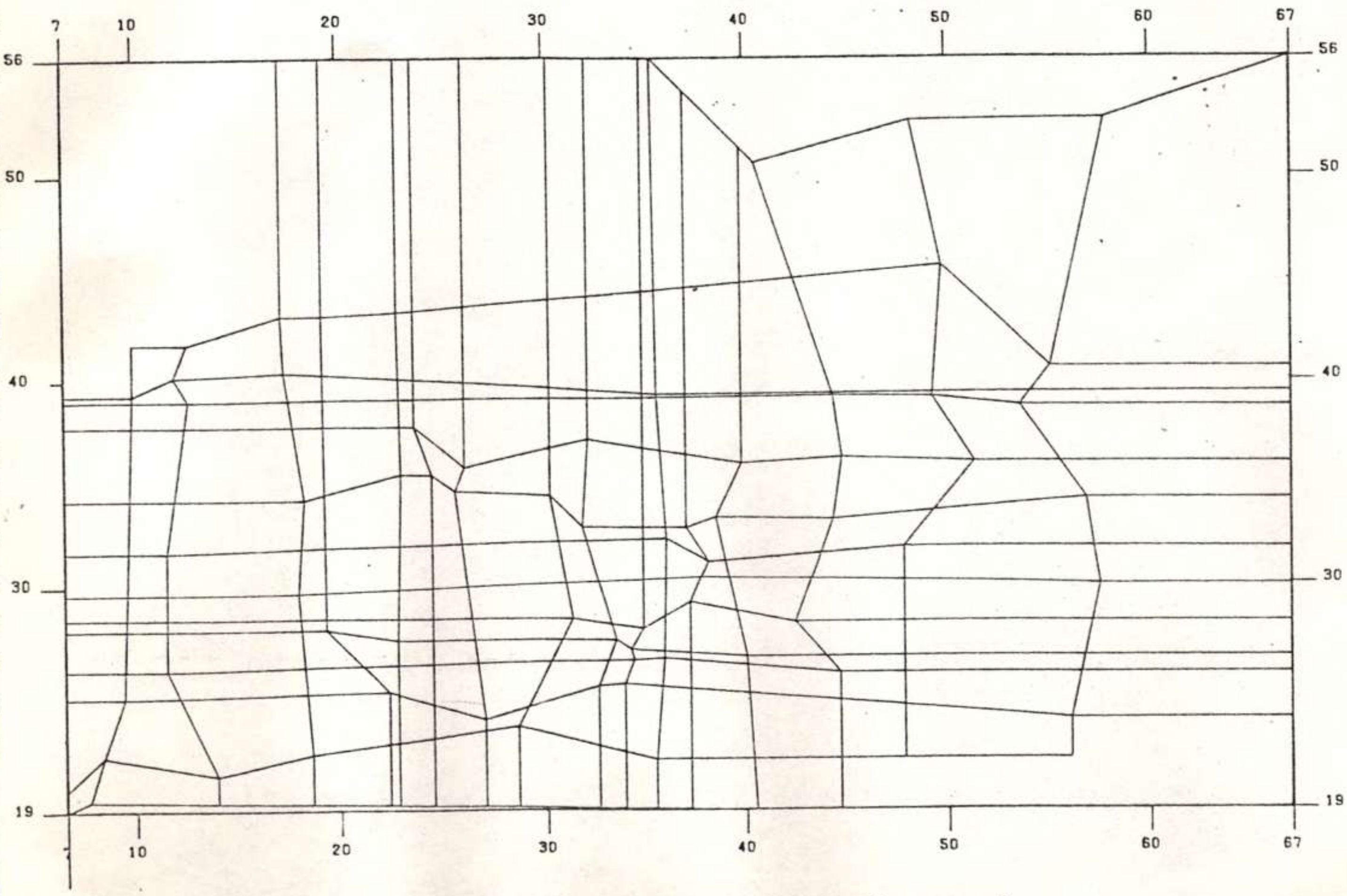


Fig. 3. Irregular grid connecting the data points to be contoured.

a) all points contained in the set corresponding to one row will have their ordinates smaller (larger) than the ordinates of the elements of the set correspondent to the next (preceding) row. The same can be said for the columns.

b) No two points may belong to the set corresponding to the same row (or column) if the straight line connecting them makes an angle greater than $\pi/4$ with the horizontal (or vertical).

Once these rules have been applied, L and M are determined and each set corresponding to a row or to a column of the grid will contain at least one datum point.

Abscissae $x_{i,k}$ and ordinates $y_{i,k}$, of the points in the $x-y$ plane correspondent to the other crosspoints of the regular grid are independently determined using linear interpolation between abscissae and ordinates of the neighbouring data points respectively (see fig. 2 and 3).

At this stage we know that

$$X(i, k) = x_{i,k}$$
$$Y(i, k) = y_{i,k}. \tag{4}$$

The transformation is extended to the points having non-integer coordinates using bi-cubic spline interpolation (Ahlberg et al 1967).

4. Interpolation of Missing Grid Crosspoints Through Harmonization

As was observed before, the task of interpolating the surface at the other crosspoints of the grid may be accomplished in several ways: we will first discuss the case in which we will take as a prime requisite that the surface be the smoothest possible, compatibly with the given data points.

We can then think of the surface as the electric potential of a field having boundary conditions corresponding to the data points.

Should this be the case, then the surface would have to satisfy the harmonicity condition, i.e.

$$\nabla^2 z(x, y) = 0 \qquad x, y \neq x_{i,k},\ y_{i,k}. \tag{5}$$

This condition can be easily converted to the corresponding one, valid for the finite difference case, using only the values of the surface at the crosspoints of the irregular grid in the $x-y$ plane.

The relation that has to be satisfied is approximately

$$z_0 = \frac{\Sigma p_i \cdot z_i}{\Sigma p_i} \quad \text{with } p_i = \left(\frac{r_0}{r_i}\right)^2 \tag{6}$$

where

z_0 = the surface value of the point to be interpolated;
z_i = the surface values of the four given points surrounding the points to be interpolated;
r_0 = unit distance;
r_i = distance between z_i and z_0.

(Obviously this formula is not applied to the data points).

We start from a trial solution and then iterate the algorithm

$$z_0^{(m+1)} = \tfrac{1}{2}\left[z_0^{(m)} + \frac{\Sigma p_i \cdot z_i^{(m)}}{\Sigma p_i}\right] \tag{7}$$

until the variations are small enough.

This method may appear rather odd, but in fact it is none other than smoothing the surface (or at least some of its points) leaving the data points unmodified. The smoothing operation is continued until an asymptotic situation is reached. To guarantee the stability of the method it is proper to refer to the well known algorithms in use for the determination of the electric potential fields.

Instead of referring to the harmonicity equation one could have referred to the biharmonic equation, i.e. to the displacement of an elastic plate of a given rigidity, constrained to pass through the data points; however, this could create fictitious structures.

The reason for which we referred to the harmonic equation is simply the fact that the smoothing algorithm is much cheaper to apply, but there are no valid reasons not to refer to other elliptic partial differential equations.

In conclusion, this method of interpolation has no particular meaning except that of giving results very similar to those that could be found in "hand made" maps.

5. Interpolation of Missing Grid Crosspoints Through a Two-Dimensional Estimation

To state it crudely, the essence of the interpolation method analyzed in the previous section corresponds to affirming that the surface is the smoothest possible where we do not know its value.

Another interpolation method, having a completely different philosophy, is the one corresponding to two-dimensional estimation.

In fact a more relevant approach could be the one of saying that the surface "behaves" in the same way in the known and in the unknown regions. If now we want to give a quantitative meaning to this idea, then we must introduce the concepts of stochastic fields (Papoulis 1968) uniform in the two directions.

We will suppose now that $z(x, y)$ is a random surface of which we can determine the covariance function

$$E\{z[(x+\xi), (y+\eta)] \cdot z[x, y]\} = R(\xi, \eta) \tag{8}$$

where E is the ensemble mean operator and ξ and η are the components along the two axes of the distance between the points. We are interested in the cases in which z behaves similarly everywhere, and therefore the covariance function $R(x, y, \xi, \eta)$ depends only on ξ, η and not on x, y.

The problem is therefore the one of assigning a value to $R(\xi, \eta)$: this can be done by exploiting the data, finding their cross-covariance using spatial averaging, and then fitting some parameter of a two-dimensional function.

For example, we could assign to $R(\xi, \eta)$ an exponential form

$$R(\xi, \eta) = e^{-\alpha|\xi| - \beta|\eta|}. \tag{9}$$

and fit the parameters α, β to the cross-covariance of the data in the x, y directions, respectively.

Other forms of $R(\xi, \eta)$ might be

$$\begin{cases} R(\xi, \eta) = 1 - \dfrac{\sqrt{\xi^2 + \eta^2}}{\xi_0}, & \sqrt{\xi^2 + \eta^2} < \xi_0 \\ R(\xi, \eta) = 0 & \text{elsewhere;} \end{cases} \tag{10-a}$$

$$R(\xi, \eta) = 1 \Big/ \left[1 + \frac{\xi^2 + \eta^2}{\xi_0^2}\right]^{3/2}, \quad \text{or} \tag{10-b}$$

$$R(\xi, \eta) = e^{-\frac{\xi^2 + \eta^2}{\xi_0^2}}. \tag{10-c}$$

In the first case, $R(\xi, \eta)$ is anisotropical; in the other cases the behaviour in the two directions is identical.

In the appendix the effects of the choice of a given covariance function will be analyzed in more detail.

Let us suppose now that this problem is solved: then the problem of finding the optimal linear interpolator of the surface at the point $z(x_0, y_0)$ in the mean square sense is easily solved.

In fact we know that the estimator $\hat{z}(x_0, y_0)$ is

$$\hat{z}(x_0, y_0) = \sum_{m=1}^{N} \alpha_m(x_0, y_0) \cdot z_m(x_m, y_m), \tag{11}$$

since we seek a linear interpolator. The weights α_m are found from the condition that

$$E[\hat{z}_0 - z_0]^2 = \varepsilon^2, \tag{12}$$

be minimal. Developing the square and applying relation (8) we see that the search for the unconditioned minimum leads to the so called *normal* (*Wiener-Hopf*) *equations*

$$\Sigma\, \alpha_m \cdot r_{m,n} = r_n \qquad n = 1, \ldots N \tag{13}$$

where

$$\begin{aligned} r_{m,n} &= R(x_m - x_n, \quad y_m - y_n) \\ r_n &= R(x_0 - x_n, \quad y_0 - y_n) \end{aligned} \tag{14}$$

and x_m, y_m is the location of datum point z_m. This interpolating procedure is optimal in the case that $z(x, y)$ is a spatially stationary, zero mean Gaussian variate.

6. Interpolation in the Presence of a Trend

If a known trend is superimposed on the data, the best thing to do is to remove it, to apply to the residuals this interpolation method and then to replace the trend. If it is not known, a reasonable approach might be the one of subtracting a low order two-dimensional polynomial, such that the residuals will appear stationary. There is no general rule for this operation, at least not until one has specified what is intended by trend, as well as conditions that may ensure that the trend has been properly removed.

Another approach is that of imposing that the coefficients of the interpolator be "transparent" to the trend. In other words let the trend be expressed by

$$f(x, y) = \sum_{i=1}^{q} t_i \cdot f_i(x, y) \tag{15}$$

where the t_i are coefficients and $f_i(x, y)$ are a set of q given smooth functions of x, y. Then we have to impose the following set of conditions on the N interpolation weights

$$\sum_{m=1}^{N} \alpha_m \cdot f_i(x_m, y_m) = f_i(x_0, y_0), \qquad i = 1, \ldots q. \tag{16}$$

These conditions correspond to the imposition that the result of the interpolation be independent of the chosen trend.

In the simplest case, we can pose $q = 1$, $f(x, y) = 1$, and conditions (16) reduces to

$$\sum_{m=1}^{N} \alpha_m = 1 \tag{17}$$

In this case the interpolation result is independent of a possible constant that may be added to all values of the surface.

Applying (17) we are in a case of bounded minimum, that can be solved using Lagrangian multipliers.

The aspect of the interpolated surface changes depending on whether condition (17) is applied or not: in the latter case, in fact, the weights (and therefore their sum) will tend to zero if the distance of the point to be interpolated from the other points is large enough. This may contribute to give an aspect to the surface of being "peaked" around the data points.

7. Comparison Between the Two Methods of Interpolation

The most noticeable difference between the two methods discussed in the previous sections is that using the "harmonization" procedure the interpolated values never exceed the range of the data; maxima and minima of the surface are data points. This is peculiar to harmonic functions where maxima are always at the boundaries. The interpolation procedure corresponding to two-dimensional estimation leads to interpolated values whose range may greatly exceed that of the data, depending on the covariance function chosen, as will better be seen in the appendix. In general, this algorithm will tend to add new features to the surface extrapolating local trends.

Another interesting feature of the "harmonization" procedure is the smoothness of the resulting surface. Application of the algorithm (11) in general creates "noisy" surfaces, and more so if the set of data points for the interpolation is changed from one point to the other.

This may happen when N is very large and the solution of the system (13) becomes clumsy. In this case, one could exploit the fact that if a point P_1 is between the point to be interpolated P_0 and another point P_2, then the weight to be given to the value of the surface at P_2 is much smaller than that corresponding to the valued at P_1. Then to get a good approximation of the solution of the set (13), it is enough to use for the interpolation the 8 or 16 points (one or two for each octant) nearer to P_0. This indeed simplifies the search for the solution, but may add disturbances to the results, since the set of points used as the basis for the interpolation changes from one point to the other.

Good results may be obtained using the two-dimensional estimation as a basis for the harmonization. After the first interpolation using (11) the iterative algorithm (7) is used until the noise has been smoothed out. If the iterations are continued, all the structures generated by the extrapolation of the local trends are flattened and a complete smoothing is obtained.

By properly choosing the number of iterations a good compromise between the two approaches can be found.

8. Bi-Cubic Spline Interpolation

Up to now we have determined how to interpolate the functions at all the crosspoints of the irregular grid. To determine the function for all the values of x, y we turn to bicubic spline interpolation.

We know that

$$\begin{aligned} X(u, v) &= x_{i,k} \\ Y(u, v) &= y_{i,k} \qquad\qquad u = i = 1, \ldots L \\ Z(u, v) &= z_{i,k} \qquad\qquad v = k = 1, \ldots M. \end{aligned} \tag{18}$$

We will first determine the functions $X(u, v)$, $Y(u, v)$ for non integer values of u, v. To determine the bicubic splines we need the derivatives

$$\frac{\partial X}{\partial u}, \ \frac{\partial Y}{\partial u}, \ \frac{\partial X}{\partial v}, \ \frac{\partial Y}{\partial v}.$$

These are found as follows: the parabolas with vertical axes passing through the points $(x_{i-1,k}, y_{i-1,k})$, $(x_{i,k}, y_{i,k})$, $(x_{i+1,k}, y_{i+1,k})$ are found and their tangents at the point $(x_{i,k}, y_{i,k})$ are determined. This value is assigned to

$$t = \frac{\partial X}{\partial u} \Big/ \frac{\partial Y}{\partial u} \qquad \begin{aligned} u &= i \\ v &= k \end{aligned} \tag{19}$$

The actual values of the derivatives are found by multiplying t times the length of the minor of the two sides linking the point with the neighbouring ones in the u direction. The same is done for the $v =$ *constant* links.

Once the transformations $X(u, v)$, $Y(u, v)$ are completely determined the two derivatives $\partial Z/\partial u$, $\partial Z/\partial v$ have to be found. This could be done using the two-dimensional estimation algorithm, but the result is too noisy. The tangent plane to the surface at the point P_0 is therefore defined to be the plane passing through P_0 best fitting the four adjacent points. Once $\partial Z/\partial x$, $\partial Z/\partial y$ are found, the derivatives are determined with the following formula:

$$\begin{aligned} \frac{\partial Z}{\partial u} &= \frac{\partial Z}{\partial x} \cdot \frac{\partial X}{\partial u} + \frac{\partial Z}{\partial y} \cdot \frac{\partial Y}{\partial u} \\ \frac{\partial Z}{\partial v} &= \frac{\partial Z}{\partial x} \cdot \frac{\partial X}{\partial v} + \frac{\partial Z}{\partial v} \cdot \frac{\partial Y}{\partial v} \end{aligned} \tag{20}$$

9. The Contouring

Once the surface has been defined, its level curves remain to be determined. This is accomplished by first finding their intersections with the irregular grid in the $x - y$ plane, or with the regular grid in the $u - v$ plane. These intersections have now to be interconnected: the tangents to the interconnecting curve are already known, at least at the points of intersection previously found.

If the curve intersects the patch at only two points then it may be enough to approximate it with a cubic, defined with end-points and end-tangents. If the intersecting points are more than two, say four or more, information has to be added to determine which point has to be connected with which other.

In general, this could be accomplished only by decomposing the patch into smaller surfaces and determining the intersections of the curve with the new boundaries. A simple method, but effective enough, consists in adopting the interconnection that corresponds to the minimal total tensile strain of the splines.

10. Enrichment of the Lattice

With the algorithm described in section 3, the lattice may be too uneven and the interpolation may show signs of that. It is useful therefore to regularize the grid adding new rows and columns, so that the maximum interval between subsequent rows or columns is limited to a preset value. All the remainder of the contouring is left unchanged.

11. Practical Examples

The contouring of the data in fig. 2 are shown in order to give an idea of the results of the application of the algorithms previously derived. If the rules in section 3 are followed, the irregular grid is that in fig. 3. The contouring of the data using two-dimensional estimation is illustrated in fig. 4. The covariance used is that of formula (10-b); the distance ξ_0 varies to adapt to the local structure of the data. The result of twenty iterations of algorithm (7) is shown in fig. 5.

The anomaly at abscissa 23 is reduced, the trough at right is flattened, and the 5000 line is split. Also the contour lines are smoother.

If the lattice is enriched as in fig. 6, the contouring without harmonization is that in fig. 7. The surface appears more noisy and with more structures: see, for example, the 4000 line at the left. After twenty harmonization steps (fig. 8), the noise is much smaller.

If the lattice is enriched further (fig. 9), the surface becomes very noisy if no harmonization is applied (fig. 10). On the other hand, with twenty iterations we obtain the best results (fig. 11). The structure at the right now is much less elongated due to the higher regularity of the lattice.

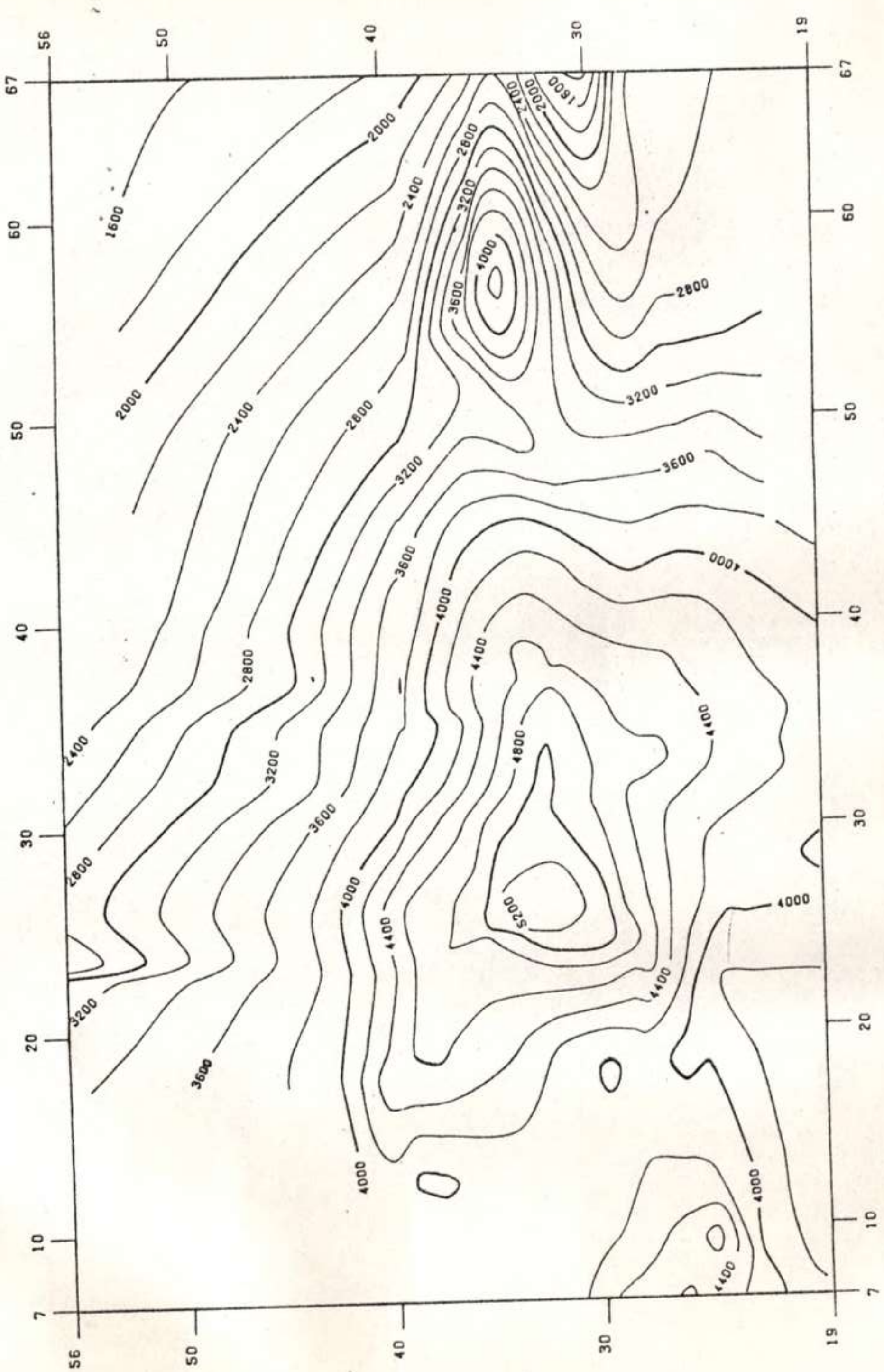


Fig. 4. Data contouring after interpolation on the grid of fig. 3 using two-dimensional estimation followed by twenty harmonizing iterations.

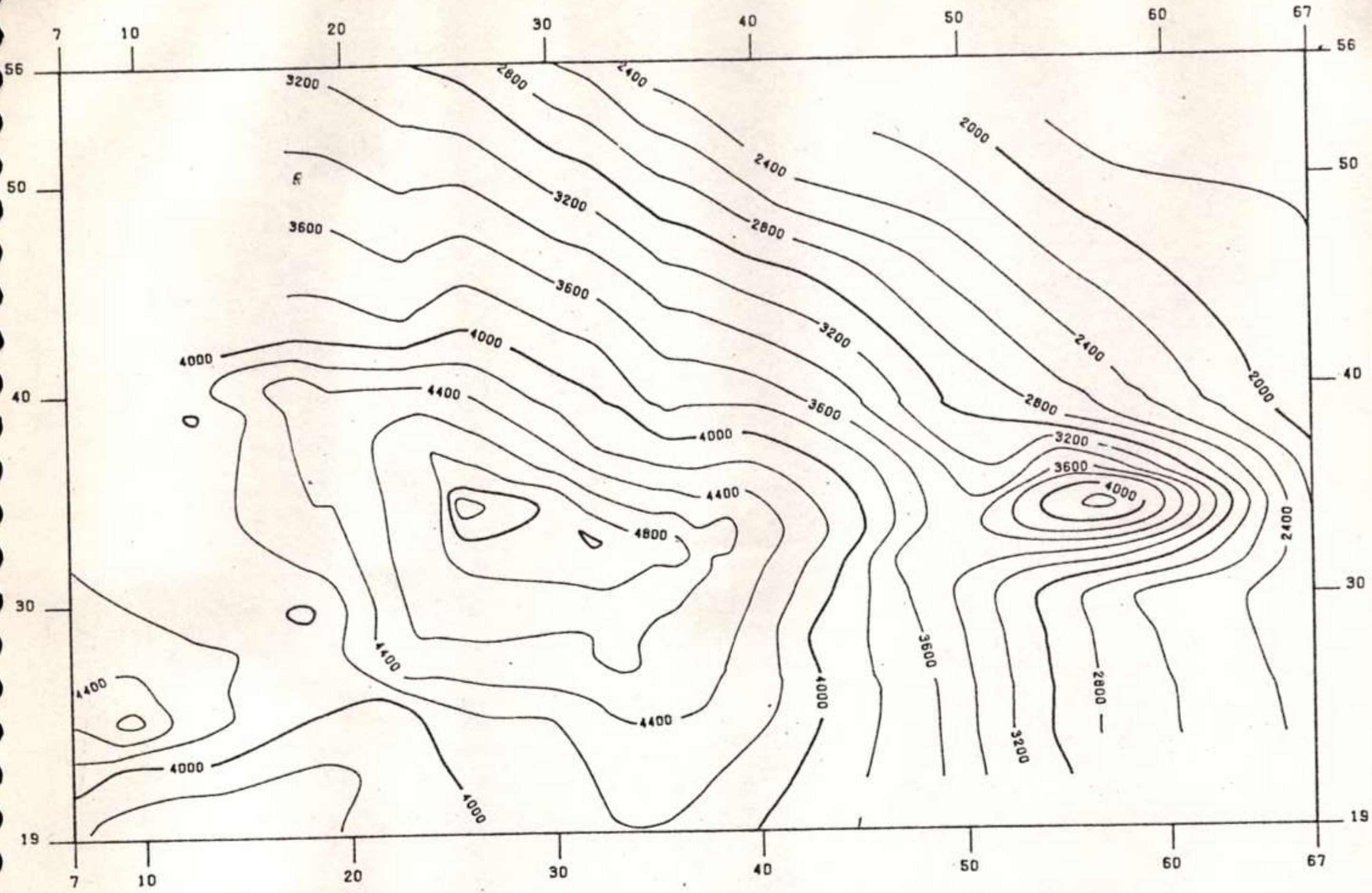


Fig. 5. Interpolation on the grid of fig. 3 using two-dimensional estimation.

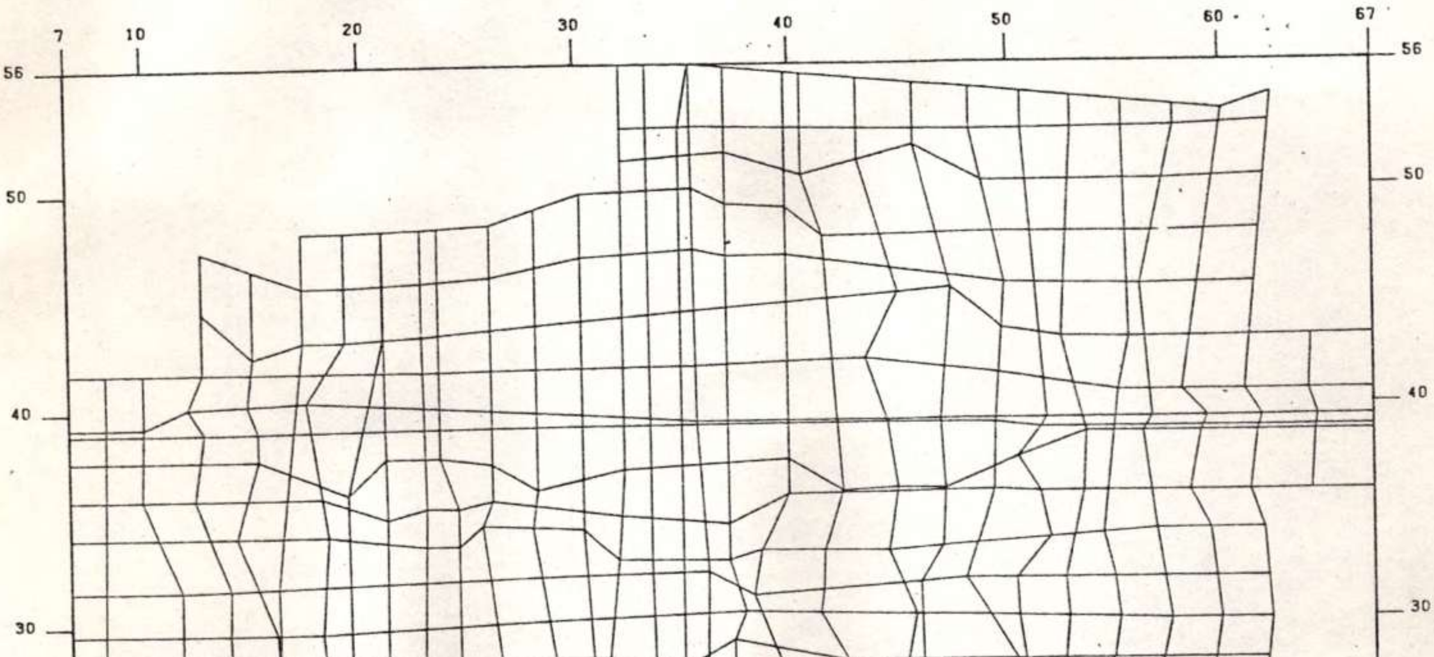


Fig. 6. Irregular grid with increased number of rows and columns.

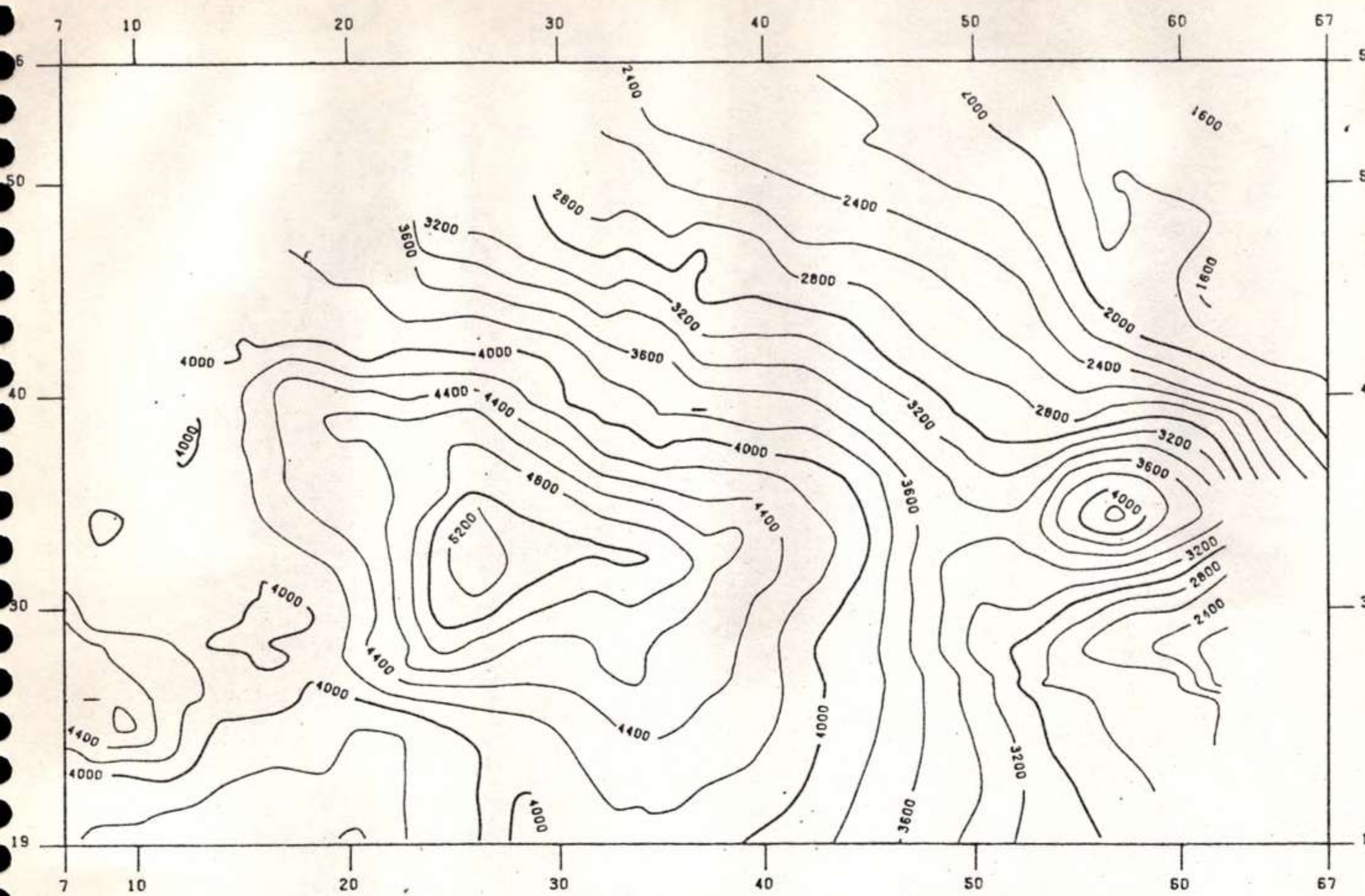


Fig. 7. Data contouring after interpolation of the grid of fig. 6 using two-dimensional estimation.

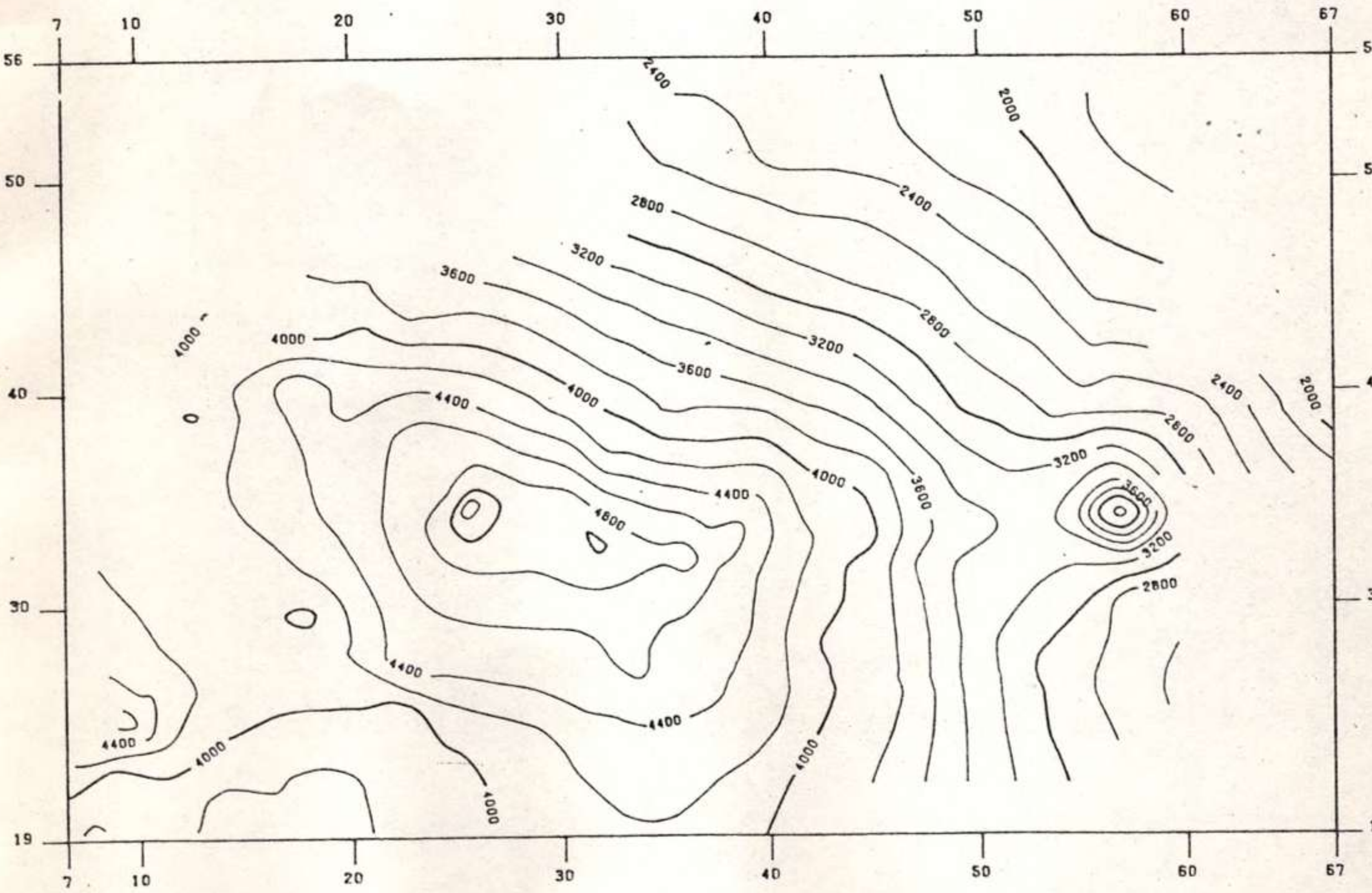


Fig. 8. Data contouring after interpolation on the grid of fig. 6 using two-dimensional estimation followed by twenty harmonizing iterations.

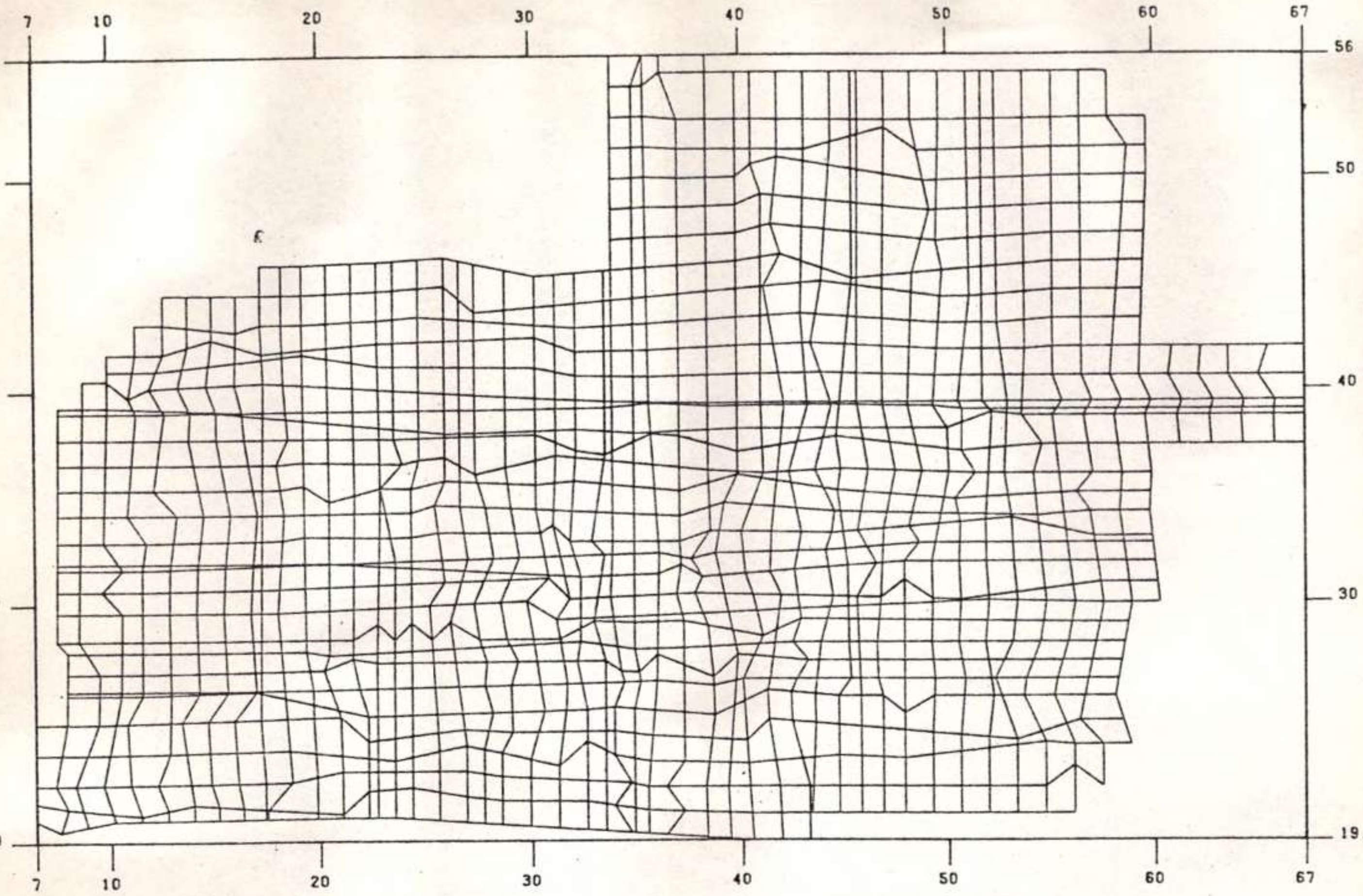


Fig. 9. Irregular grid with rows and columns increased with respect to the grid of fig. 6.

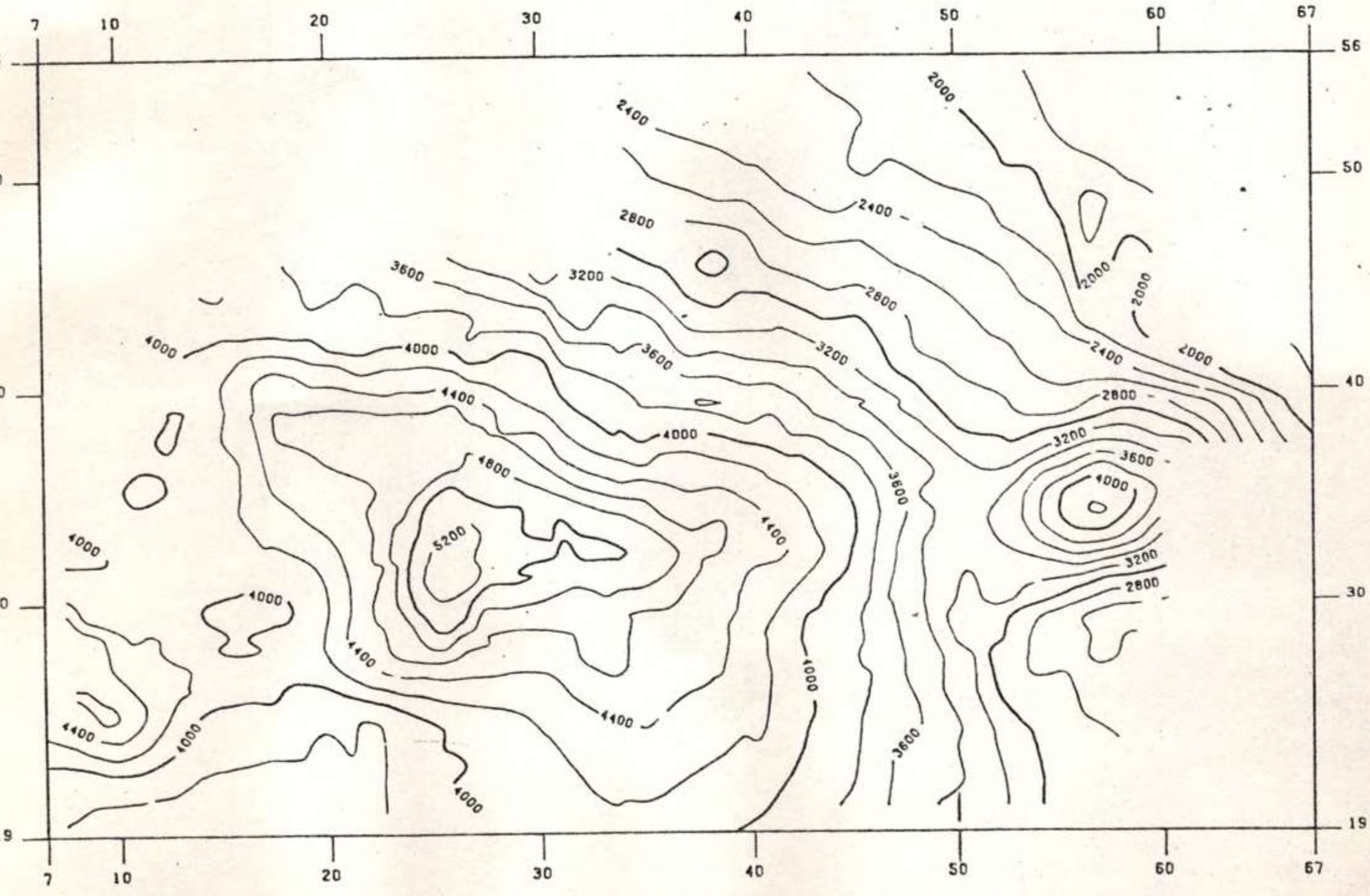


Fig. 10. Data contouring after interpolation on the grid of fig. 9, using two-dimensional estimation.

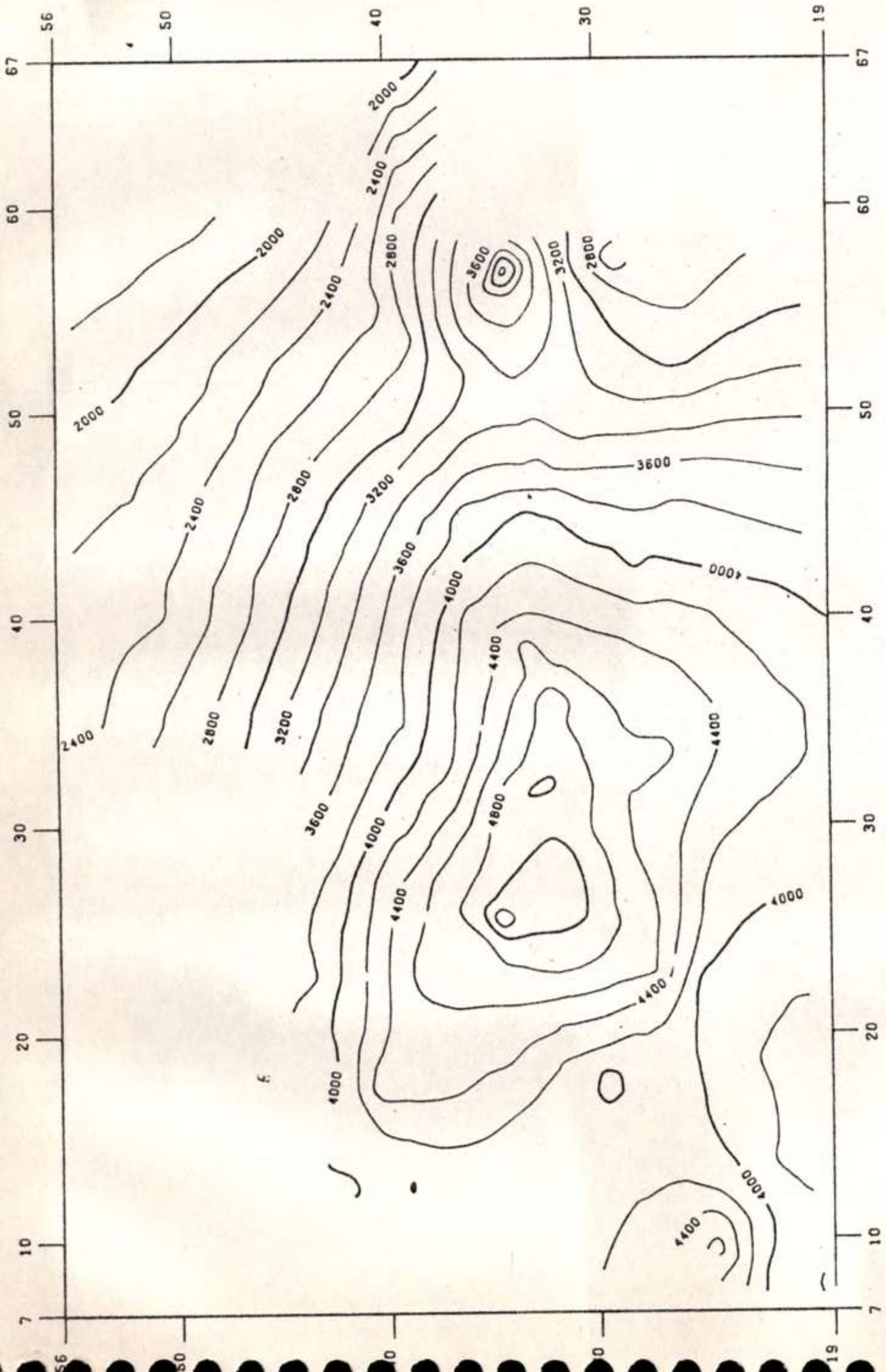


Fig. 11. Data contouring after interpolation on the grid of fig. 9 using two-dimensional estimation followed by twenty harmonizing iterations.

12. Conclusions

We have discussed a system for the contouring of randomly spaced data, guaranteeing the full respect of the given points. The two algorithms for the estimation of the crosspoints of the irregular grid are somehow complementary in that the two dimensional estimation may be a good starting point for the successive iterated smoothing. The enrichment of the lattice in respect to the minimal covering may be helpful to regularize the lattice itself, even if it may be expensive in computer time.

The same algorithm could have been applied in presence of faults: in this case though, the use of a regular lattice, and therefore the renouncement to the full respect of the data, is of great help to simplify the data management.

Aknowledgements

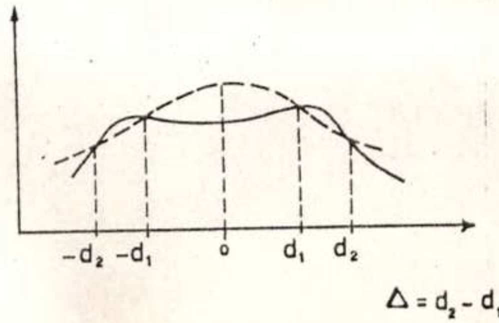
The authors wish to thank AGIP - Attività Minerarie for the permission to present this paper. They would also like to thank Dr. D. Fenati for his suggestions and criticism during the preparation of the paper.

Appendix

Effects of the choice of a covariance function

We will now see the effects of the choice of a covariance function. For simplicity, we will examine a monodimensional, symmetric case. We want to estimate $z(0)$, from $z(d_1)$, $z(-d_1)$, $z(d_2)$, $z(-d_2)$. We will also suppose that $d_2 > d_1$ (fig. A-1). For symmetry we require that the estimator be

$$\begin{cases} \hat{z}(0) = k_1 \cdot [z(d_1) + z(-d_1)] + k_2[z(d_2) + z(-d_2)] \\ 1 = k_1 \cdot 2 + k_2 \cdot 2 \end{cases} \tag{A-1}$$



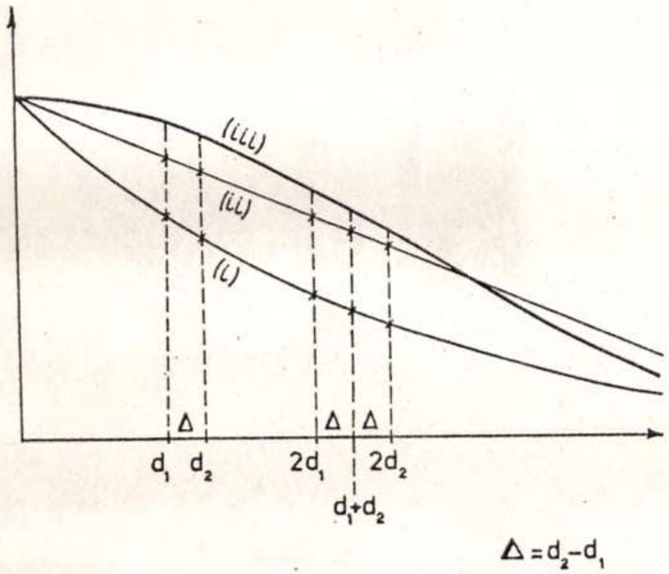
A-1. Interpolation using mono-dimensional estimation with different covariance functions. Dashed curve: covariance concave down. Continuous curve: covariance concave up.

The matrix of the system in this case of bounded minimum of the error is

$$\begin{bmatrix} R(0) & R(2d_1) & R(d_2-d_1) & R(d_2+d_1) & 1 \\ R(2d_1) & R(0) & R(d_2+d_1) & R(d_2-d_1) & 1 \\ R(d_2-d_1) & R(d_2+d_1) & R(0) & R(2d_2) & 1 \\ R(d_2+d_1) & R(d_2-d_1) & R(2d_2) & R(0) & 1 \\ 1 & 1 & 1 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} k_1 \\ k_1 \\ k_2 \\ k_2 \\ \lambda/2 \end{bmatrix} = \begin{bmatrix} R(d_1) \\ R(d_1) \\ R(d_2) \\ R(d_2) \\ 1 \end{bmatrix}. \qquad \text{(A-2)}$$

Calling

$$a = R(0) - R(d_2 - d_1) \qquad c = R(d_2 + d_1) - R(2d_2)$$
$$b = R(2d_1) - R(d_2 + d_1) \qquad d = R(d_1) - R(d_2) \qquad \text{(A-3)}$$



A-2. Possible covariance functions.

we have

$$k_1 = \tfrac{1}{2}\,\frac{a - c + 2d}{2a + b - c} \qquad k_2 = \tfrac{1}{2}\,\frac{a + b - 2d}{2a + b - c}. \qquad \text{(A-4)}$$

If we have extrapolation, k_1, k_2 must have opposite sign and therefore one of the two following sets of inequalities has to be true:

$$\left.\begin{matrix} a + 2d - c > 0 \\ a + b - 2d < 0 \end{matrix}\right\} \quad \text{or} \quad \left.\begin{matrix} a + 2d - c < 0 \\ a + b - 2d > 0 \end{matrix}\right\}. \qquad \text{(A-5)}$$

Now let us suppose that in the range of interest (fig. A-2):

(i) $R(x)$ is concave upward
(ii) $R(x)$ is linear
(iii) $R(x)$ is concave downward

In case (i), $a > b > c > d > 0$ and therefore we can never have extrapolation since

$$a + 2d - c > 0$$
$$a + b - 2d > 0;$$

in case (ii), we have always $k_1 = 1/2$ and $k_2 = 0$;
in case (iii), $0 < a < b < c < d$ and therefore (A-5) is satisfied and we do have extrapolation.

This very simple example shows that, at least in this case, if we choose an exponential function for $R(\xi)$, the interpolated values will not exceed the range of the data; the opposite would have happened had we chosen for the covariance function a Gaussian of the type $R(\xi) = e^{-\xi^2/\xi_0^2}$ and also we had $d_1, d_2 < \xi_0$.

In the two-dimensional non-symmetric case, these results are probably still valid, but no definite proof is available at present. It is nonetheless intuitive that if the points which are the basis of the interpolation are at such distance from the point to be estimated that the covariance function taken at these points is still rather flat, the algorithm will tend to interpret the surface as locally very even; this may correspond to the strong extrapolation of local trends.

References

Ahlberg, J. N., Nilsson, E. N., and Walsh, J. L., 1967, The theory of splines and their applications, Academic Press, New York.

Crain, E. R., 1972, Review of gravity and magnetic data processing systems, Journal of the Canadian Society of Exploration Geophysicists 8, 54-77.

Hessing, R. C., Lee, H. K., Pierce, A., and Powers, E. N., 1972, Automatic contouring using bicubic functions, Geophysics 37, 669-674.

Merril, R. D., 1973, Representation of contours and regions for efficient computer search. Communications Association for Computing Machinery, 16, 69-82.

Merriam, D. F., 1970, Geostatistic: a Colloquium, Plenum Press, New York.

Papoulis, A., 1968, Systems and transforms with applications to optics, Mc Graw Hill, New York.

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

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, 1974 a,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

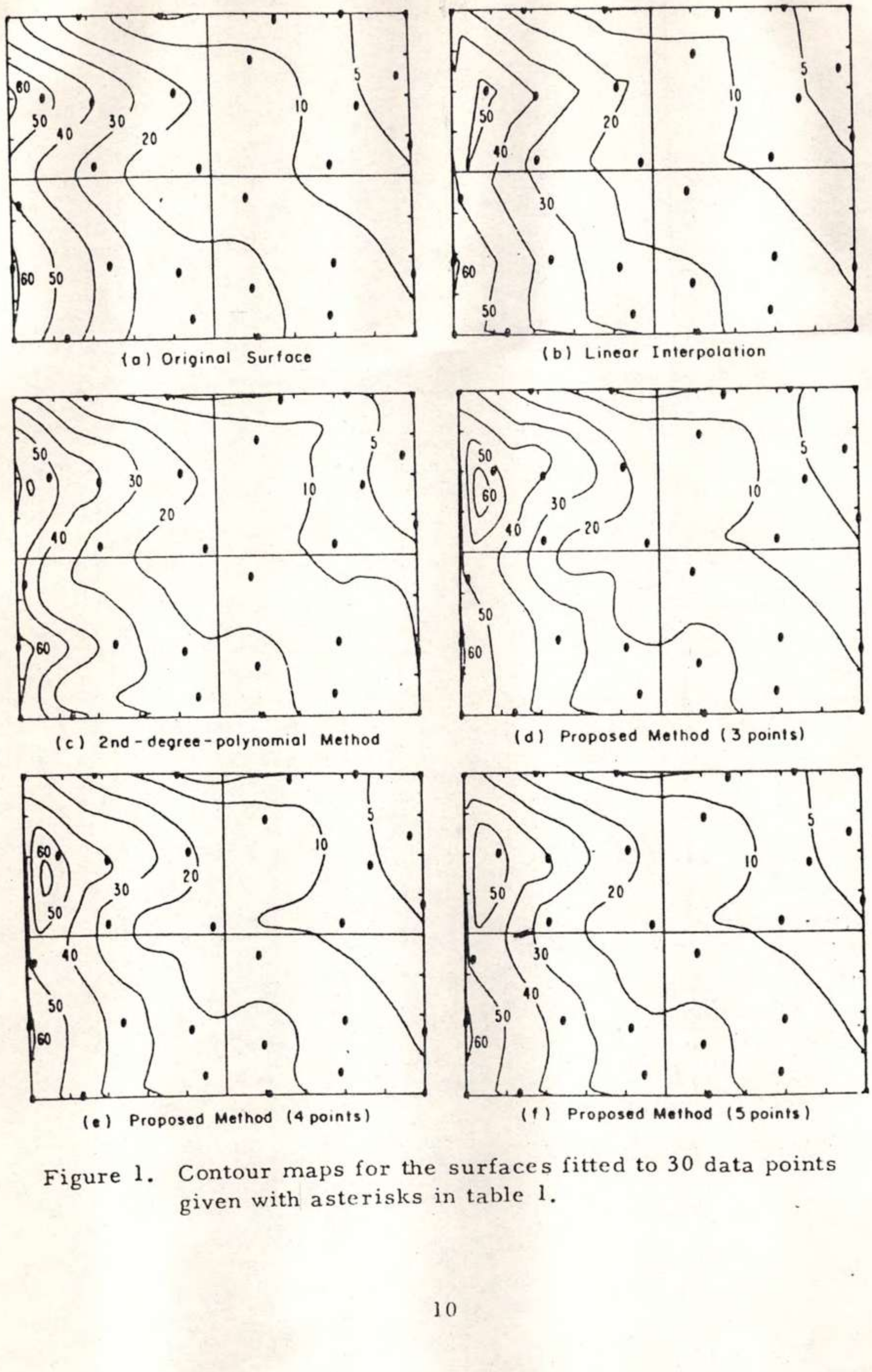


Figure 1. Contour maps for the surfaces fitted to 30 data points given with asterisks in table 1.

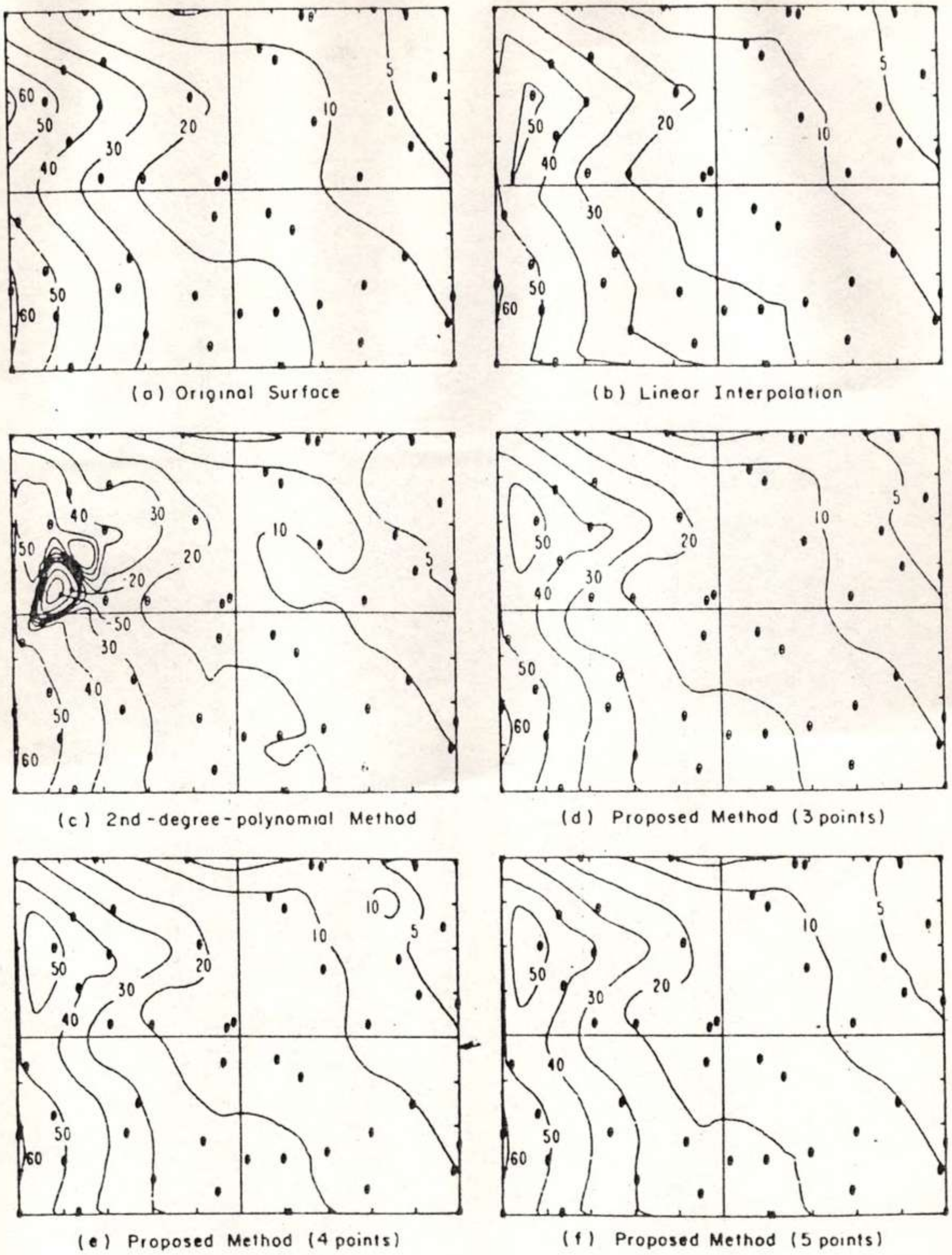


Figure 2. Contour maps for the surfaces fitted to 50 data points given in table 1.

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

$$x = au + bv + x_0 ,$$
$$y = cu + dv + y_0 , \qquad \text{(A-3)}$$

where

$$a = x_2 - x_1 ,$$
$$b = x_3 - x_1 ,$$
$$c = y_2 - y_1 ,$$
$$d = y_3 - y_1 , \qquad \text{(A-4)}$$
$$x_0 = x_1 ,$$
$$y_0 = y_1 .$$

The inverse relation is

$$u = [\ d(x - x_0) - b(y - y_0)] / (ad - bc) ,$$
$$v = [-c(x - x_0) + a(y - y_0)] / (ad - bc) . \qquad \text{(A-5)}$$

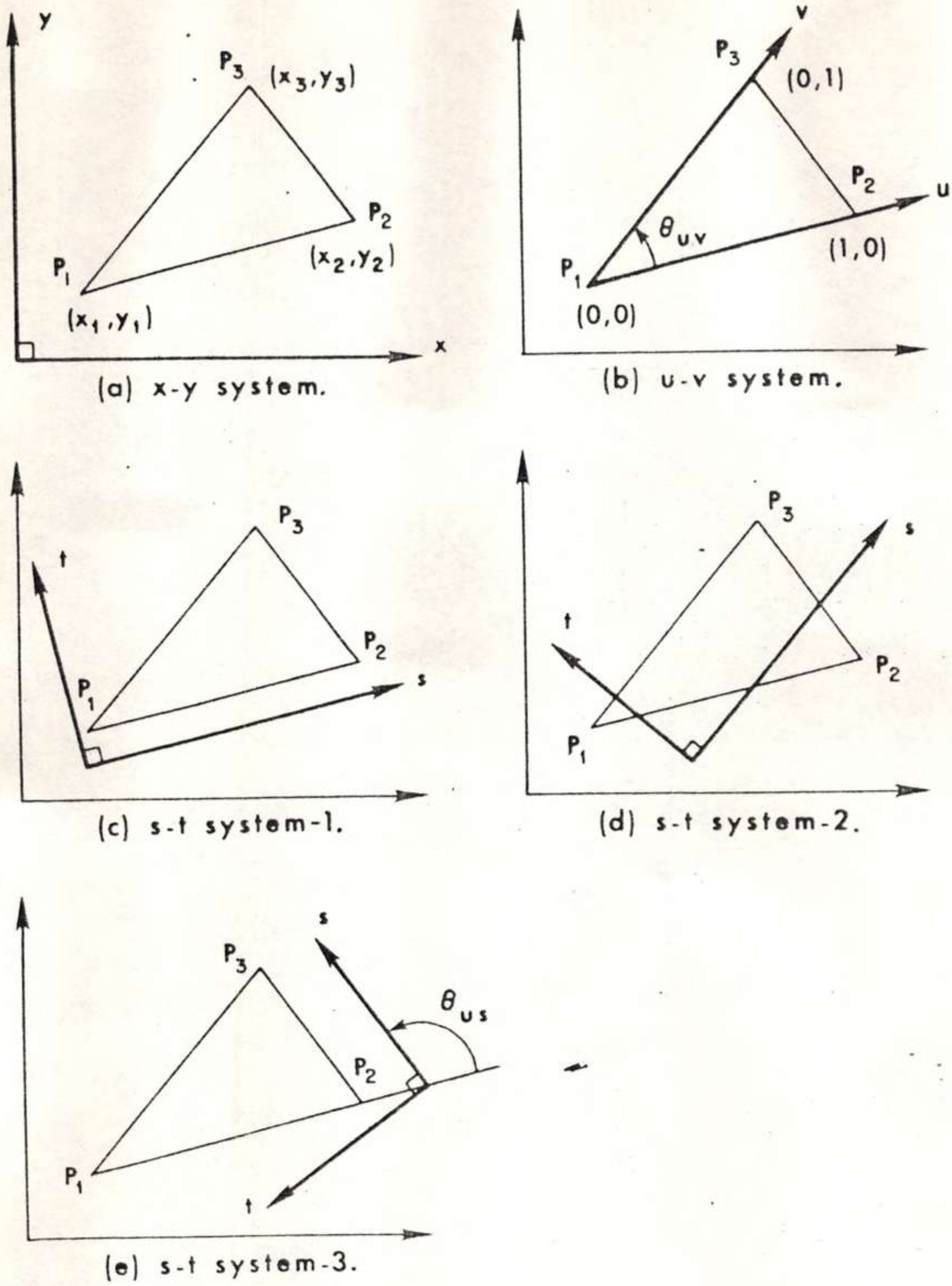


Figure A-1. Various coordinate systems.

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

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

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

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

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

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

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

We denote the lengths of the unit vectors in the u-v system (i.e., the lengths of sides $\overline{V_1 V_2}$ and $\overline{V_1 V_3}$) by L_u and L_v, respectively, and the angle between the u and v axes by θ_{uv}. They are given by

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

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 $\overline{V_1 V_2}$, as shown in figure A-1(c). The coordinate transformation between the u-v system and the s-t system is expressed by

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

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

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

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

$$L_u P_{41} - 5 L_v \cos\theta_{uv} P_{50} = 0 . \tag{A-12}$$

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

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

Partial derivatives are expressed by

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

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

$$L_v P_{14} - 5 L_u \cos\theta_{uv} P_{05} = 0 . \tag{A-15}$$

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

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

where

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

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

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

$$p_{00} = z(0,0) ,$$

$$p_{10} = z_u(0,0) ,$$

$$p_{01} = z_v(0,0) , \qquad \text{(A-20)}$$

$$p_{20} = z_{uu}(0,0) / 2 ,$$

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

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

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:

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

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

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

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

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

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

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

$$p_{14} = \frac{5\,L_u \cos\theta_{uv}}{L_v}\, p_{05}\,. \tag{A-23}$$

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

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

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

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

Equation (A-19) is rewritten as

$$g_1\,p_{32} + g_2\,p_{23} = h_1\,, \tag{A-26}$$

where

$$g_1 = A^2 C\,(3\,BC + 2\,AD)\,,$$

$$g_2 = A C^2\,(2\,BC + 3\,AD)\,, \tag{A-27}$$

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

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

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)
C THIS SUBROUTINE PERFORMS BIVARIATE INTERPOLATION WHEN THE PRO-
C JECTIONS OF THE DATA POINTS IN THE X-Y PLANE ARE IRREGULARLY
C DISTRIBUTED IN THE PLANE.
C THE INPUT PARAMETERS ARE
C     MD  = MODE OF COMPUTATION (MUST BE 1, 2, OR 3),
C         = 1 FOR NEW XD-YD,
C         = 2 FOR OLD XD-YD, NEW XI-YI,
C         = 3 FOR OLD XD-YD, OLD XI-YI,
C     NDP = NUMBER OF DATA POINTS (MUST BE 4 OR GREATER),
C     XD  = ARRAY OF DIMENSION NDP STORING THE X COORDINATES
C           OF THE DATA POINTS,
C     YD  = ARRAY OF DIMENSION NDP STORING THE Y COORDINATES
C           OF THE DATA POINTS,
C     ZD  = ARRAY OF DIMENSION NDP STORING THE Z COORDINATES
C           OF THE DATA POINTS,
C     WK  = ARRAY OF DIMENSION (2*NDP+NNP+5)*NDP+NIP
C           TO BE USED AS A WORK AREA,
C     NIP = NUMBER OF INTERPOLATED POINTS
C           (MUST BE 1 OR GREATER),
C     XI  = ARRAY OF DIMENSION NIP STORING THE X COORDINATES
C           OF THE INTERPOLATED POINTS,
C     YI  = ARRAY OF DIMENSION NIP STORING THE Y COORDINATES
C           OF THE INTERPOLATED POINTS,
C WHERE NNP IS THE NUMBER OF ADDITIONAL DATA POINTS USED FOR
C ESTIMATING PARTIAL DERIVATIVES AT EACH DATA POINT.  THE VALUE
C OF NNP MUST BE GIVEN THROUGH THE IDNN COMMON.  NNP MUST BE 2
C OR GREATER, BUT SMALLER THAN NDP.
C THE OUTPUT PARAMETER IS
C     ZI  = ARRAY OF DIMENSION NIP, WHERE THE Z COORDINATES
C           OF THE INTERPOLATED POINTS ARE TO BE DISPLAYED.
C THE LUN CONSTANT IN THE DATA INITIALIZATION STATEMENT IS THE
C LOGICAL UNIT NUMBER OF THE STANDARD OUTPUT UNIT AND IS,
C THEREFORE, SYSTEM DEPENDENT.
C DECLARATION STATEMENTS
      DIMENSION   XD(10),YD(10),ZD(10),WK(1000),
     1            XI(10),YI(10),ZI(10)
      COMMON/IDNN/NNP
      COMMON/IDGM/NDPC,NNPC,NT,NL
      COMMON/IDPI/NCF,ICF
      EQUIVALENCE (FNDPO,NDPO),(FNDPPV,NDPPV),
     1            (FNNPO,NNPO),(FNNPPV,NNPPV),
     2            (FNIPO,NIPO),(FNIPPV,NIPPV),
     3            (FNT,NT),(FNL,NL)
      DATA  LUN/6/
C SETTING OF SOME INPUT PARAMETERS TO LOCAL VARIABLES.  (ALL MD)
   10 MDO=MD
      NDPO=NDP
      NDPC=NDPO
      NIPO=NIP
      NNPO=NNP
      NNPC=NNPO
C ERROR CHECK.  (ALL MD)
   20 IF(MDO.LT.1.OR.MDO.GT.3)       GO TO 90
      IF(NDPO.LT.4)       GO TO 90
      IF(NIPO.LT.1)       GO TO 90
      IF(NNPO.LT.2.OR.NNPO.GE.NDPO) GO TO 90
      IF(MDO.NE.1)        GO TO 22
   21 WK(1)=FNDPO
      WK(2)=FNNPO
      GO TO 24
   22 FNDPPV=WK(1)
      FNNPPV=WK(2)
      IF(NDPO.NE.NDPPV)    GO TO 90
      IF(NNPO.NE.NNPPV)    GO TO 90
```

```
      IF(MDO.NE.3)            GO TO 24
   23 FNIPPV=WK(3)
      IF(NIPO.NE.NIPPV)    GO TO 90
      GO TO 30
   24 WK(3)=FNIPO
C ALLOCATION OF STORAGE AREAS IN THE WK ARRAY.   (ALL MD)
   30 NDNDM1=NDPO*(NDPO-1)
      IWIPT=7
      IWIPL =IWIPT+NDNDM1
      IWIPN=IWIPL+NDNDM1
      IWPD  =IWIPN+NDPO*NNPO
      IWIT  =IWPD  +NDPO*5
C DIVIDES THE X-Y PLANE INTO A NUMBER OF TRIANGLES AND
C DETERMINES NNP POINTS NEAREST EACH DATA POINT.   (MD=1)
   40 IF(MD.GT.1)       GO TO 42
   41 CALL IDGEOM(XD,YD,WK(IWIPT),WK(IWIPL),WK(IWIPN))
      WK(5)=FNT
      WK(6)=FNL
      GO TO 50
   42 FNT=WK(5)
      FNL=WK(6)
C ESTIMATES PARTIAL DERIVATIVES AT ALL DATA POINTS.   (ALL MD)
   50 CALL IDPDRV(XD,YD,ZD,WK(IWIPN),WK(IWPD))
C LOCATES ALL INTERPOLATED POINTS.   (MD=1,2)
   60 IF(MDO.EQ.3)    GO TO 70
      JWIT=IWIT-1
      DO 61   IIP=1,NIPO
        JWIT=JWIT+1
        CALL  IDLCTN(XD,YD,WK(IWIPT),WK(IWIPL),
     1                XI(IIP),YI(IIP),WK(JWIT))
   61 CONTINUE
C INTERPOLATION OF THE ZI VALUES.   (ALL MD)
   70 NCF=0
      ICF=0
      JWIT=IWIT-1
      DO 71   IIP=1,NIPO
        JWIT=JWIT+1
        CALL  IDPTIP(XD,YD,ZD,WK(IWIPT),WK(IWIPL),WK(IWPD),
     1               WK(JWIT),XI(IIP),YI(IIP),ZI(IIP))
   71 CONTINUE
C NORMAL EXIT
   80 RETURN
C ERROR EXIT
   90 WRITE (LUN,2090) MDO,NDPO,NIPO,NNPO
      RETURN
C FORMAT STATEMENT FOR ERROR MESSAGE
 2090 FORMAT(1X/41H ***     IMPROPER INPUT PARAMETER VALUE(S)./
     1    7H    MD =,I4,10X,5HNDP =,I6,10X,5HNIP =,I6,
     2    10X,5HNNP =,I6/
     3    35H ERROR DETECTED IN ROUTINE    IDBVIP/)
      END
```

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

```
C THE INPUT PARAMETERS ARE
C     XD,YD = ARRAYS STORING THE X AND Y COORDINATES, RESP.,
C             OF DATA POINTS.
C THE OUTPUT PARAMETERS ARE
C     IPT = ARRAY OF DIMENSION 3*NT, WHERE THE POINT NUMBERS
C           OF THE VERTEXES OF THE (IT)TH TRIANGLE ARE TO BE
C           DISPLAYED AS THE (3*IT-2)ND, (3*IT-1)ST, AND
C           (3*IT)TH ELEMENTS, IT=1,2,...,NT,
C     IPL = ARRAY OF DIMENSION 3*NL, WHERE THE POINT NUMBERS
C           OF THE END POINTS OF THE (IL)TH BORDER LINE
C           SEGMENT AND ITS RESPECTIVE TRIANGLE NUMBER ARE
C           TO BE DISPLAYED AS THE (3*IL-2)ND, (3*IL-1)ST,
C           AND (3*IL)TH ELEMENTS, IL=1,2,..., NL,
C     IPN = ARRAY OF DIMENSION NDP*NNP, WHERE THE POINT
C           NUMBERS OF NNP DATA POINTS NEAREST TO EACH OF
C           THE DATA POINTS ARE TO BE DISPLAYED,
C WHERE NDP IS THE TOTAL NUMBER OF DATA POINTS, NNP IS THE
C NUMBER OF DATA POINTS NEAREST TO EACH DATA POINT, NL IS
C THE NUMBER OF BORDER LINE SEGMENTS, AND NT IS THE NUMBER
C OF TRIANGLES.  NDP AND NNP ARE GIVEN TO THIS SUBROUTINE
C THROUGH THE IDGM COMMON.  NL AND NT ARE CALCULATED BY THIS
C SUBROUTINE AND ARE LEFT IN THE IDGM COMMON AT COMPLETION.
C DECLARATION STATEMENTS
      DIMENSION   XD(10),YD(10),IPT(100),IPL(100),IPN(50)
      COMMON/IDGM/NDP,NNP,NT,NL
      EQUIVALENCE (DSQ1,IDSQ1),(DSQ2,IDSQ2),(DSQM,IDSQM)
C PRELIMINARY PROCESSING
   10 NDP0=NDP
      NDPM1=NDP0-1
      NNP0=NNP
      NNPM1=NNP0-1
C DETERMINES THE NEAREST NNP POINTS.
   20 DO 29  IP1=1,NDP0
        X1=XD(IP1)
        Y1=YD(IP1)
        J1MX=IP1*NNP0
        J1MN=J1MX-NNPM1
        DO 28  J1=J1MN,J1MX
          J2MX=J1-1
          IDMN=0
          DO 27  IP2=1,NDP0
            IF(IP2.EQ.IP1)            GO TO 27
            IF(J1.GT.J1MN)            GO TO 22
   21       DSQ1=(XD(IP2)-X1)**2+(YD(IP2)-Y1)**2
            IPT(IP2)=IDSQ1
            GO TO 23
   22       IDSQ1=IPT(IP2)
   23       IF(IDMN.EQ.0)             GO TO 24
            IF(DSQ1.GE.DSQMN)         GO TO 27
   24       IF(J1MN.GT.J2MX)          GO TO 26
            DO 25  J2=J1MN,J2MX
              IF(IP2.EQ.IPN(J2))      GO TO 27
   25       CONTINUE
   26       DSQMN=DSQ1
            IDMN=IP2
   27     CONTINUE
          IPN(J1)=IDMN
   28   CONTINUE
   29 CONTINUE
C LISTS ALL THE POSSIBLE LINE SEGMENTS IN THE IPL ARRAY.
C CALCULATES THE SQUARES OF THE LINE SEGMENT LENGTHS, AND STORE
C THEM IN THE IPT ARRAY.
   30 IL=0
      DO 32  IP1=1,NDPM1
        X1=XD(IP1)
```

```
          Y1=YD(IP1)
          IP1P1=IP1+1
          DO 31   IP2=IP1P1,NDPO
             IL=IL+1
             ILT2=IL+IL
             IPL(ILT2-1)=IP1
             IPL(ILT2)   =IP2
             DSQ1=(XD(IP2)-X1)**2+(YD(IP2)-Y1)**2
             IPT(IL)=IDSQ1
   31     CONTINUE
   32 CONTINUE
      NLO=IL
C SORTS THE IPL AND IPT ARRAYS IN ASCENDING ORDER OF THE LINE
C SEGMENT LENGTH (DISTANCE).
   35 NLM1=NLO-1
      DO 37   IL1=1,NLM1
         IDSQ1=IPT(IL1)
         ILM=IL1
         DSQM=DSQ1
         IL2MN=IL1+1
         DO 36   IL2=IL2MN,NLO
            IDSQ2=IPT(IL2)
            IF(DSQ2.GE.DSQM)        GO TO 36
            ILM=IL2
            DSQM=DSQ2
   36    CONTINUE
         IPT(ILM)=IDSQ1
         IPT(IL1)=IDSQM
         IL1T2=IL1+IL1
         ILMT2=ILM+ILM
         ITS=IPL(IL1T2-1)
         IPL(IL1T2-1)=IPL(ILMT2-1)
         IPL(ILMT2-1)=ITS
         ITS=IPL(IL1T2)
         IPL(IL1T2)=IPL(ILMT2)
         IPL(ILMT2)=ITS
   37 CONTINUE
C ELIMINATES LINE SEGMENTS THAT CROSS OR LIE OVER SHORTER ONE.
   40 ILO=1
      DO 46   IL1=2,NLO
         IL1T2=IL1+IL1
         IP1=IPL(IL1T2-1)
         IP2=IPL(IL1T2)
         X1=XD(IP1)
         X2=XD(IP2)
         Y1=YD(IP1)
         Y2=YD(IP2)
         DX21=X2-X1
         DY21=Y2-Y1
         DO 45   IL2=1,ILO
            IL2T2=IL2+IL2
            IP3=IPL(IL2T2-1)
            IP4=IPL(IL2T2)
            X3=XD(IP3)
            X4=XD(IP4)
            Y3=YD(IP3)
            Y4=YD(IP4)
            DX43=X4-X3
            DX42=X4-X2
            DX41=X4-X1
            DX32=X3-X2
            DX31=X3-X1
            DY43=Y4-Y3
            DY42=Y4-Y2
            DY41=Y4-Y1
```

```
      DY32=Y3-Y2
      DY31=Y3-Y1
      IF(IP3.NE.IP1)  GO TO 41
      IF(DY41*DX21-DX41*DY21.NE.0.0) GO TO 45
      IF(DX41*DX21+DY41*DY21)          45,45,46
41    IF(IP4.NE.IP1)  GO TO 42
      IF(DY31*DX21-DX31*DY21.NE.0.0) GO TO 45
      IF(DX31*DX21+DY31*DY21)          45,45,46
42    IF(IP3.NE.IP2)  GO TO 43
      IF(DY42*DX21-DX42*DY21.NE.0.0) GO TO 45
      IF(DX42*DX21+DY42*DY21)          46,45,45
43    IF(IP4.NE.IP2)  GO TO 44
      IF(DY32*DX21-DX32*DY21.NE.0.0) GO TO 45
      IF(DX32*DX21+DY32*DY21)          46,45,45
44    IF((DY31*DX21-DX31*DY21)*(DY41*DX21-DX41*DY21).GE.0.0)
     1                                 GO TO 45
      IF((DY31*DX43-DX31*DY43)*(DY32*DX43-DX32*DY43).LT.0.0)
     1                                 GO TO 46
45    CONTINUE
        ILO=ILO+1
        ILOT2=ILO+ILO
        IPL(ILOT2-1)=IP1
        IPL(ILOT2)  =IP2
   46 CONTINUE
      NLO=ILO
C RE-SORTS THE IPL ARRAY IN ASCENDING ORDER OF ITS ELEMENTS.
   50 NLT2=NLO+NLO
      NLM1T2=NLT2-2
      DO 54  IL1T2=2,NLM1T2,2
        ILMT2=IL1T2
        IPM1=IPL(ILMT2-1)
        IPM2=IPL(ILMT2)
        IL2T2M=IL1T2+2
        DO 53  IL2T2=IL2T2M,NLT2,2
          IP21=IPL(IL2T2-1)
          IP22=IPL(IL2T2)
          IF(IPM1-IP21)      53,51,52
51        IF(IPM2-IP22)      53,53,52
52        ILMT2=IL2T2
          IPM1=IP21
          IPM2=IP22
53      CONTINUE
        IPL(ILMT2-1)=IPL(IL1T2-1)
        IPL(ILMT2)  =IPL(IL1T2)
        IPL(IL1T2-1)=IPM1
        IPL(IL1T2)  =IPM2
   54 CONTINUE
C DETERMINES TRIANGLES.
   60 IT=0
      NLM1=NLO-1
      NLM2=NLO-2
      DO 67  IL1=1,NLM2
        IL1T2=IL1+IL1
        IP1=IPL(IL1T2-1)
        IP2=IPL(IL1T2)
        IL1P1=IL1+1
        DO 66  IL2=IL1P1,NLM1
          IL2T2=IL2+IL2
          IF(IPL(IL2T2-1).NE.IP1)    GO TO 67
          IP3=IPL(IL2T2)
          IL2P1=IL2+1
          DO 62  IL3=IL2P1,NLO
            IL3T2=IL3+IL3
            IF(IPL(IL3T2-1)-IP2)     62,61,66
61          IF(IPL(IL3T2)  -IP3)     62,63,66
```

```
62      CONTINUE
        GO TO 66
63      IPT1=IP1
        IPT2=IP2
        IPT3=IP3
        IF((YD(IPT3)-YD(IPT1))*(XD(IPT2)-XD(IPT1))-
     1     (XD(IPT3)-XD(IPT1))*(YD(IPT2)-YD(IPT1)).GE.0.0)
     2                                              GO TO 64
        ITS=IPT2
        IPT2=IPT3
        IPT3=ITS
64      X1=XD(IPT1)
        X2=XD(IPT2)
        X3=XD(IPT3)
        Y1=YD(IPT1)
        Y2=YD(IPT2)
        Y3=YD(IPT3)
        DX32=X3-X2
        DX21=X2-X1
        DX13=X1-X3
        DY32=Y3-Y2
        DY21=Y2-Y1
        DY13=Y1-Y3
        DO 65  IPO=1,NDPO
          IF(IPO.EQ.IPT1.OR.IPO.EQ.IPT2.OR.IPO.EQ.IPT3)
     1                                              GO TO 65
          XO=XD(IPO)
          YO=YD(IPO)
          IF((YO-Y1)*DX21-(XO-X1)*DY21.LT.0.0)    GO TO 65
          IF((YO-Y2)*DX32-(XO-X2)*DY32.LT.0.0)    GO TO 65
          IF((YO-Y3)*DX13-(XO-X3)*DY13.GE.0.0)    GO TO 66
65      CONTINUE
        IT=IT+1
        ITT3=IT*3
        IPT(ITT3-2)=IPT1
        IPT(ITT3-1)=IPT2
        IPT(ITT3)   =IPT3
66    CONTINUE
67 CONTINUE
      NTO=IT
      NT=NTO
C SELECTS AND SORTS LINE SEGMENTS THAT FORM THE BORDER.
70 ILO=0
      DO 75  IL1=1,NLO
        IL1T2=IL1+IL1
        IP1=IPL(IL1T2-1)
        IP2=IPL(IL1T2)
        X1=XD(IP1)
        Y1=YD(IP1)
        X2=XD(IP2)
        Y2=YD(IP2)
        DX21=X2-X1
        DY21=Y2-Y1
        DO 71   IPO=1,NDPO
          IF(IPO.EQ.IP1.OR.IPO.EQ.IP2)    GO TO 71
          S=(YD(IPO)-Y1)*DX21-(XD(IPO)-X1)*DY21
          IF(S.NE.0.0)     GO TO 72
71      CONTINUE
72      IPOMN=IPO+1
        DO 73   IPO=IPOMN,NDPO
          IF(IPO.EQ.IP1.OR.IPO.EQ.IP2)    GO TO 73
          IF(((YD(IPO)-Y1)*DX21-(XD(IPO)-X1)*DY21)*S.LT.0.0)
     1                                              GO TO 75
73      CONTINUE
```

```
      ILO=ILO+1
      ILOT2=ILO+ILO
      IF(S.LT.0.0) GO TO 74
      IPL(ILOT2-1)=IP1
      IPL(ILOT2)  =IP2
      GO TO 75
   74 IPL(ILOT2-1)=IP2
      IPL(ILOT2)  =IP1
   75 CONTINUE
      NLO=ILO
      NLM1=NLO-1
      DO 79  IL1=2,NLM1
        IL1T2=IL1+IL1
        IP2=IPL(IL1T2-2)
        IF(IPL(IL1T2-1).EQ.IP2)      GO TO 79
        IL1P1=IL1+1
        DO 77  IL2=IL1P1,NLO
          IL2T2=IL2+IL2
          IF(IPL(IL2T2-1).EQ.IP2)    GO TO 78
   77   CONTINUE
   78   IP1=IPL(IL1T2-1)
        IP2=IPL(IL1T2)
        IPL(IL1T2-1)=IPL(IL2T2-1)
        IPL(IL1T2)  =IPL(IL2T2)
        IPL(IL2T2-1)=IP1
        IPL(IL2T2)  =IP2
   79 CONTINUE
      NL=NLO
C FINDS OUT TRIANGLES CORRESPONDING TO THE BORDER LINE
C SEGMENTS.
   80 NLP1=NLO+1
      DO 83  ILR=1,NLO
        IL=NLP1-ILR
        ILT2=IL+IL
        ILT3=ILT2+IL
        IPL1=IPL(ILT2-1)
        IPL2=IPL(ILT2)
        DO 81  IT=1,NTO
          ITT3=IT*3
          IPT1=IPT(ITT3-2)
          IPT2=IPT(ITT3-1)
          IPT3=IPT(ITT3)
          IF(IPL1.NE.IPT1.AND.IPL1.NE.IPT2.AND.IPL1.NE.IPT3)
     1                                                   GO TO 81
          IF(IPL2.EQ.IPT1.OR.IPL2.EQ.IPT2.OR.IPL2.EQ.IPT3)
     1                                                   GO TO 82
   81   CONTINUE
   82   IPL(ILT3-2)=IPL1
        IPL(ILT3-1)=IPL2
        IPL(ILT3)  =IT
   83 CONTINUE
      RETURN
      END
```

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

```
C      XD,YD = ARRAYS STORING THE X AND Y COORDINATES, RESP.,
C              OF DATA POINTS.
C      IPT = ARRAY STORING THE POINT NUMBERS OF THE VERTEXES
C            OF THE TRIANGLES,
C      IPL = ARRAY STORING THE POINT NUMBERS OF THE END
C            POINTS OF THE BORDER LINE SEGMENTS AND THEIR
C            RESPECTIVE TRIANGLE NUMBERS,
C      XII,YII = X AND Y COORDINATES, RESP., OF
C                INTERPOLATED POINT.
C THE OUTPUT PARAMETER IS
C      ITI = TRIANGLE NUMBER, WHEN THE POINT IS INSIDE THE
C            DATA AREA, OR
C            TWO BORDER LINE SEGMENT NUMBERS, IL1 AND IL2,
C            CODED TO IL1*(NT+NL)+IL2, WHEN THE POINT IS
C            OUTSIDE THE DATA AREA, WHERE NT IS THE NUMBER OF
C            TRIANGLES AND NL, THAT OF BORDER LINE SEGMENTS.
C DECLARATION STATEMENTS
      DIMENSION   XD(10),YD(10),IPT(100),IPL(100)
      COMMON/IDGM/NDP,NNP,NT,NL
      DATA  NTPV/0/,NLPV/0/
C PRELIMINARY PROCESSING
   10 NT0=NT
      NL0=NL
      NTL=NT0+NL0
      X0=XII
      Y0=YII
C CHECK IF IN THE SAME TRIANGLE AS PREVIOUS
   20 IF(NT0.NE.NTPV)      GO TO 35
      IF(NL0.NE.NLPV)      GO TO 35
      IT0=ITIPV
      IF(IT0.GT.NT0)       GO TO 25
      IT0T3=IT0*3
      IP1=IPT(IT0T3-2)
      IP2=IPT(IT0T3-1)
      IP3=IPT(IT0T3)
      X1=XD(IP1)
      X2=XD(IP2)
      X3=XD(IP3)
      Y1=YD(IP1)
      Y2=YD(IP2)
      Y3=YD(IP3)
      IF((Y0-Y1)*(X2-X1)-(X0-X1)*(Y2-Y1))         50,21,21
   21 IF((Y0-Y2)*(X3-X2)-(X0-X2)*(Y3-Y2))         50,22,22
   22 IF((Y0-Y3)*(X1-X3)-(X0-X3)*(Y1-Y3))         50,80,80
C CHECK IF ON THE SAME BORDER LINE SEGMENT
   25 IL1=IT0/NTL
      IL2=IT0-IL1*NTL
      IL1T3=IL1*3
      IP1=IPL(IL1T3-2)
      IP2=IPL(IL1T3-1)
      X1=XD(IP1)
      X2=XD(IP2)
      Y1=YD(IP1)
      Y2=YD(IP2)
      DX02=X0-X2
      DY02=Y0-Y2
      DX21=X2-X1
      DY21=Y2-Y1
      CS0221=DX02*DX21+DY02*DY21
      IF(IL2.NE.IL1)      GO TO 30
      IF(CS0221)      26,26,50
   26 DX01=X0-X1
      DY01=Y0-Y1
      IF(DY01*DX21-DX01*DY21)   27,27,50
   27 IF(DX01*DX21+DY01*DY21)   50,80,80
```

```
C CHECK IF BETWEEN THE SAME TWO BORDER LINE SEGMENTS
   30 IF(CS0221)      50,31,31
   31 IL2T3=IL2*3
      IP3=IPL(IL2T3-1)
      X3=XD(IP3)
      Y3=YD(IP3)
      DX32=X3-X2
      DY32=Y3-Y2
      IF(DX02*DX32+DY02*DY32)  80,80,50
C WHEN CALLED WITH A NEW SET OF NT AND NL
   35 NTPV=NT0
      NLPV=NL0
      ITIPV=0
C LOCATION INSIDE THE DATA AREA
   50 ITOT3=0
      DO 69  IT0=1,NT0
        ITOT3=ITOT3+3
        IF(IT0.EQ.ITIPV)  GO TO 69
        IP1=IPT(ITOT3-2)
        IP2=IPT(ITOT3-1)
        IP3=IPT(ITOT3)
        X1=XD(IP1)
        X2=XD(IP2)
        X3=XD(IP3)
        IF(X0-X1)       53,55,51
   51   IF(X0-X2)       55,55,52
   52   IF(X0-X3)       55,55,69
   53   IF(X0-X2)       54,55,55
   54   IF(X0-X3)       69,55,55
   55   Y1=YD(IP1)
        Y2=YD(IP2)
        Y3=YD(IP3)
        IF(Y0-Y1)       58,60,56
   56   IF(Y0-Y2)       60,60,57
   57   IF(Y0-Y3)       60,60,69
   58   IF(Y0-Y2)       59,60,60
   59   IF(Y0-Y3)       69,60,60
   60   IF((Y0-Y1)*(X2-X1)-(X0-X1)*(Y2-Y1))     69,61,61
   61   IF((Y0-Y2)*(X3-X2)-(X0-X2)*(Y3-Y2))     69,62,62
   62   IF((Y0-Y3)*(X1-X3)-(X0-X3)*(Y1-Y3))     69,80,80
   69 CONTINUE
C LOCATION OUTSIDE THE DATA AREA
   70 NLOT3=NL0*3
      IP1=IPL(NLOT3-2)
      IP2=IPL(NLOT3-1)
      X1=XD(IP1)
      Y1=YD(IP1)
      X2=XD(IP2)
      Y2=YD(IP2)
      DX02=X0-X2
      DY02=Y0-Y2
      DX21=X2-X1
      DY21=Y2-Y1
      CS0221=DX02*DX21+DY02*DY21
      DO 74  IL0=1,NL0
        X1=X2
        Y1=Y2
        DX01=DX02
        DY01=DY02
        IP2=IPL(3*IL0-1)
        X2=XD(IP2)
        Y2=YD(IP2)
        DX02=X0-X2
        DY02=Y0-Y2
```

```
      DX21=X2-X1
      DY21=Y2-Y1
      CSPV=CS0221
      CS0221=DX02*DX21+DY02*DY21
      IF(CS0221)     71,71,74
   71 IF(DX01*DX21+DY01*DY21)      73,72,72
   72 IF(DY01*DX21-DX01*DY21)      76,76,74
   73 IF(CSPV)      74,74,75
   74 CONTINUE
      ILO=1
   75 ITO=ILO-1
      IF(ITO.EQ.0)     ITO=NLO
      GO TO 77
   76 ITO=ILO
   77 ITO=ITO*NTL+ILO
C NORMAL EXIT
   80 ITI=ITO
      ITIPV=ITO
      RETURN
      END
```

```
      SUBROUTINE  IDPDRV(XD,YD,ZD,IPN,PD)
C THIS SUBROUTINE ESTIMATES PARTIAL DERIVATIVES OF THE FIRST AND
C SECOND ORDER AT THE DATA POINTS.
C THE INPUT PARAMETERS ARE
C     XD,YD,ZD = ARRAYS STORING THE X, Y, AND Z COORDINATES,
C                RESP., OF DATA POINTS,
C     IPN = ARRAY STORING THE POINT NUMBERS OF NNP DATA
C           POINTS NEAREST TO EACH OF THE DATA POINTS,
C WHERE NNP IS THE NUMBER OF DATA POINTS USED FOR ESTIMATION
C OF PARTIAL DERIVATIVES AT EACH DATA POINT.  NNP IS GIVEN
C THROUGH THE IDGM COMMON.
C THE OUTPUT PARAMETER IS
C     PD  = ARRAY OF DIMENSION 5*NDP, WHERE THE ESTIMATED
C           ZX, ZY, ZXX, ZXY, AND ZYY VALUES AT THE DATA
C           POINTS ARE TO BE DISPLAYED,
C WHERE NDP IS THE TOTAL NUMBER OF DATA POINTS.  NDP IS GIVEN
C THROUGH THE IDGM COMMON.
C DECLARATION STATEMENTS
      DIMENSION   XD(10),YD(10),ZD(10),IPN(100),PD(50)
      COMMON/IDGM/NDP,NNP,NT,NL
      REAL        NMX,NMY,NMZ,NMXX,NMXY,NMYX,NMYY
C PRELIMINARY PROCESSING
   10 NDP0=NDP
      NNP0=NNP
      NNPM1=NNP0-1
C ESTIMATION OF ZX AND ZY
   20 JPD0=-5
      JIPN0=-NNP0
      DO 24  IP0=1,NDP0
        JPD0=JPD0+5
        X0=XD(IP0)
        Y0=YD(IP0)
        Z0=ZD(IP0)
        NMX=0.0
        NMY=0.0
        NMZ=0.0
        JIPN0=JIPN0+NNP0
        DO 23   IN1=1,NNPM1
          JIPN=JIPN0+IN1
          IPI=IPN(JIPN)
          DX1=XD(IPI)-X0
```

```
      DY1=YD(IPI)-Y0
      DZ1=ZD(IPI)-Z0
      IN2MN=IN1+1
      DO 22  IN2=IN2MN,NNPO
        JIPN=JIPN0+IN2
        IPI=IPN(JIPN)
        DX2=XD(IPI)-X0
        DY2=YD(IPI)-Y0
        DZ2=ZD(IPI)-Z0
        DNMX=DY1*DZ2-DZ1*DY2
        DNMY=DZ1*DX2-DX1*DZ2
        DNMZ=DX1*DY2-DY1*DX2
        IF(DNMZ.GE.0.0)    GO TO 21
        DNMX=-DNMX
        DNMY=-DNMY
        DNMZ=-DNMZ
21      NMX=NMX+DNMX
        NMY=NMY+DNMY
        NMZ=NMZ+DNMZ
22    CONTINUE
23  CONTINUE
    PD(JPD0+1)=-NMX/NMZ
    PD(JPD0+2)=-NMY/NMZ
24 CONTINUE
C ESTIMATION OF ZXX, ZXY, AND ZYY
30 JPD0=-5
   JIPN0=-NNPO
   DO 34  IPO=1,NDPO
     JPD0=JPD0+5
     X0=XD(IPO)
     Y0=YD(IPO)
     ZX0=PD(JPD0+1)
     ZY0=PD(JPD0+2)
     NMXX=0.0
     NMXY=0.0
     NMYX=0.0
     NMYY=0.0
     NMZ =0.0
     JIPN0=JIPN0+NNPO
     DO 33  IN1=1,NNPM1
       JIPN=JIPN0+IN1
       IPI=IPN(JIPN)
       DX1=XD(IPI)-X0
       DY1=YD(IPI)-Y0
       JPD=5*(IPI-1)
       DZX1=PD(JPD+1)-ZX0
       DZY1=PD(JPD+2)-ZY0
       IN2MN=IN1+1
       DO 32  IN2=IN2MN,NNPO
         JIPN=JIPN0+IN2
         IPI=IPN(JIPN)
         DX2=XD(IPI)-X0
         DY2=YD(IPI)-Y0
         JPD=5*(IPI-1)
         DZX2=PD(JPD+1)-ZX0
         DZY2=PD(JPD+2)-ZY0
         DNMXX=DY1*DZX2-DZX1*DY2
         DNMXY=DZX1*DX2-DX1*DZX2
         DNMYX=DY1*DZY2-DZY1*DY2
         DNMYY=DZY1*DX2-DX1*DZY2
         DNMZ =DX1*DY2 -DY1*DX2
         IF(DNMZ.GE.0.0)    GO TO 31
         DNMXX=-DNMXX
         DNMXY=-DNMXY
```

```
            DNMYX=-DNMYX
            DNMYY=-DNMYY
            DNMZ =-DNMZ
   31       NMXX=NMXX+DNMXX
            NMXY=NMXY+DNMXY
            NMYX=NMYX+DNMYX
            NMYY=NMYY+DNMYY
            NMZ =NMZ +DNMZ
   32     CONTINUE
   33   CONTINUE
        PD(JPD0+3)=-NMXX/NMZ
        PD(JPD0+4)=-(NMXY+NMYX)/(2.0*NMZ)
        PD(JPD0+5)=-NMYY/NMZ
   34 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE  IDPTIP(XD,YD,ZD,IPT,IPL,PDD,ITI,XII,YII,ZII)
C THIS SUBROUTINE PERFORMS PUNCTUAL INTERPOLATION OR EXTRAPO-
C LATION, I.E., DETERMINES THE Z VALUE AT A POINT.
C THE INPUT PARAMETERS ARE
C     XD,YD,ZD = ARRAYS STORING THE X, Y, AND Z COORDINATES,
C                RESP., OF DATA POINTS,
C     IPT = ARRAY STORING THE POINT NUMBERS OF THE VERTEXES
C           OF THE TRIANGLES,
C     IPL = ARRAY STORING THE POINT NUMBERS OF THE END
C           POINTS OF THE BORDER LINE SEGMENTS AND THEIR
C           RESPECTIVE TRIANGLE NUMBERS,
C     PDD = ARRAY STORING THE PARTIAL DERIVATIVES AT THE
C           DATA POINTS,
C     ITI = TRIANGLE NUMBER OF THE TRIANGLE IN WHICH
C           THE INTERPOLATED POINT LIES,
C     XII,YII = X AND Y COORDINATES, RESP., OF
C               INTERPOLATED POINT.
C THE OUTPUT PARAMETER IS
C     ZII = INTERPOLATED Z VALUE.
C DECLARATION STATEMENTS
      DIMENSION   XD(10),YD(10),ZD(10),IPT(100),IPL(100),PDD(50)
      COMMON/IDGM/NDP,NNP,NT,NL
      COMMON/IDPI/NCF,ICF
      DIMENSION   CF0(27)
      EQUIVALENCE (X0,CF0(1)),   (Y0,CF0(2)),   (AP,CF0(3)),
     1            (BP,CF0(4)),   (CP,CF0(5)),   (DP,CF0(6)),
     2            (P00,CF0(7)), (P10,CF0(8)), (P20,CF0(9)),
     3            (P30,CF0(10)),(P40,CF0(11)),(P50,CF0(12)),
     4            (P01,CF0(13)),(P11,CF0(14)),(P21,CF0(15)),
     5            (P31,CF0(16)),(P41,CF0(17)),(P02,CF0(18)),
     6            (P12,CF0(19)),(P22,CF0(20)),(P32,CF0(21)),
     7            (P03,CF0(22)),(P13,CF0(23)),(P23,CF0(24)),
     8            (P04,CF0(25)),(P14,CF0(26)),(P05,CF0(27))
      DIMENSION   CF(980)
      DIMENSION   X(3),Y(3),Z(3),PD(15),
     1            ZU(3),ZV(3),ZUU(3),ZUV(3),ZVV(3)
      EQUIVALENCE (IT0,FLIT0),(ITJ,FLITJ)
      REAL        LU,LV,LUSNUV,LVSNUV
      EQUIVALENCE (P5,P05)
      DATA  NCFMX/35/
C SETTING OF SOME LOCAL VARIABLES.
   10 IT0=ITI
      XI0=XII
      YI0=YII
      NTL=NT+NL
```

```
C DETERMINES IF SIMPLE INTERPOLATION IS APPLICABLE.
   20 IF(ITO.LE.NTL)        GO TO 30
      IL1=ITO/NTL
      IL2=ITO-IL1*NTL
      IL1T3=IL1*3
      IL2T3=IL2*3
      ITO=IPL(IL1T3)
      IF(IL1.NE.IL2)        GO TO 40
C CALCULATION OF ZII BY SIMPLE INTERPOLATION OR EXTRAPOLATION.
   30 ASSIGN 31 TO LBL
      GO TO 50
   31 ZII=ZIO
      RETURN
C CALCULATION OF ZII AS A WEIGHTED MEAN OF TWO EXTRAPOLATED
C VALUES.
   40 ASSIGN 41 TO LBL
      GO TO 50
   41 ZI1=ZIO
      ITO=IPL(IL2T3)
      ASSIGN 42 TO LBL
      GO TO 50
   42 ZI2=ZIO
C CALCULATES THE WEIGHTING COEFFICIENTS FOR EXTRAPOLATED VALUES.
   45 IP1=IPL(IL1T3-2)
      IP2=IPL(IL1T3-1)
      IP3=IPL(IL2T3-1)
      X1=XD(IP1)
      Y1=YD(IP1)
      X2=XD(IP2)
      Y2=YD(IP2)
      X3=XD(IP3)
      Y3=YD(IP3)
      DX02=XI0-X2
      DY02=YI0-Y2
      DX32=X3-X2
      DY32=Y3-Y2
      DX21=X2-X1
      DY21=Y2-Y1
      W1=(DX02*DX32+DY02*DY32)**2/(DX32*DX32+DY32*DY32)
      W2=(DX02*DX21+DY02*DY21)**2/(DX21*DX21+DY21*DY21)
C CALCULATES ZII AS A WEIGHTED MEAN.
   46 ZII=(W1*ZI1+W2*ZI2)/(W1+W2)
      RETURN
C INTERNAL ROUTINE FOR PUNCTUAL INTERPOLATION.
C CHECKS IF THE NECESSARY CFO VALUES ARE SAVED.
   50 IF(NCF.EQ.0)          GO TO 60
      JCF=-27
      DO 51  LCF=1,NCF
        JCF=JCF+28
        FLITJ=CF(JCF)
        IF(ITO.EQ.ITJ)      GO TO 70
   51 CONTINUE
C CALCULATION OF NEW CFO VALUES.
C DETERMINES THE COEFFICIENTS FOR THE COORDINATE SYSTEM TRANS-
C FORMATION FROM THE X-Y SYSTEM TO THE U-V SYSTEM, AND CALCU-
C LATES THE COEFFICIENTS OF THE POLYNOMIAL FOR INTERPOLATION.
C LOADS COORDINATE AND PARTIAL DERIVATIVE VALUES AT THE
C VERTEXES.
   60 JIPT=3*(ITO-1)
      JPD=0
      DO 62  I=1,3
        JIPT=JIPT+1
        IDP=IPT(JIPT)
        X(I)=XD(IDP)
        Y(I)=YD(IDP)
```

```
      Z(I)=ZD(IDP)
      JPDD=5*(IDP-1)
      DO 61  KPD=1,5
        JPD=JPD+1
        JPDD=JPDD+1
        PD(JPD)=PDD(JPDD)
   61   CONTINUE
   62 CONTINUE
C DETERMINING THE COEFFICIENTS FOR THE COORDINATE SYSTEM
C TRANSFORMATION FROM THE X-Y SYSTEM TO THE U-V SYSTEM
C AND VICE VERSA
   63 X0=X(1)
      Y0=Y(1)
      A=X(2)-X0
      B=X(3)-X0
      C=Y(2)-Y0
      D=Y(3)-Y0
      AD=A*D
      BC=B*C
      DLT=AD-BC
      AP= D/DLT
      BP=-B/DLT
      CP=-C/DLT
      DP= A/DLT
C CONVERSION OF THE PARTIAL DERIVATIVES AT THE VERTEXES OF THE
C TRIANGLE FOR THE U-V COORDINATE SYSTEM
   64 AA=A*A
      ACT2=2.0*A*C
      CC=C*C
      AB=A*B
      ADBC=AD+BC
      CD=C*D
      BB=B*B
      BDT2=2.0*B*D
      DD=D*D
      DO 65  I=1,3
        JPD=5*I
        ZU(I)=A*PD(JPD-4)+C*PD(JPD-3)
        ZV(I)=B*PD(JPD-4)+D*PD(JPD-3)
        ZUU(I)=AA*PD(JPD-2)+ACT2*PD(JPD-1)+CC*PD(JPD)
        ZUV(I)=AB*PD(JPD-2)+ADBC*PD(JPD-1)+CD*PD(JPD)
        ZVV(I)=BB*PD(JPD-2)+BDT2*PD(JPD-1)+DD*PD(JPD)
   65 CONTINUE
C CALCULATION OF THE COEFFICIENTS OF THE POLYNOMIAL
   66 P00=Z(1)
      P10=ZU(1)
      P01=ZV(1)
      P20=0.5*ZUU(1)
      P11=ZUV(1)
      P02=0.5*ZVV(1)
      H1=Z(2)-P00-P10-P20
      H2=ZU(2)-P10-ZUU(1)
      H3=ZUU(2)-ZUU(1)
      P30= 10.0*H1-4.0*H2+0.5*H3
      P40=-15.0*H1+7.0*H2     -H3
      P50=  6.0*H1-3.0*H2+0.5*H3
      H1=Z(3)-P00-P01-P02
      H2=ZV(3)-P01-ZVV(1)
      H3=ZVV(3)-ZVV(1)
      P03= 10.0*H1-4.0*H2+0.5*H3
      P04=-15.0*H1+7.0*H2     -H3
      P05=  6.0*H1-3.0*H2+0.5*H3
      LU=SQRT(AA+CC)
      LV=SQRT(BB+DD)
      THXU=ATAN2(C,A)
```

```
      THUV=ATAN2(D,B)-THXU
      CSUV=COS(THUV)
      P41=5.0*LV*CSUV/LU*P50
      P14=5.0*LU*CSUV/LV*P05
      H1=ZV(2)-P01-P11-P41
      H2=ZUV(2)-P11-4.0*P41
      P21= 3.0*H1-H2
      P31=-2.0*H1+H2
      H1=ZU(3)-P10-P11-P14
      H2=ZUV(3)-P11-4.0*P14
      P12= 3.0*H1-H2
      P13=-2.0*H1+H2
      THUS=ATAN2(D-C,B-A)-THXU
      THSV=THUV-THUS
      SNUV=SIN(THUV)
      LUSNUV=LU*SNUV
      LVSNUV=LV*SNUV
      AA= SIN(THSV)/LUSNUV
      BB=-COS(THSV)/LUSNUV
      CC= SIN(THUS)/LVSNUV
      DD= COS(THUS)/LVSNUV
      AC=AA*CC
      AD=AA*DD
      BC=BB*CC
      G1=AA*AC*(3.0*BC+2.0*AD)
      G2=CC*AC*(3.0*AD+2.0*BC)
      H1=-AA*AA*AA*(5.0*AA*BB*P50+(4.0*BC+AD)*P41)
     1   -CC*CC*CC*(5.0*CC*DD*P05+(4.0*AD+BC)*P14)
      H2=0.5*ZVV(2)-P02-P12
      H3=0.5*ZUU(3)-P20-P21
      P22=(G1*H2+G2*H3-H1)/(G1+G2)
      P32=H2-P22
      P23=H3-P22
C SAVES THE CF0 VALUES IN THE CF ARRAY.
   67 IF(NCF.LT.NCFMX)     NCF=NCF+1
      ICF=ICF+1
      IF(ICF.GT.NCFMX)     ICF=1
      JCF=28*ICF-27
      CF(JCF)=FLIT0
      DO 68  KCF=1,27
        JCF=JCF+1
        CF(JCF)=CF0(KCF)
   68 CONTINUE
      GO TO 80
C LOADS THE CF0 VALUES FROM THE CF ARRAY.
   70 DO 71  KCF=1,27
        JCF=JCF+1
        CF0(KCF)=CF(JCF)
   71 CONTINUE
C TRANSFORMATION OF THE COORDINATE SYSTEM FROM X-Y TO U-V
   80 DX=XII-X0
      DY=YII-Y0
      U=AP*DX+BP*DY
      V=CP*DX+DP*DY
C EVALUATION OF THE POLYNOMIAL
   85 P0=P00+U*(P10+U*(P20+U*(P30+U*(P40+U*P50))))
      P1=P01+U*(P11+U*(P21+U*(P31+U*P41)))
      P2=P02+U*(P12+U*(P22+U*P32))
      P3=P03+U*(P13+U*P23)
      P4=P04+U*P14
      ZI0=P0+V*(P1+V*(P2+V*(P3+V*(P4+V*P5))))
      GO TO LBL, (31,41,42)
      END
```

```
      GO TO 24
   22 FNDPPV=WK(1)
      FNNPPV=WK(2)
      IF(NDPO.NE.NDPPV)    GO TO 90
      IF(NNPO.NE.NNPPV)    GO TO 90
      IF(MDO.NE.3)         GO TO 24
   23 FNXIPV=WK(3)
      FNYIPV=WK(4)
      IF(NXIO.NE.NXIPV)    GO TO 90
      IF(NYIO.NE.NYIPV)    GO TO 90
      GO TO 30
   24 WK(3)=FNXIO
      WK(4)=FNYIO
C ALLOCATION OF STORAGE AREAS IN THE WK ARRAY.   (ALL MD)
   30 NDNDM1=NDPO*(NDPO-1)
      IWIPT=7
      IWIPL=IWIPT+NDNDM1
      IWIPN=IWIPL+NDNDM1
      IWPD =IWIPN+NDPO*NNPO
      IWIT =IWPD +NDPO*5
C DIVIDES THE X-Y PLANE INTO A NUMBER OF TRIANGLES AND
C DETERMINES NNP POINTS NEAREST EACH DATA POINT.   (MD=1)
   40 IF(MD.GT.1)      GO TO 42
   41 CALL IDGEOM(XD,YD,WK(IWIPT),WK(IWIPL),WK(IWIPN))
      WK(5)=FNT
      WK(6)=FNL
      GO TO 50
   42 FNT=WK(5)
      FNL=WK(6)
C ESTIMATES PARTIAL DERIVATIVES AT ALL DATA POINTS.   (ALL MD)
   50 CALL IDPDRV(XD,YD,ZD,WK(IWIPN),WK(IWPD))
C LOCATES ALL INTERPOLATED POINTS.   (MD=1,2)
   60 IF(MDO.EQ.3)    GO TO 70
      IXI=0
      JWIT=IWIT-1
      INC=-1
      DO 62  IYI=1,NYIO
        INC=-INC
        YII=YI(IYI)
        DO 61  IXIO=1,NXIO
          IXI=IXI+INC
          JWIT=JWIT+INC
          CALL IDLCTN(XD,YD,WK(IWIPT),WK(IWIPL),
     1                XI(IXI),YII,WK(JWIT))
   61   CONTINUE
        IXI=IXI+INC
        JWIT=JWIT+INC+NXIO
   62 CONTINUE
C INTERPOLATION OF THE ZI VALUES.   (ALL MD)
   70 NCF=0
      ICF=0
      JWIT=IWIT-1
      IXI=0
      IZI=0
      INC=-1
      DO 72  IYI=1,NYIO
        INC=-INC
        YII=YI(IYI)
        DO 71  IXIO=1,NXIO
          JWIT=JWIT+INC
          IXI=IXI+INC
          IZI=IZI+INC
          CALL IDPTIP(XD,YD,ZD,WK(IWIPT),WK(IWIPL),WK(IWPD),
     1                WK(JWIT),XI(IXI),YII,ZI(IZI))
   71   CONTINUE
```

```
      SUBROUTINE  IDSFFT(MD,NDP,XD,YD,ZD,WK,NXI,NYI,XI,YI,ZI)
C THIS SUBROUTINE PERFORMS SMOOTH SURFACE FITTING WHEN THE PRO-
C JECTIONS OF THE DATA POINTS IN THE X-Y PLANE ARE IRREGULARLY
C DISTRIBUTED IN THE PLANE.
C THE INPUT PARAMETERS ARE
C     MD  = MODE OF COMPUTATION (MUST BE 1, 2, OR 3),
C         = 1 FOR NEW XD-YD,
C         = 2 FOR OLD XD-YD, NEW XI-YI,
C         = 3 FOR OLD XD-YD, OLD XI-YI,
C     NDP = NUMBER OF DATA POINTS (MUST BE 4 OR GREATER),
C     XD  = ARRAY OF DIMENSION NDP STORING THE X COORDINATES
C           OF THE DATA POINTS,
C     YD  = ARRAY OF DIMENSION NDP STORING THE Y COORDINATES
C           OF THE DATA POINTS,
C     ZD  = ARRAY OF DIMENSION NDP STORING THE Z COORDINATES
C           OF THE DATA POINTS,
C     WK  = ARRAY OF DIMENSION (2*NDP+NNP+5)*NDP+NXI*NYI
C           TO BE USED AS A WORK AREA,
C     NXI = NUMBER OF OUTPUT GRID POINTS IN THE X COORDINATE
C           (MUST BE 1 OR GREATER),
C     NYI = NUMBER OF OUTPUT GRID POINTS IN THE Y COORDINATE
C           (MUST BE 1 OR GREATER),
C     XI  = ARRAY OF DIMENSION NXI STORING THE X COORDINATES
C           OF THE OUTPUT GRID POINTS,
C     YI  = ARRAY OF DIMENSION NYI STORING THE Y COORDINATES
C           OF THE OUTPUT GRID POINTS.
C WHERE NNP IS THE NUMBER OF ADDITIONAL DATA POINTS USED FOR
C ESTIMATING PARTIAL DERIVATIVES AT EACH DATA POINT.  THE VALUE
C OF NNP MUST BE GIVEN THROUGH THE IDNN COMMON.  NNP MUST BE 2
C OR GREATER, BUT SMALLER THAN NDP.
C THE OUTPUT PARAMETER IS
C     ZI  = DOUBLY-DIMENSIONED ARRAY OF DIMENSION (NXI,NYI),
C           WHERE THE INTERPOLATED Z VALUES AT THE OUTPUT
C           GRID POINTS ARE TO BE DISPLAYED.
C THE LUN CONSTANT IN THE DATA INITIALIZATION STATEMENT IS THE
C LOGICAL UNIT NUMBER OF THE STANDARD OUTPUT UNIT AND IS,
C THEREFORE, SYSTEM DEPENDENT.
C DECLARATION STATEMENTS
      DIMENSION   XD(10),YD(10),ZD(10),WK(1000),
     1            XI(10),YI(10),ZI(100)
      COMMON/IDNN/NNP
      COMMON/IDGM/NDPC,NNPC,NT,NL
      COMMON/IDPI/NCF,ICF
      EQUIVALENCE (FNDPO,NDPO),(FNDPPV,NDPPV),
     1            (FNNPO,NNPO),(FNNPPV,NNPPV),
     2            (FNXIO,NXIO),(FNXIPV,NXIPV),
     3            (FNYIO,NYIO),(FNYIPV,NYIPV),
     4            (FNT,NT),(FNL,NL)
      DATA  LUN/6/
C SETTING OF SOME INPUT PARAMETERS TO LOCAL VARIABLES.  (ALL MD)
   10 MDO=MD
      NDPO=NDP
      NDPC=NDPO
      NXIO=NXI
      NYIO=NYI
      NNPO=NNP
      NNPC=NNPO
C ERROR CHECK.  (ALL MD)
   20 IF(MDO.LT.1.OR.MDO.GT.3)           GO TO 90
      IF(NDPO.LT.4)         GO TO 90
      IF(NXIO.LT.1.OR.NYIO.LT.1)     GO TO 90
      IF(NNPO.LT.2.OR.NNPO.GE.NDPO)  GO TO 90
      IF(MDO.NE.1)          GO TO 22
   21 WK(1)=FNDPO
      WK(2)=FNNPO
```

```
      JWIT=JWIT+INC+NXIO
      IXI=IXI+INC
      IZI=IZI+INC+NXIO
   72 CONTINUE
C NORMAL EXIT
   80 RETURN
C ERROR EXIT
   90 WRITE (LUN,2090) MDO,NDPO,NXIO,NYIO,NNPO
      RETURN
C FORMAT STATEMENT FOR ERROR MESSAGE
 2090 FORMAT(1X/41H ***   IMPROPER INPUT PARAMETER VALUE(S)./
     1   7H   MD =,I4,10X,5HNDP =,I6,10X,5HNXI =,I6,
     2   10X,5HNYI =,I6,10X,5HNNP =,I6/
     3   35H ERROR DETECTED IN ROUTINE    IDSFFT/)
      END
```

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

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

[X] UNLIMITED.

[ ] FOR OFFICIAL DISTRIBUTION.

18. Security Class *(This report)*
Unclassified

19. Security Class *(This page)*
Unclassified

20. Number of pages
58

21. Price:

U.S. GOVERNMENT PRINTING OFFICE 1975

GEOPHYSICS, VOL. 50, NO. 12 (DECEMBER 1985); P. 2831-2848, 9 FIGS., 5 TABLES.

Splines in geophysics

Pedro Gonzalez-Casanova* and Roman Alvarez‡

ABSTRACT

Modeling and contouring of geophysical data often require distributions of regularly spaced values. Splines have been shown to be the most accurate methods to obtain such distributions. We emphasize the general problem of interpolating random distributions of data on a given surface.

Splines are classified into unidimensional, quasi-bidimensional, and strictly bidimensional; based on this classification, a systematic derivation of the corresponding interpolating techniques is conducted. Two approaches are presented to obtain unidimensional splines: one based on the continuity of the first and second derivatives of the polynomials involved, and the other based on a variational approach. Quasi-bidimensional splines are constructed based on the unidimensional approach, while strictly bidimensional splines are generated by minimizing the bidimensional curvature. Quasi-bidimensional splines can be used for processing data distributions along nearly parallel lines; linear projections and parameterization are the techniques used in interpolating this type of distribution. Strictly bidimensional splines minimize curvature through the analytic solution of the Euler-Lagrange equation or by a finite-difference algorithm. The maximum error, mean error, and standard deviation between interpolated data and exact field values produced by various prisms show that quasi-bidimensional splines are 2.7 percent more accurate in the maximum error than strictly bidimensional splines when both techniques are applied to regularly spaced data. However, for irregularly spaced data, three examples containing 300, 600, and 900 random data points show the superiority of the thin-plate approach over the quasi-bidimensional splines. A comparison between various interpolation densities on regular grids, starting from a set of 327 randomly distributed magnetic stations, illustrates some differences between geophysically meaningful interpolations and interpolations carried out only for contouring purposes.

INTRODUCTION

When geophysical data are presented in the form of contours, automatic contouring is often performed by interpolating a distribution of regularly spaced discrete data points. Since field data are seldom regularly spaced, methods must be provided for obtaining a regular network from irregularly spaced data.

Interpolating irregularly spaced field data by hand is essentially a linear process in which the concept of maximum smoothness is implicitly applied. Mathematically, the continuity of a function and its first and second derivatives at discrete, observed points lead to contour smoothness.

Crain and Bhattacharyya (1967) introduced qualitative criteria to evaluate accuracy versus computation time. The highest quantitative accuracy was attained by least-squares methods using orthogonal polynomials generated by the Gram-Schmidt method. Subsequently, Bhattacharyya (1969) proposed a method of interpolation based on cubic-order surfaces; he concluded this method was more accurate than using orthogonal polynomials. Various papers followed this approach and introduced variations to such an algorithm (Thomson, 1970; Heissing et al., 1972; Rasmussen and Sharma, 1979).

Crain (1970) established a two-fold classification scheme of interpolation methods: (1) methods of mathematical surfaces, and (2) methods of numerical surfaces. The first group produces an analytical surface passing through a set of observed points, while the second group develops interpolated points, from the distribution of neighboring, observed points.

SPLINE CLASSIFICATION

A thin metallic strip when flexed and forced to pass through a set of points in the XY plane generates a unidimensional spline. Reinsch (1967) showed that such a curve corresponds to a set of cubic polynomials, each describing the curve between two successive observed values. These polynomials

Manuscript received by the Editor November 28, 1984; revised manuscript received April 4, 1985.
*Instituto de Geofisica, Universidad Nacional Autonoma de Mexico, 04510 Mexico, D. F., Mexico.
‡Instituto de Investigaciones, en Matematicas Aplicadas y en Sistemas, Universidad Nacional Autonoma de Mexico, 01000 Mexico, D. F., Mexico.

maintain continuity of the cubic functions and their first and second derivatives at every point (joint) in which the polynomials meet. Bicubic splines contain essentially the same concept. The metallic plate is assumed deformed by forces acting at each observed point applied to fit it; boundary conditions are required to fix the perimeter of the plate.

Since 1960 splines have been used for the interpolation of irregularly spaced geophysical data along a line (e.g., Ahlberg et al., 1967). Spline interpolation also found its way into bidimensional interpolation starting from regular grids (Bhattacharyya, 1969; Heissing et al., 1972; Dooley, 1976). Other algorithms have been proposed for the interpolation of irregularly spaced data on a plane (e.g., Briggs, 1974). Rasmussen and Sharma (1979) brought up the problem of precise interpolation in terms of absolute errors. They evaluated spline interpolation and concluded that for various causative bodies and conditions, spline interpolation yielded excellent results. However, their results are for data aligned along parallel lines, not for completely random data points. The distinction is important since the same algorithms do not work properly for random data.

Here we emphasize interpolation of truly random distributions of data points on a given surface using variational and minimum-curvature approaches. We base the work on that of Duchon (1975), Paihua and Utrera (1976), and Wahba and Wendelberger (1980) who proposed a substantially different construction of the analytical surfaces involved. Their algorithms are only beginning to find their way into the treatment of geophysical data.

For clarity, we classify splines as: (1) unidimensional, (2) quasi-bidimensional, and (3) strictly bidimensional. Quasi-bidimensional splines correspond to data roughly aligned along straight lines, such as flight lines, while strictly bidimensional splines correspond to distributions of truly random data.

UNIDIMENSIONAL SPLINES

We review two approaches for generation of unidimensional splines. The first approach adjusts a curve of cubic polynomials to n points distributed in the XY plane. The second approach uses a thin metallic strip flexed in such a way that it passes through all the points of the distribution. According to Hamilton's principle of least action, the curve obtained will be the one that minimizes the energy of flexure of the metallic strip.

Cubic splines

We consider a distribution of points $x_i (i = 1, \ldots, n)$ with $x_{i-1} < x_i$, and a set of functions $\{u(x)\}$ that take values u_i at such points. The interpolation function $u(x)$, as pointed out previously, is formed by cubic polynomials at each interval $[x_{i-1}, x_i]$. We call $P_i = u_i'(x)$ the first derivative of the function $u_i(x)$ at a joint. The cubic polynomial between two joints can be properly described if its four coefficients are known. Assuming the values of the function and its first derivatives are known at two successive joints, this procedure yields four linear equations that can be solved for the four unknown coefficients.

At two successive joints a and b

$$C(a) = \alpha_0,$$

$$C'(a) = \alpha_1,$$

$$C(b) = \alpha_0 + \alpha_1(b-a) + \alpha_2(b-a)^2 + \alpha_3(b-a)^3, \quad (1)$$

and

$$C'(b) = \alpha_1 + 2\alpha_2(b-a) + 3\alpha_3(b-a)^2.$$

Let $C = \underset{\sim}{M}\alpha$ be the corresponding matrix equation. Solving the system

$$\alpha = (\underset{\sim}{M}^{-1})C$$

and substituting the values of α in the general formula for a cubic polynomial, we obtain

$$\begin{aligned} C(x) = {} & C(a) + C'(a)(x-a) \\ & + \left[3\frac{C(b)-C(a)}{(b-a)^2} - \frac{C'(b)+2C'(a)}{(b-a)}\right](x-a)^2 \\ & + \left[\frac{-2}{(b-a)^3}(C(b)-C(a)) + \frac{C'(a)+C'(b)}{(b-a)^2}\right](x-a)^3. \quad (2) \end{aligned}$$

Expression (2) corresponds to the polynomial between joints a and b. Therefore, given the values of a function at two successive points, as well as its first derivatives evaluated at such points, there is one and only one cubic polynomial that passes through them. Thus with n points and their first derivatives, there is one and only one set of cubic polynomials passing through them.

However, we assume that at each u_i the value of the first derivative is also known. In practice this is a restrictive condition since normally the field values (e.g., magnetic, gravity) are known but not the values of their derivatives (i.e., the derivatives at the joints). However, this can be overcome assuming that (1) the first derivatives at the first and last points are known, and (2) the curve formed by the polynomials is continuous in the second derivative. The latter condition results in a new set of linear equations from which we can obtain the first derivatives at intermediate points $P_2, \ldots, P_{n-1}$.

Take three successive points x_0, x_1, and x_2 and let $v(x)$ and $w(x)$ be two cubic polynomials that satisfy

$$u_1 = v(x_1) = w(x_1) \quad \text{and} \quad P_1 = v'(x_1) = w'(x_1). \quad (3)$$

Differentiating equation (2) twice and making $a = x_1$ and $b = x_0$, and equating $v''(x_1)$ to $w''(x_1)$, we obtain

$$\begin{aligned} & \Delta x_1 v'(x_0) + 2(\Delta x_1 + \Delta x_0)P_1 + \Delta x_0 w'(x_2) \\ & = 3\left\{\frac{\Delta x_0}{\Delta x_1}\left[w(x_2) - w(x_1) + \frac{\Delta x_1}{\Delta x_0}v(x_1) - v(x_0)\right]\right\}. \end{aligned}$$

A recurrence relation is obtained in the following form; it allows computation of intermediate derivatives:

$$\begin{aligned} & \Delta x_i P_{i-1} + 2(\Delta x_i + \Delta x_{i-1})P_i + \Delta x_{i-1}P_{i+1} \\ & = 3\left[\Delta x_{i-1}\frac{\Delta u_i}{\Delta x_i} + \Delta x_i \frac{\Delta u_{i-1}}{\Delta x_{i-1}}\right], \qquad i = 1, \ldots, I-1. \quad (4) \end{aligned}$$

Such a relationship is linear with $I-1$ unknowns in P_i, with $i = 1, \ldots, I-1$; it has a tridiagonal matrix that is strictly diagonal dominant. This is easily shown by dividing equation (4) by $(\Delta x_i + \Delta x_{i-1})$ and writing the corresponding matrix equation. Since

$$\frac{\Delta x_{i-1}}{\Delta x_i + \Delta x_{i-1}} = 1 - \frac{\Delta x_i}{\Delta x_{i-1} + \Delta x_i},$$

and since a_{ij} are the elements of the coefficient's matrix, then

$$|a_{ii}| > \sum_{j \neq i} |a_{ij}| \quad (\text{i.e., } |a_{ii}| = 2$$

and

$$\sum_{j \neq i} |a_{ij}| = 1).$$

Thus, the only additional condition for the matrix to be diagonal dominant is that the off-diagonal elements in the first and last rows, are less than 2. In this case equation (4) contains nonzero eigenvalues (Marcus, 1960). Consequently, we are dealing with a singular matrix with $I - 1$ linearly independent equations, implying that the $P_i (i = 1, \ldots, I - 1)$ unknowns are uniquely determined. Thus, demanding continuity of the first and second derivatives at the joints results in a piece-wise smooth, cubic polynomial, with values u_i in $u(x_i)$.

Variational approach

According to Hamilton's principle of least action, a flexed metallic strip adopts the shape that minimizes the flexing energy. This energy is proportional to the square of the curvature of the strip. The integral of action J is given by

$$J = \int (g'')^2 \, dx. \tag{5}$$

The action J has to be minimized, yielding a function $g(x_i) = y_i$ $(i = 0, \ldots, n)$, such that y_i are known field values and x_i are the coordinates of these values. If we let $g(x, \alpha) = g(x, 0) + \alpha h(x)$ be the variated trajectory, then the condition for finding an extremum is

$$\left[\frac{\partial J(\alpha)}{\partial \alpha}\right]_{\alpha=0} = 0. \tag{6}$$

We could now directly obtain the Euler-Lagrange equation, forcing the strip to pass exactly through the observed points. However, we take advantage of the simplicity of the unidimensional case and introduce the concept of smoothing in the variational approach.

If the strip passes not exactly on the points but in their neighborhood, the strip deformation is smaller and the corresponding curve is smoother. The smoothness depends upon the radius of the neighborhood; this condition is expressed mathematically by

$$\sum_{i=0}^{n} \left(\frac{g(x_i) - y_i}{\delta y_i}\right) \leq s, \tag{7}$$

where s is a factor that multiplies the radii of all the neighborhoods. The parameter δy_i controls the degree of smoothness at each observed point. The inequality (7) can be expressed as an equality with the aid of an additional, arbitrary parameter z:

$$\sum_{i=0}^{n} \left(\frac{g(x_i) - y_i}{\delta y_i}\right)^2 = s - z^2. \tag{8}$$

Therefore, z represents how close the deformation lies to y_i; the smaller z, the closer the deformation to y_i.

The function $g(x)$ minimizes J and complies with the restrictions imposed on the strip; thus, from equations (5) and (8)

$$J = \int_{x_o}^{x_n} [g''(x)]^2 \, dx + P\left[\sum_{i=0}^{n} \left(\frac{g(x_i) - y_i}{\delta y_i}\right)^2 + z^2 - s\right], \tag{9}$$

where P represents a Lagrangian multiplier.

By demanding that the additional condition

$$\frac{\partial J(z)}{\partial z} = 0 \tag{10}$$

is met, the strip is forced to be in the neighborhood (δy_i) of, but as close as possible to, each y_i.

From equations (6), (9), and (10) and assuming that the derivatives at the end of the strip are zero (i.e., the strip is fixed horizontally at the extrema), we obtain

$$\begin{aligned} \int_{x_0}^{x_n} (g'''' h(x) \, dx &+ \sum_{i=1}^{n-1} h'(x_i)[g''(x_i)_- - g''(x_i)_+] \\ &+ \sum_{i=1}^{n-1} h(x_i)\left[(g'''(x_i)_+ - g'''(x_i)_-) \right. \\ &\left. + 2P\left(\frac{g(x_i) - y_i}{\delta y_i^2}\right)\right] = 0. \end{aligned} \tag{11}$$

However, since the function $h(x)$ is arbitrary, the terms in $h(x_i)$ and $h'(x_i)$ are linearly independent. On the other hand, summations contain only discrete values of h, while the integral contains all the values of $h(x)$ between values x_0 and x_n. The only possibility for the terms to be zero in equation (11) is that they are zero independently.

Consequently, we obtain from the first term

$$g''''(x) = 0 \quad \text{(Euler-Lagrange's equation)}, \tag{12}$$

and from the second and third terms

$$\begin{aligned} -g'''(x_i)_+ + g'''(x_i)_- &= 2P\left(\frac{g(x_i) - y_i}{\delta y_i^2}\right), \\ g''(x_i)_+ - g''(x_i)_- &= 0, \\ g'(x_i)_+ - g'(x_i)_- &= 0, \\ \text{and} \qquad g(x_i)_+ - g(x_i)_- &= 0. \end{aligned} \tag{13}$$

One solution of equation (12) is

$$g_{x_i}(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3. \tag{14}$$

Determination of the coefficients of this polynomial is made starting from the continuity conditions in equation (13). From the continuity of the second derivative we have

$$d_i = \frac{c_{i+1} - c_i}{3h_i} \tag{15}$$

and from the continuity of $g(x)$ we have

$$b_i = \frac{a_{i+1} - a_i}{h_i} - c_i h_i - d_i h_i^2. \tag{16}$$

Thus, we define

$$h_i = x_{i+1} - x_i,$$

$$\mathbf{C} = (C_1, \ldots, C_{n-1})^T,$$

$$y = (y_0, \ldots, y_n)^T,$$

$$a = (a_0, \ldots, a_n)^T,$$

and

$$\underset{\sim}{D} = \text{diag}\,(\delta y_0, \ldots, \delta y_n).$$

$\underset{\sim}{T}$ is a tridiagonal matrix of order $n - 1$ with terms

$$t_{i,i} = 2(h_{i-1} + h_i)/3,$$

and

$$t_{i,i+1} = t_{i+1,i} = h_i/3,$$

and $\underset{\sim}{Q}$ is a tridiagonal matrix with terms

$$q_{i-1,i} = 1/h_{i-1};\quad q_{ii} = -1/h_{i-1} - 1/h_i;\quad q_{i+1,i} = 1/h_i.$$

We obtain from the continuity of the first and third derivatives

$$(\underset{\sim}{Q}^T \underset{\sim}{D}^2 \underset{\sim}{Q} + P\underset{\sim}{T})C = P\underset{\sim}{Q}^T y, \tag{17}$$

and

$$a = y - P^{-1}\underset{\sim}{D}^2 \underset{\sim}{Q}\underset{\sim}{C}. \tag{18}$$

Thus, if we know the value of the Lagrangian parameter P, from equation (17) we can obtain the vector C, and from equation (18) we obtain vector a. The remaining coefficients d_i and b_i can be calculated from the recurrence formulas (15) and (16). We have thus determined the values of the coefficients of the cubic polynomial at each interval.

From equations (8) and (10) we obtain

$$Pz = 0, \tag{19}$$

and defining a function $F(P)$ as

$$F(P) = \underset{\sim}{D}\underset{\sim}{Q}(\underset{\sim}{Q}^T \underset{\sim}{D}^T \underset{\sim}{D}^2 \underset{\sim}{Q} + P\underset{\sim}{T})^{-1} \underset{\sim}{Q}^T y, \tag{20}$$

equation (8) can be written as

$$F^2(P) = s - z^2. \tag{21}$$

From equation (19) two possible solutions are obtained: $P = 0$ or $z = 0$. In the former, boundary conditions do not affect J and the polynomial is reduced to a straight line. In the latter, and using equation (21), the polynomial reduces to

$$F(P) = s^{1/2}. \tag{22}$$

The solution of equation (22) lets us get the value of P. With this last step the parameters necessary to interpolate are completely determined.

In summary, equations (15), (16), and (18) represent the algorithm for obtaining 1-D interpolations with the variational (i.e., cubic functions) approach. Equation (12) shows that one possible solution of the functional of equation (5) is a set of cubic splines. Obviously, equation (12) can yield other solutions.

QUASI-BIDIMENSIONAL SPLINES

The quasi-bidimensional method deals with obtaining a cubic, bidimensional surface, starting from one-dimensional (1-D) splines. Assume that we are dealing with a series of straight lines, parallel to the x-axis but irregularly spaced in the y-direction. Along the x-axis, 1-D splines can be fit to the data to interpolate and regularize in that direction. We then have equally spaced data (at intervals Δx) on those lines. A second interpolation in the y-direction, along fixed values of x and with a Δy not necessarily equal to Δx, will yield a regularly spaced grid.

These operations are mathematically equivalent to performing a composition of orthogonal splines (tensor products)

$$u(x, y) = \sum_{m=0}^{I+2} \sum_{n=0}^{J+2} \beta_{mn} \phi_m(x) \psi_n(y), \tag{23}$$

where $u(x, y)$ represents the bidimensional surface, β_{mn} is a set of coefficients, and ϕ_m and ψ_n are piece-wise smooth, cubic functions of class C^2 (De Boor, 1962).

From the previous section, given n points $(u_1, \ldots, u_n)$ and the derivatives (P_0, P_n) at both ends of the interval, there is one and only one unidimensional spline that passes through these points. Thus, given a point in $\mathbb{R}^{n+2}$, there is one and only one associated unidimensional spline. Since $\mathbb{R}^{n+2}$ is a linear space and since there is an isomorphism between $\mathbb{R}^{n+2}$ and the set of unidimensional splines $S(x; x_0, \ldots, x_I)$, this set is also linear. Equation (23) involves the product of two linear spaces; the space of splines in the x-direction ϕ_m and the space of splines in the y-direction, ψ_n. These spaces have dimensions $(I + 3)$ and $(J + 3)$, respectively. The product space has dimension $(I + 3)$ $(J + 3)$ and constitutes the linear space of bidimensional surfaces. However, if the linear spaces ϕ_m and ψ_n are not orthogonal, as in the case of nonparallel flight lines, the tensor product does not yield a linear space. This is the main source of error in the quasi-bidimensional approach.

De Boor (1962) showed that the cubic splines given by equation (23) are uniquely determined if the following values are known:

$$\begin{aligned} U_{ij} &= u(x_i, y_j) \quad i = 0, \ldots, I;\ \ j = 0, \ldots, J \\ P_{ij} &= u_x(x_i, y_j) \quad i = 0, I; \qquad j = 0, \ldots, J \\ q_{ij} &= u_y(x_i, y_j) \quad i = 0, \ldots, I;\ \ j = 0, J \\ S_{ij} &= u_{xy}(x_i, y_j)\ i = 0, I; \qquad j = 0, J. \end{aligned} \tag{24}$$

Assuming that the values of the polynomial $u(x, y)$ were known, as well as its derivatives in x, y, and xy, De Boor (1962) obtained a matrix equation for determination of the β_{mn} coefficients of the polynomial surface

$$\underset{\sim}{A}\underset{\sim}{K}\underset{\sim}{A}' = \underset{\sim}{\beta}. \tag{25}$$

The matrix $\underset{\sim}{K}$ contains values of the polynomial and its derivatives, while $[\underset{\sim}{A}]$ is formed by elements that depend upon the coordinates of the regular, but not equally spaced, grid.

Similar to the unidimensional case, the intermediate derivatives with respect to x, y, and xy are obtained by the recurrence formula derived from the continuity of the second derivative:

$$\Delta Y_{j-1} Q_{i,j+1} + 2(\Delta Y_{j-1} + \Delta Y_j) Q_{ij} + \Delta Y_j Q_{i,j-1}$$
$$= 3\left[\frac{\Delta Y_{j-1}}{\Delta Y_j}(U_{i,j+1} - U_{ij}) + \frac{\Delta Y_j}{\Delta Y_{j-1}}(U_{ij} - U_{i,j-1})\right],\quad j = 1, \ldots, J-1.$$

$$\Delta Y_{j-1} S_{i,j+1} + 2(\Delta Y_{j-1} + \Delta Y_j) S_{ij} + \Delta Y_j S_{i,j-1}$$
$$= 3\left[\frac{\Delta Y_{j-1}}{\Delta Y_j}(P_{i,j+1} - P_{ij}) + \frac{\Delta Y_j}{\Delta Y_{j-1}}(P_{ij} - P_{i,j-1})\right], \quad j = 1, \ldots, J-1,$$

$$\Delta X_{i-1} P_{i+1,j} + 2(\Delta X_{i-1} + \Delta X_i) P_{ij} + \Delta X_i P_{i-1,j}$$
$$= 3\left[\frac{\Delta X_{i-1}}{\Delta X_i}(U_{i+1,j} - U_{ij}) + \frac{\Delta X_i}{\Delta X_{i-1}}(U_{ij} - U_{i-1,j})\right] \quad i = 1, \ldots, I-1,$$

and

$$\Delta X_{i-1} S_{i+1,j} + 2(\Delta X_{i-1} + \Delta X_i) S_{ij} + \Delta X_i S_{i-1,j}$$
$$= 3\left[\frac{\Delta X_{i-1}}{\Delta X_i}(Q_{i+1,j} - Q_{ij}) + \frac{\Delta X_i}{\Delta X_{i-1}}(Q_{ij} - Q_{i-1,j})\right], \quad i = 1, \ldots, I-1. \quad (26)$$

This approach is used to obtain the splines if the field values are regularly distributed, since the only requirement for the validity of the algorithm is that the 1-D splines lie along straight lines. However, it is well-known that in geophysical exploration such a requisite is seldom fulfilled, although good approximations are sometimes obtained in aeromagnetic flight lines. Generally the lack of parallelism and the lack of straightness as sources of error must be considered.

Two approaches were followed to minimize the above difficulties: (1) correcting the data on the basis of least-squares and linear projections (Bhattacharyya, 1969; Rasmussen and Sharma, 1979), and (2) correcting with parametric functions (Heissing et al., 1972; Rasmussen and Sharma, 1979). The first approach uses straight lines, while the second uses polygons approaching straight lines. In the second case interpolation is performed with the original field values, although slope dispersion (i.e., the range of bearing variations along a flight line) is high and inconvenient. In the first approach fitting a straight line and projecting the original values onto it is the source of error. We briefly discuss both methods.

Linear projections

This approach considers data points distributed in the vicinity of straight, parallel lines. The algorithm, proposed in Bhattacharyya (1969), deals with aeromagnetic data using De Boor's tensor product formulation. The coefficients of the straight lines are obtained by least-square fits of the data sets, and then the data points that do not lie on the straight line are projected linearly onto it by the following procedure.

(1) Search for the datum point closest to the fitted line; such a point is called a pivot point and is imposed so its projection onto the line has the same field value F_{pl} (i.e., we assume that it lies on the line).

(2) Take the closest point to the pivot; then the field projected F_p onto the point on the straight line is given by (Figure 1):

$$F_p = (d_2/d_1)F_0 + [(1 - d_2)/d_1]F_{pl}. \quad (27)$$

where d_1 and d_2 are distances projected from the field point to the pivot and from the projected point to the pivot, respectively. F_0 is the field corresponding to the point to be projected.

(3) The projected point is now considered a pivot and the process repeated for the next closest point.

Once the projections are performed, the tensor product of the splines expressed by equation (23) is readily obtained: to each straight line fit a unidimensional spline and obtain equally spaced points along the y-axis. Next, adjust straight lines parallel to the x-axis with splines and obtain equally spaced points in this direction.

As pointed out before, this algorithm introduces sources of error before De Boor's treatment can be applied: the point values on the straight lines are not actual field values but are projections.

Parametric splines

An irregular distribution of data points is the basis for producing a set of parametric bicubic splines (Ferguson, 1964; Heissing et al., 1972). The interpolating surface is formed by irregular, bicubic surfaces (Figure 2a). Each bicubic surface must preserve continuity of the function and of the first derivative with respect to contiguous surfaces. Figure 2b shows a bicubic surface $A_{i,j}$ in which two parametric coordinates u, v are defined along the directions shown; curves k_1 and k_2 are parameterized with respect to the variable u and curves k_3 and k_4 are parameterized with respect to v. Two points are located: $P(u)$ on k_1 and $Q(u)$ on k_2. An interpolation is made from $P(u)$ to $Q(u)$ with a curve segment parameterized in the coordinate v, defining the bicubic polynomial

$$P(u, v) = v^3\{-2[Q(u) - P(u)] + x(u) + y(u)\} + v^2\{3[Q(u) - P(u)] - 2x(u) + y(u)\} + vx(u) + P(u). \quad (28)$$

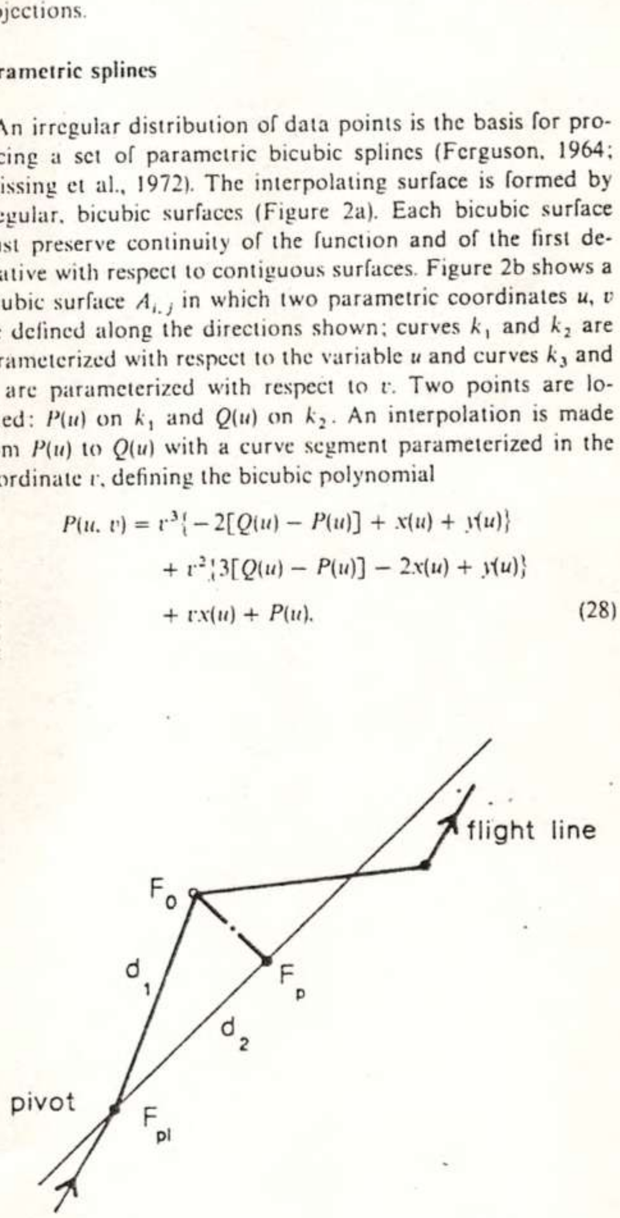


FIG. 1. The method of linear projections uses field values F_p that have been projected onto a straight line from the actual flight line.

where $x(u)$ and $y(u)$ are the tangents to the curves in the parametric direction v at points $P(u)$ and $Q(u)$. Both tangents are given by cubic interpolations in the u direction. Notice the similarity between equation (28) and equation (2) with $a = 0$ and $b = 1$ [i.e., equation (28) is normalized in the interval (0, 1)].

The bicubic polynomial (28) is expressed as

$$P(u, v) = \sum_{p=0}^{3} \sum_{q=0}^{3} u^p v^q R_{pq} \tag{29}$$

which corresponds to De Boor's tensorial product [equation (23)].

The 16 constants R_{pq} are determined through continuity of the first derivatives along the perimeter of the surfaces $A_{i,j}$ (Ferguson, 1964):

$$\begin{aligned}
R_{00} &= P_{ij}, \\
R_{01} &= S_{ij}, \\
R_{02} &= 3(P_{i,j+1} - P_{ij}) - (2S_{ij} + S_{i,j+1}), \\
R_{03} &= 2(P_{ij} - P_{i,j+1}) + (S_{ij} + S_{i,j+1}), \\
R_{10} &= T_{ij}, \\
R_{11} &= 0, \\
R_{12} &= 3(T_{i,j+1} - T_{ij}), \\
R_{13} &= 2(T_{ij} - T_{i,j+1}), \\
R_{20} &= 3(P_{i+1,j} - P_{ij}) - (2T_{ij} + T_{i+1,j}), \\
R_{21} &= 3(S_{i+1,j} - S_{ij}), \\
R_{22} &= 3[3(P_{i+1,j+1} - P_{i,j+1} + P_{ij} - P_{i+1,j}) \\
&\quad + 2(T_{ij} - T_{i,j+1}) + (T_{i+1,j} - T_{i+1,j+1}) \\
&\quad + 2(S_{ij} - S_{i+1,j}) + (S_{i,j+1} - S_{i+1,j+1})], \\
R_{23} &= 2[3(P_{i+1,j} - P_{ij} + P_{i,j+1} - P_{i+1,j+1}) \\
&\quad + 2(T_{i,j+1} - T_{ij}) + (T_{i+1,j+1} - T_{i+1,j})] \\
&\quad + 3(S_{i+1,j} + S_{i+1,j+1} - S_{ij} - S_{i,j+1}), \\
R_{30} &= 2(P_{ij} - P_{i+1,j}) + T_{ij} + T_{i+1,j}, \\
R_{31} &= 2(S_{ij} - S_{i+1,j}), \\
R_{32} &= 3[2(P_{i,j+1} - P_{i+1,j+1} + P_{i+1,j} - P_{ij}) \\
&\quad + (T_{i,j+1} + T_{i+1,j+1}) - (T_{ij} + T_{i+1,j})] \\
&\quad + 4(S_{i+1,j} - S_{ij}) + 2(S_{i+1,j+1} - S_{i,j+1}),
\end{aligned} \tag{30}$$

and

$$\begin{aligned}
R_{33} &= 2[2(P_{ij} - P_{i+1,j} + P_{i+1,j+1} - P_{i,j+1}) \\
&\quad + (T_{ij} + T_{i+1,j}) - (T_{i,j+1} + T_{i+1,j+1}) \\
&\quad + (S_{ij} - S_{i+1,j} + S_{i,j+1} - S_{i+1,j+1}),
\end{aligned}$$

where T_{ij} are derivatives with respect to the parametric direction u and S_{ij} are derivatives with respect to the parametric direction v.

Parametric splines eliminate the error introduced by the linear projections discussed previously; however, parameterization entails a much larger slope dispersion (i.e., the lines are not straight lines). Therefore, while one source of error is eliminated, another one is introduced. The parametric algorithm of Rasmussen and Sharma (1979) is a particular case of the one presented above and is not discussed here. Thus, the two approaches reviewed aim to adapt nonuniformly distributed data to a form suitable for the application of the tensor-product method proposed in De Boor (1962). In performing the adaptations, errors are introduced; the larger the departure of the data set from straight lines, the greater the error introduced in the interpolations.

STRICTLY BIDIMENSIONAL SPLINES

Two methods in the geophysical literature deal with strictly bidimensional splines—one method uses numerical surfaces, and the other uses mathematical surfaces. Numerical surfaces yield only approximate solutions; Briggs (1974) used a finite-difference method to find a solution. We show that the analytical solution offered by the mathematical surfaces yields better interpolations than those obtained with the numerical surfaces. In addition, the analytical method has as many solutions as there are kernels in the Euler-Lagrange equation in two dimensions. The pseudocubic splines are one such solution; however, a more precise approach is the thin-plate solution. We discuss the thin-plate solution in detail, since only a short note has appeared about its use in the treatment of geophysical data (Campos et al., 1983). These methods can interpolate adequately between sets of truly random data.

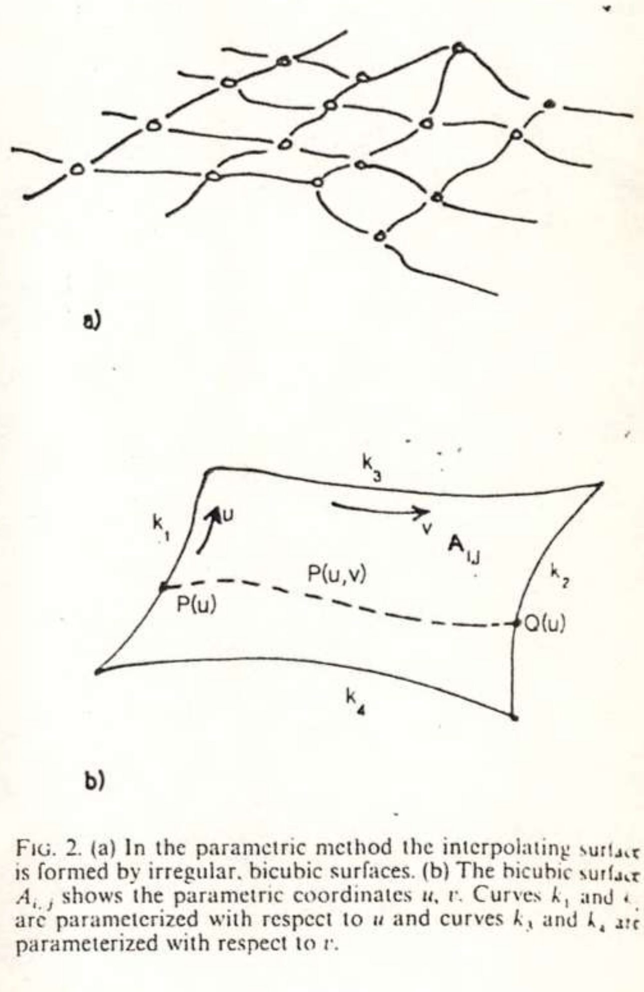


FIG. 2. (a) In the parametric method the interpolating surface is formed by irregular, bicubic surfaces. (b) The bicubic surface $A_{i,j}$ shows the parametric coordinates u, v. Curves k_1 and k_2 are parameterized with respect to u and curves k_3 and k_4 are parameterized with respect to v.

The thin-plate approach

The flexure energy of a curved plate is given by (e.g., Landau and Lifshitz, 1970, 46)

$$E = K \iint \left\{ \left(\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \right)^2 + 2(1 - \sigma) \left[\left(\frac{\partial^2 f}{\partial x \partial y} \right)^2 - \frac{\partial^2 f}{\partial x^2} \frac{\partial^2 f}{\partial y^2} \right] \right\} dx\, dy, \qquad (31)$$

where K is a constant that depends upon Young's modulus and Poisson's coefficient σ represents the transverse contraction of the plate under longitudinal tensile strain. Here σ varies between zero (metals) and 1/2 (rubber). The smoothness of the deformation depends upon the rigidity of the plate; since we are interested in the smoothest interpolation, we consider $\sigma = 0$ and $K = 1$. Thus equation (31) becomes

$$E = \iint \left[\left(\frac{\partial^2 f}{\partial x^2} \right)^2 + 2 \left(\frac{\partial^2 f}{\partial x \partial y} \right)^2 + \left(\frac{\partial^2 f}{\partial y^2} \right)^2 \right] dx\, dy. \qquad (32)$$

As for the 1-D problem of the deformed bar presented above, we require a function $g(x, y)$ that minimizes the functional equation (32) while taking the values $g(x_i, y_i)$ at coordinates (x_i, y_i). That is, the thin plate acquires the observed field values at their corresponding positions.

We can obtain the Euler-Lagrangian equation by a variational approach, just as we did for the 1-D case, although this procedure is more involved. However, another possible derivation is considerably more concise. The functional analysis aspects of such a derivation were emphasized in Duchon (1975); we emphasize here the minimum-norm concept and the boundary conditions problem, which enhances its applications aspects. Instead of finding a minimal trajectory, we minimize a norm in a given functional space [i.e., minimizing equation (32) is equivalent to minimizing a distance in such a space]. The resulting set of functions $g(x, y)$ fulfills the requirement; however, of those functions we use only the ones that take the (field) values $g(x_i, y_i)$, $i = 1, \ldots, n$, where n is the number of known (field) values. First, we deal with this problem of boundary conditions and subsequently we consider the problem of minimum curvature.

Boundary conditions for the plate. The interpolating surface can generally be represented by a function $g(x, y)$ belonging to a functional space X. The requirement that the plate pass through the set of observed points implies that $g(x, y)$ takes the values $g(x_i, y_i)$ for (x_i, y_i), $i = 1, \ldots, n$; thus the set $[g(x_1, y_1), \ldots, g(x_n, y_n)]$ belongs to $\mathbb{R}^n$. A transformation can be defined from the functional space X to $\mathbb{R}^n$ (Duchon, 1975):

$$A: \; X \to \mathbb{R}^n$$

with the property

$$A(g) = [g(x_1, y_1), \ldots, g(x_n, y_n)],$$

i.e., A links the function $g(x, y)$ in X to the point $[g(x_1, y_1), \ldots, g(x_n, y_n)]$ in $\mathbb{R}^n$. Figure 3 shows pertinent relationships between the two spaces X and $\mathbb{R}^n$ and the transformations A and A^{-1}. Assuming known field values, we have a vector $\mathbf{z}_0$ of $\mathbb{R}^n$. The set of functions that relates $g(x, y)$ to $\mathbf{z}_0$ by means of A is needed. $A^{-1}(\mathbf{z}_0)$ is the subset of functions that take the known field values; the function needed minimizes the curvature.

Curvature minimization.—As pointed out previously, the curvature must represent a norm in a functional space Y; the problem is solved by minimizing such a norm in Y.

The inner product of two vectors $\mathbf{y}$ and $\mathbf{y}'$ in Y is defined by

$$(\mathbf{y}, \mathbf{y}')_Y = \sum_{i,j=1}^{2} \int_{\mathbb{R}^2} y_{ij}(x, y) y'_{ij}(x, y)\, dx\, dy, \qquad (33)$$

where the space Y is formed by four-dimensional vectors $\mathbf{y} = (y_{11}, y_{12}, y_{21}, y_{22})$. These quadruplets are necessary since the curvature in A has four differential operators:

$$\frac{\partial^2}{\partial x^2}, \quad \frac{\partial^2}{\partial x \partial y}, \quad \frac{\partial^2}{\partial y \partial x}, \quad \text{and} \quad \frac{\partial^2}{\partial y^2}.$$

When $\mathbf{y} = \mathbf{y}'$, the internal product is reduced to the norm in Y,

$$|\mathbf{y}|_Y^2 = \sum_{i,j=1}^{2} \int_{\mathbb{R}^2} |y_{ij}(x, y)|^2\, dx\, dy, \qquad (34)$$

and if

$$y_{11} = \frac{\partial^2 g}{\partial x^2}, \qquad y_{12} = \frac{\partial^2 g}{\partial x\, \partial y}, \qquad y_{21} = \frac{\partial^2 g}{\partial y\, \partial x}, \qquad y_{22} = \frac{\partial^2 g}{\partial y^2},$$

then the norm is precisely the curvature.

An additional transformation T will be needed since the interpolating functions $g(x, y)$ are in space X, whereas the quadruplets $(y_{11}, y_{12}, y_{21}, y_{22})$ are in the Y space, where the norm (curvature) is minimized (Figure 3). Let

$$T: \; X \to Y$$

with

$$T(g) = \left(\frac{\partial^2 g}{\partial x^2}, \frac{\partial^2 g}{\partial x \partial y}, \frac{\partial^2 g}{\partial y \partial x}, \frac{\partial^2 g}{\partial y^2} \right).$$

Duchon (1975) showed that the set of functions denoted X has the structure of a Hilbert space if the inner product is given by

$$|g|_X^2 = |g(x_1, y_1)|^2 + |g(x_2, y_2)|^2 + |g(x_3, y_3)|^2 + \sum_{i,j} \int |D_{ij} g(x, y)|^2\, dx\, dy, \qquad (35)$$

where (x_1, y_1), (x_2, y_2), and (x_3, y_3) are three arbitrary, nonaligned points, and

$$D_{11} = \frac{\partial^2}{\partial x^2}, \quad D_{12} = \frac{\partial^2}{\partial x \partial y}, \quad D_{21} = \frac{\partial^2}{\partial y \partial x}, \quad \text{and} \quad D_{22} = \frac{\partial^2}{\partial y^2}.$$

Let $\mathbf{z}_0$ be the vector in space Z containing all the point values to which the plate must be deformed, and let $A^{-1}(\mathbf{z}_0)$ be the set of functions (linear variety) in X that fulfill the above restriction in Z. From this set let σ be the wanted solution. Thus, in the set $T[A^{-1}(\mathbf{z}_0)]$ the element of the set $T(\sigma)$ is closest to the zero of the Y-space (i.e., the element yielding the minimum norm). This statement can be expressed as an internal product by letting $T(x)$ be an element of $T[A^{-1}(0)]$:

$$\langle T(\sigma), T(x) \rangle_Y = 0,$$

where $\langle\ \rangle$ means interior product. We now find the equivalent property in X by changing the domain and the norm; thus

$$\langle T(\sigma), T(x) \rangle_Y = 0 \; \langle T^T T, \sigma x \rangle_X = 0.$$

We show that $T^T T\sigma$ may be expressed as a linear combination of the rows of A (i.e., when A is considered a matrix):

$$T^T T\sigma = \underset{\sim}{A}^T \lambda.$$

Thus

$$\langle T(\sigma), T(x)\rangle_Y = \langle A^T\lambda, x\rangle_X = \langle \lambda, Ax\rangle_Z, \qquad (36)$$

where $T(x)$ is now not necessarily in $T[A^{-1}(0)]$. Therefore, the minimization of the norm in Y has been related to a condition in space Z.

Let $A(g) = g(a_i)$ such that $g \in X$ and a_i are the plate deformations at (x_i, y_i). The term $g(a_i)$ is, of course, a point of $\mathbb{R}^n$; from equation (35)

$$\int D_{ik}\sigma(x, y) D_i D_k g(x, y)\, dx\, dy = \sum_{i=1}^{n} \lambda_i g(a_i). \qquad (37)$$

From the definition of Dirac's delta function and letting $\delta(t - a_i) = \delta a_i$ and $t = (x, y)$ be a point of $\mathbb{R}^2$, then the right-hand side can be written as

$$\sum_{i=1}^{n} \lambda_i g(a_i) = \langle \sum_i \lambda_i \delta_{a_i}, g\rangle,$$

from which (Gel'fand and Shilov, 1964)

$$\sum_{j,k} \langle D_j D_k \sigma, D_j D_k g\rangle = \sum_{j,k} \langle D_j^2 D_k^2 \sigma, g\rangle = \langle \Delta^2\sigma, g\rangle$$
$$= \langle \sum_i \lambda_i \delta_{a_i}, g\rangle, \qquad (38)$$

and finally

$$\Delta^2\sigma = \sum_{i=1}^{n} \lambda_i \delta_{a_i}. \qquad (39)$$

Equation (39) is the Euler-Lagrange equation of the functional (32).

Boundary conditions.—To determine the boundary conditions, we consider equation (37). The kernel of the bigradient is the set of all polynomials of first degree in $\mathbb{R}^2$. If g is an element of such a kernel, then

$$\sum_{i=1}^{n} \lambda_i \gamma(a_i) = 0, \qquad (40)$$

where γ is a bidimensional polynomial of the first degree. Since the plate deformations a_i have coordinates (x_i, y_i), then equation (40) can be expressed as

$$\sum \lambda_i = 0; \quad \sum \lambda_i x_i = 0; \quad \sum \lambda_i y_i = 0. \qquad (41)$$

Equations (39) and (41) determine the deformation function or

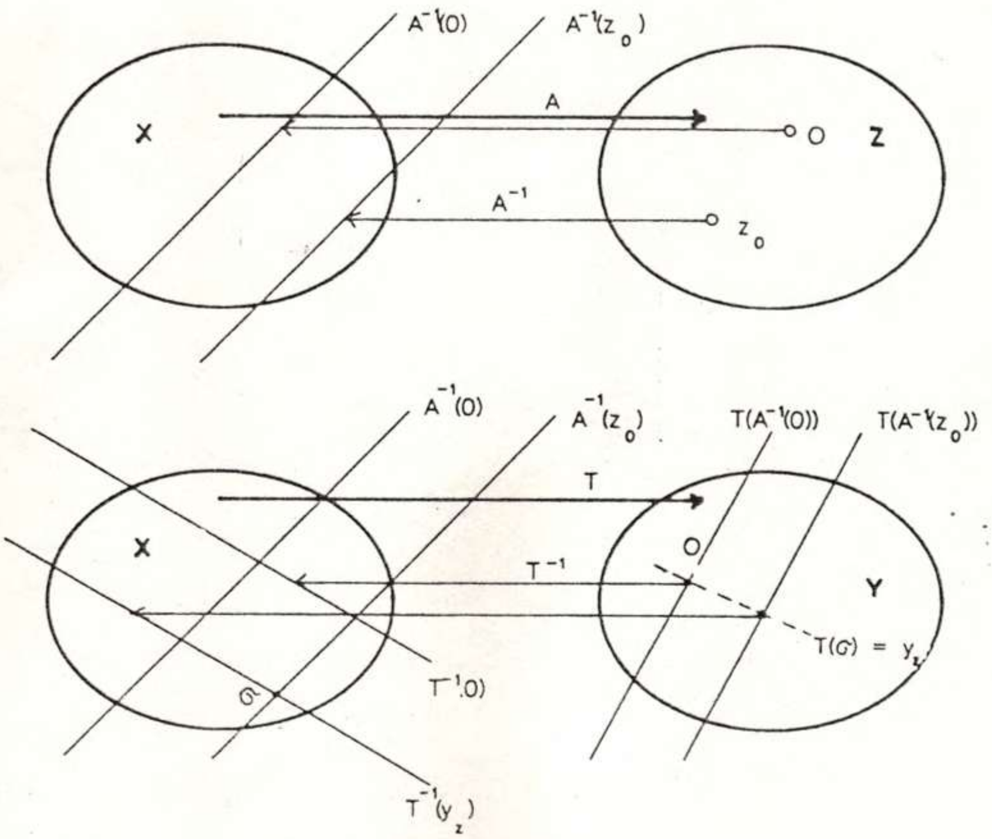


FIG. 3. Transformation A between spaces X and Z is defined in order to impose boundary conditions on the deformed, interpolating surface. Transformation T is defined between spaces X and Y in order to minimize the norm in Y-space (i.e., the curvature). z_0 is the set of observational (i.e., field) values. $A^{-1}(0)$ is the kernel of the A transformation and $A^{-1}(z_0)$ is the linear variety representing all the functions arising from z_0. y_z is the element of the Y-space that belongs to the linear variety $T(A^{-1}(z_0))$ and is closest to the zero of such a space. $T^{-1}(0)$ is the kernel of the T-transformation and $T^{-1}(y_z)$ is the linear variety that arises from y_z. σ is the required solution.

spline. Equation (39) is known as the biharmonic equation; to find its solution, we consider the function (Aronszajn, 1950)

$$H(\mathbf{t}) = \frac{1}{2\pi} |\mathbf{t}|^2 \log |\mathbf{t}|^2 \tag{42}$$

with $\mathbf{t} = (x, y)$. Then

$$\Delta^2 H = \delta$$

which means that $H(\mathbf{t})$ is a solution of the biharmonic equation. We now consider the function

$$u = \sum \lambda_i \delta_{u_i} * H, \tag{43}$$

where $*$ means convolution; then

$$\Delta^2 u = \sum \lambda_i \delta_{u_i} * \Delta^2 H = \sum \lambda_i \delta_{u_i} * \delta = \sum \lambda_i \delta_{u_i}.$$

We have shown that the function u is a solution of the biharmonic equation. The general solution is the sum of u plus a polynomial of first degree, which is the kernel of the biharmonic operator.

Writing these results in matrix form, the interpolating function $\sigma(t)$ that minimizes the functional (32) is given by (Duchon, 1975)

$$\sigma(\mathbf{t}) = \sum_{i=1} \lambda_i K(\mathbf{t} - \mathbf{t}^i) + \alpha_1 x + \alpha_2 y + \alpha_3, \tag{44}$$

with

$$\mathbf{t} = (x, y); \quad K(\mathbf{t}) = |\mathbf{t}|^2 \log |\mathbf{t}|^2; \quad |\mathbf{t}|^2 = t_x^2 + t_y^2.$$

The coefficients $\lambda_1, \ldots, \lambda_n, \alpha_1, \alpha_2, \alpha_3$ satisfy the equations,

$$\underset{\sim}{K}\Lambda + \underset{\sim}{E}\alpha = \mathbf{z},$$

and (45)

$$\underset{\sim}{E}^T \Lambda = 0,$$

with

$$\underset{\sim}{E} = \begin{vmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ \vdots & \vdots & \vdots \\ 1 & x_n & y_n \end{vmatrix};$$

$$\underset{\sim}{K} = (k_{ij}); \quad k_{ij} = k(t^i - t^j), \quad i \neq j; \quad k_{ii} = 0;$$

and

$$\Lambda = \begin{vmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_n \end{vmatrix} \qquad \alpha = \begin{vmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{vmatrix}.$$

Method of projections.—The matrix system of equations (45) is simplified by the method of projections (Paihua and Utrera, 1976). Consider the vectorial subspace of $\mathbb{R}^n$ defined as

$$E = \{(v_1, v_2, \ldots, v_n)^T \varepsilon \mathbb{R}^n \mid v_i = a + bx_i + cy_i \; i = 1, \ldots, n\}.$$

Each vector in the subspace has components formed by first-degree polynomials generated by the column vectors of matrix $\underset{\sim}{E}$. Since the vectors

$$\mathbf{u}_1 = (1, 1, \ldots, 1)^T,$$

$$\mathbf{u}_2 = (x_1, x_2, \ldots, x_n)^T, \tag{46}$$

and

$$\mathbf{u}_3 = (y_1, y_2, \ldots, y_n)^T$$

are linearly independent and generate the space E, they are then a basis of such a space. We generate an orthogonal basis by the Gram-Schmidt procedure (e.g., Nering, 1963). Let

$$\underset{\sim}{V} = (\hat{\mathbf{v}}_1, \hat{\mathbf{v}}_2, \hat{\mathbf{v}}_3)$$

be the components of the basis with

$$\hat{\mathbf{v}}_1 = \mathbf{v}_1/\sqrt{n}, \quad \hat{\mathbf{v}}_2 = \mathbf{v}_2/|\mathbf{v}_2|, \quad \hat{\mathbf{v}}_3 = \mathbf{v}_3/|\mathbf{v}_3|,$$

and with

$$\mathbf{v}_1 = \mathbf{u}_1,$$

$$\mathbf{v}_2 = \mathbf{u}_2 - \langle \mathbf{u}_2, \mathbf{v}_1 \rangle \mathbf{v}_1, \tag{47}$$

and

$$\mathbf{v}_3 = \mathbf{u}_3 - \langle \mathbf{u}_3, \mathbf{v}_1 \rangle \mathbf{v}_1 - \langle \mathbf{u}_3, \mathbf{v}_2 \rangle \mathbf{v}_2.$$

$\underset{\sim}{E}$ appears in the two equations numbered (45). Furthermore,

$$\underset{\sim}{E}^t \Lambda = 0 \tag{48}$$

implies that Λ is in the subspace orthogonal to $\mathbb{E}$ (i.e., $\mathbb{F} \equiv \mathbb{E}^\perp$); thus projections can be used to simplify equations (45).

Since the column vectors of $\underset{\sim}{V}$ are orthogonal, then the orthogonal projection of $\mathbb{R}^n$ onto $\mathbb{E}$ is given by

$$\underset{\sim}{Q} = \underset{\sim}{V}\underset{\sim}{V}^T,$$

while the projection of $\mathbb{R}^n$ onto the orthogonal subspace $\mathbb{F}$ is given by

$$\underset{\sim}{P} = \underset{\sim}{I} - \underset{\sim}{Q}.$$

Equations (45) can be projected by applying $\underset{\sim}{P}$

$$\underset{\sim}{P}\underset{\sim}{K}\Lambda + \underset{\sim}{P}\underset{\sim}{E}\alpha = \underset{\sim}{P}\mathbf{z} \tag{49}$$

since, according to equation (48), Λ is in $\mathbb{F}$. Then the projection of Λ onto $\mathbb{F}$ must be Λ, or

$$\underset{\sim}{P}\Lambda = \Lambda.$$

Thus,

$$\underset{\sim}{P}\underset{\sim}{K}\underset{\sim}{P}\Lambda + \underset{\sim}{P}\underset{\sim}{E}\alpha = \underset{\sim}{P}z.$$

However, the projection of the vectors of $\underset{\sim}{E}$ onto $\mathbb{F}$ must also be zero, i.e.,

$$\underset{\sim}{P}\underset{\sim}{E} = 0.$$

Thus

$$\underset{\sim}{P}\underset{\sim}{K}\underset{\sim}{P}\Lambda = \underset{\sim}{P}z.$$

The system of equations (45) has been reduced to

$$\underset{\sim}{A}\Lambda = \mathbf{b},$$

and (50)

$$\underset{\sim}{P}\Lambda = \Lambda.$$

where $\underset{\sim}{A} = \underset{\sim}{P}\underset{\sim}{K}\underset{\sim}{P}$ and $\mathbf{b} = \underset{\sim}{P}z$.

The problem now consists of finding a Λ^* that is a solution of equation (50), and then projecting it onto $\mathbb{F}$. From the computational point of view there is a result that further simplifies the solution. The matrix $\underset{\sim}{A}$ is positive definite (Paihua and Utrera, 1976), and therefore we can use Cholesky's algorithm to solve the system (50). The solution of the coefficients Λ of the interpolating function σ is now accomplished for the thin-plate approach. Next we need to solve for the coefficients α.

Starting from

$$\underset{\sim}{E}\alpha = z - \underset{\sim}{K}\Lambda$$

and using the equivalent matrix of $\underset{\sim}{E}$ with column vectors orthonormalized by the Gram-Schmidt method, the above equation can be written

$$\underset{\sim}{V}\hat{\alpha} = z - \underset{\sim}{K}\Lambda.$$

Thus,

$$\hat{\alpha} = -\underset{\sim}{V}^T[\underset{\sim}{K}\Lambda - z],$$

and using the orthonormal basis $[\hat{v}_1, \hat{v}_2, \hat{v}_3]$, we have

$$\alpha_1 = n\mathbf{a}_2/\langle\hat{v}_2, \hat{v}_2\rangle - \langle\hat{v}_2, \hat{v}_3\rangle \cdot \mathbf{a}_3,$$
$$\alpha_2 = n\mathbf{a}_3/\langle\hat{v}_3, \hat{v}_3\rangle, \tag{51}$$

and

$$\alpha_3 = \mathbf{a}_1 - \frac{1}{n}\langle \mathbf{u}_2, \mathbf{u}_2\rangle \cdot \mathbf{a}_2 - \langle\hat{v}_2, \hat{v}_3\rangle \cdot \mathbf{a}_3.$$

Thus equations (50) and (51) constitute the complete solution of the coefficients of the functionals in equation (32). We now summarize the computational steps necessary to find the interpolating function σ.

(1) Calculate the matrix $\underset{\sim}{K}$.

(2) Obtain $\hat{v}_1, \hat{v}_2, \hat{v}_3$, and v_1, v_2, v_3 by the Gram-Schmidt method.

(3) Calculate the matrix $\underset{\sim}{A} = \underset{\sim}{P}\underset{\sim}{K}\underset{\sim}{P}$ by Cholesky factorization.

(4) Project z onto $\mathbb{F}$ to obtain $\mathbf{b} = \underset{\sim}{P}\mathbf{z}$.

(5) Find Λ by solving the system $\underset{\sim}{A}\Lambda = \mathbf{b}$.

(6) Project Λ onto $\mathbb{F}$: $\Lambda^* = \underset{\sim}{P}\Lambda$.

(7) Compute $\alpha_1, \alpha_2, \alpha_3$ by means of equation (51).

Since the total number of points can be subdivided (Paihua and Utrera, 1976) to obtain the solution of the system in step (5), the total number of points can be made arbitrarily large.

Finally, we stress that the kernel (42) of the biharmonic equation (39) is just one of the possible solutions for such an equation. Another possibility is, for instance, the pseudocubic splines where

$$K(t) = |t|^{2/3}.$$

However, Campos et al. (1983) stated that pseudocubic splines yield poorer results than the thin-plate approach, especially in places which lack measurements in a given region.

The finite-difference approach

This method belongs to the numerical surfaces classification and also deals with curvature minimization under the thin-plate approach; however, curvature here is approximated (Briggs, 1974) by

$$C = \sum_{i=1}^{I}\sum_{j=1}^{J}(C_{ij})^2, \tag{52}$$

where C_{ij} is the curvature in (x_i, y_i). Let $u_{ij} = u(x_i, y_i)$ be the displacement of the plate in (x_i, y_i). C_{ij} is then a function of u_{ij} and nearby grid values; the total curvature depends upon the precision with which the curvature is represented in u_{ij}. In order to minimize equation (52), the partial derivatives of C with respect to u_{ij} must equal zero (Stiefel, 1963).

$$\frac{\partial C}{\partial u_{ij}} = 0 \qquad \begin{array}{l} i = 1, \ldots, I \\ j = 1, \ldots, J. \end{array} \tag{53}$$

This results in a set of relations between the neighboring grid points, with one relation per point.

Two general curvature descriptions are needed: (1) for points on the regular grid, with spacing h which also have expressions for the corners, edges, and intermediate points. In the latter case, for instance, we have

$$C_{ij} = (u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - 4u_{ij})/h^2. \tag{54}$$

(2) For points not on the grid,

$$C_{ij} = \sum_{k=1}^{4} b_k u_k - u_{ij}\sum_{k=1}^{4} b_k + b_5 w_n. \tag{55}$$

The coefficients b_k, $k = 1, \ldots, 5$ are calculated from the Taylor series expansion of the points,

$$u_k\{x_0 + \xi_k, y_0 + \eta_k\}, \qquad k = 1, \ldots, 5$$

and by taking five u_k values of equation (55) as

$$(h, -h), \quad (0, -h), \quad (-h, 0), \quad (-h, h), \quad \text{and} \quad (\xi_5, \eta_5),$$

where w_n is the nearby observation value.

Equation (55) is the expression for points outside the grid. Substituting equations (54) and (55) into equation (53), we obtain a system of linear equations that allows calculation of interpolated points.

The accuracy of this method depends upon the precision with which the curvature is approximated, particularly for points outside the grid. If the grid spacing is large, or if there are few points in a given region, the Taylor series expansion is less approximate. The method may show convergence problems when the number of points is large or when the grid spacing h is also large. Even without a direct comparison based on modeled results between the finite-difference method and the analytic solution, it is not difficult to see that the results of the former method tend to those of the latter as $h \to 0$. The field values are not as accurately reproduced by the finite-difference method as they are by the analytic solution; however, the former is less costly and faster.

NUMERICAL EVALUATIONS

A quantitative comparison is possible between the various interpolation techniques following the procedure established in Rasmussen and Sharma (1979). They used geomagnetic models to compute actual field values in specified positions; interpolations at such points with parametric and nonparametric methods allow numerical comparisons between them. They computed the corresponding mean error, maximum error, and standard deviation in percentage with respect to the largest field value of the anomaly, concluding that nonparametric methods yield figures approximately 50 percent better than parametric methods. They point out, however, that the latter take 50 percent less computer time.

Following this trend, and reproducing the model prisms used by the above authors, we carried out a numerical evaluation for the strictly bidimensional thin-plate approach. The models consist of three dikes of equal length (16 km) and widths of 4, 8, and 16 km buried at depths of 0.25, 0.50, and

1.0 km. We assumed a magnetic susceptibility of 0.003 cgs and a terrestrial field of 50 000 nT, with inclination 75 degrees and declination of 0 degrees. Using a 3-D magnetic modeling program, we generated 40 × 40 field values on the surface at 1 km spacing; from these we obtained a subset of values spaced on a regular grid at 2 km separation. The latter were used as input for the thin-plate interpolation, with an interpolating interval of 1 km. The matrix of exact data was subtracted from the matrix of interpolated data and the error matrix (Table 1 was obtained). Table 2 shows the error evaluation reported in Rasmussen and Sharma (1979) for the non-parametric method (i.e., for the most exact quasi-bidimensional algorithm). From these tables note that the maximum difference for the maximum error is 2.7 percent in favor of the quasi-bidimensional method, while the standard deviation and the mean errors are practically equal. This preliminary comparison shows that on regular grids the methods yield similar results.

Tables 3, 4, and 5 were computed with the thin-plate program for random data distributions, which are the data sets pertinent to this discussion. For 300 random "field" data, we generated the regular grid of 20 × 40 nodes (1 km apart) and obtained Table 3.

Similarly, we obtained Table 4 from 600 random field data points, and Table 5 from 900 random field data points, all generated with the above computer program. Note that accuracy strongly depends upon the number of initial random data points with which the interpolating surface is generated. As an example, for the prism of 4 km width buried at a depth of 0.5 km, the mean error is 0.67 percent for 300 random data points, 0.37 percent for 600 random data points, and 0.24 percent for 900 random data points; the corresponding percentages for the maximum error are 26.7, 20.6, and 11.6. Figures 4a and 4b show, respectively, plots of the percent maximum error and the percent standard deviation versus the field gradient in nanoteslas per kilometer for the dike of 4 km width buried at a depth of 1 km. It is shown that the error increases when the gradient becomes more pronounced (i.e., when the prism is shallower) and decreases when the number of input data points increases. The maximum error in the 900

Table 1. Errors between field values and interpolated data from thin plate approach (800 points on regular grid).

Depth	Width	Mean Error	%	Max. Error	%	S.D.	%	Max. Value	Grad.
1.0	4	1.47	0.25	47.36	8.06	4.60	0.78	587	240
1.0	8	1.74	0.25	48.44	7.01	5.01	0.72	690	265
1.0	16	2.17	0.29	55.06	7.56	5.45	0.74	727	274
0.5	4	3.13	0.40	106.35	13.30	10.30	1.32	778	459
0.5	8	7.78	0.44	108.45	12.72	11.27	1.32	852	479
0.5	16	4.70	0.53	111.99	12.78	12.02	1.37	876	476
0.25	4	4.50	0.50	160.50	18.02	15.08	1.69	890	759
0.25	8	5.52	0.58	165.13	17.48	16.51	1.74	944	778
0.25	16	6.94	0.72	167.25	17.41	17.66	1.83	960	793

Table 2. Errors between field values and interpolated data from Rasmussen and Sharma (1979) (800 points on regular grid).

Depth	Width	Mean Error	%	Max. Error	%	S.D.	%	Max. Value	Grad.
1.0	4	0.68	0.12	25.06	4.39	2.68	0.47	571	237
1.0	8	1.12	0.16	21.24	3.03	3.43	0.49	701	259
1.0	16	2.09	0.28	20.94	2.80	4.78	0.64	748	263
0.5	4	1.39	0.18	74.28	9.56	6.75	0.87	777	458
0.5	8	2.38	0.28	71.65	8.43	9.26	1.09	850	477
0.5	16	4.53	0.51	71.74	8.07	13.06	1.47	889	473
0.25	4	2.40	0.27	141.67	15.3	12.87	1.39	926	748
0.25	8	4.67	0.47	141.29	14.2	18.00	1.81	995	758
0.25	16	8.95	0.87	169.78	16.5	62.15	6.04	1 029	761

Table 3. Errors between field values and interpolated data from thin plate approach (300 random data).

Depth	Width	Mean Error	%	Max. Error	%	S.D.	%	Max. Value	Grad.
1.0	4	2.44	0.41	87.28	14.86	5.33	0.90	587	240
1.0	8	3.39	0.49	74.16	10.74	5.80	0.85	690	265
1.0	16	5.94	0.81	119.30	16.39	9.11	1.25	727	274
0.5	4	5.25	0.67	207.65	26.66	12.19	1.56	778	459
0.5	8	7.53	0.88	173.92	20.41	14.40	1.69	852	479
0.5	16	12.79	1.46	232.30	26.51	20.70	2.36	876	476
0.25	4	8.17	0.91	339.40	33.12	20.05	2.25	890	759
0.25	8	12.09	1.28	269.60	28.45	24.56	2.60	944	778
0.25	16	20.24	2.10	352.92	36.73	33.94	3.53	960	793

Table 4. Errors between field values and interpolated data from thin plate approach (600 random data).

Depth	Width	Mean Error	%	Max. Error	%	S.D.	%	Max. Value	Grad.
1.0	4	1.19	0.20	67.97	11.57	3.28	0.55	587	240
1.0	8	1.51	0.21	42.14	6.10	3.26	0.4	690	265
1.0	16	2.4	0.32	42.0	5.77	3.64	0.50	727	274
0.5	4	2.90	0.37	160.08	20.55	8.52	1.09	778	459
0.5	8	3.80	0.45	110.9	13.02	9.18	1.07	852	479
0.5	16	5.71	0.65	107.4	12.26	10.08	1.15	876	476
0.25	4	5.04	0.56	247.19	27.7	15.92	1.7	890	749
0.25	8	7.01	0.74	220.13	23.31	17.9	1.90	944	778
0.25	16	10.87	1.13	215.9	22.4	20.6	2.15	960	793

Table 5. Errors between field values and interpolated data from thin plate approach (900 random data).

Depth	Width	Mean Error	%	Max. Error	%	S.D.	%	Max. Value	Grad.
1.0	4	0.71	0.12	42.25	7.19	2.05	0.35	587	240
1.0	8	0.86	0.12	21.69	3.14	1.85	0.26	690	265
1.0	16	1.32	0.18	20.96	2.88	2.03	0.27	727	274
0.5	4	1.87	0.24	90.50	11.62	5.96	0.76	778	459
0.5	8	2.13	0.25	72.10	8.46	5.36	0.63	852	479
0.5	16	3.82	0.43	72.19	8.24	6.88	0.78	876	476
0.25	4	3.86	0.43	187.06	21.00	12.87	1.44	890	749
0.25	8	2.13	0.25	72.10	8.46	5.36	0.63	852	778
0.25	16	8.41	0.87	17.95	18.6	16.82	1.75	960	793

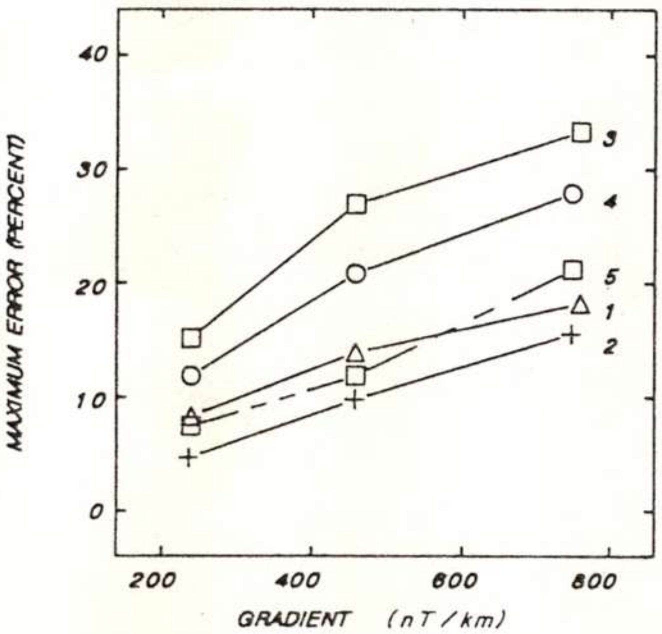


FIG. 4. (a). Percent maximum error versus magnetic field gradient for a dike 16 km long, 4 km wide, 30 km high, and buried at a depth of 1 km. The error increases when the gradient becomes more pronounced and decreases when the number of input data increases.

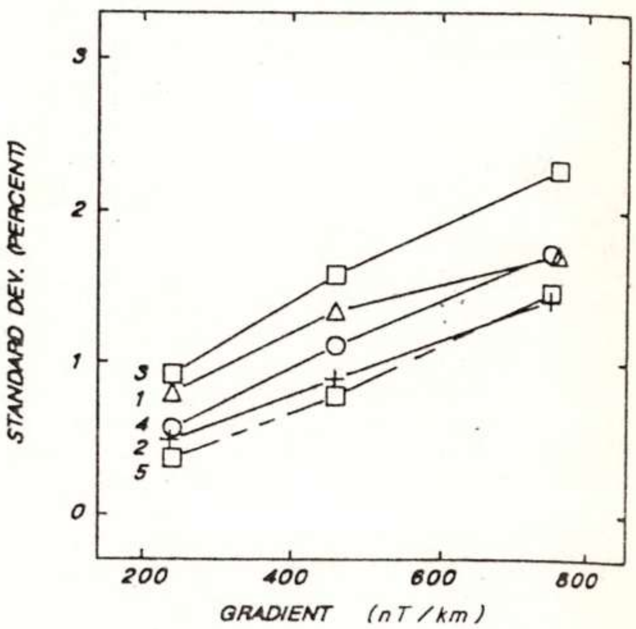


FIG. 4 (b). Standard deviation versus magnetic field gradient for the same prism as above.

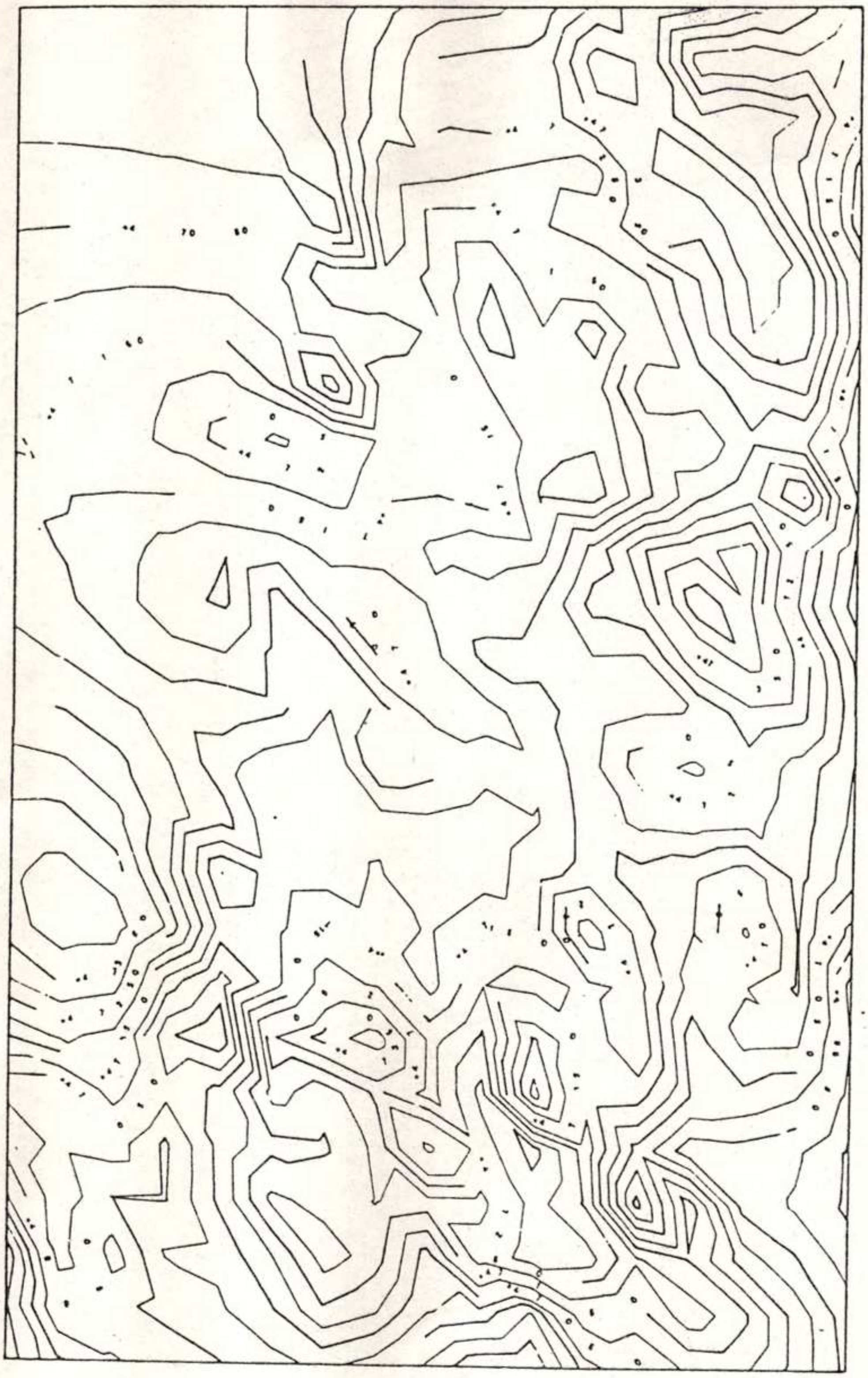


FIG. 6. Interpolation of the data set in Figure 5 on a regular grid of 17 × 26 with the thin plate approach, plus linear interpolations between the regular grid points for final contouring yield a rough-looking map.

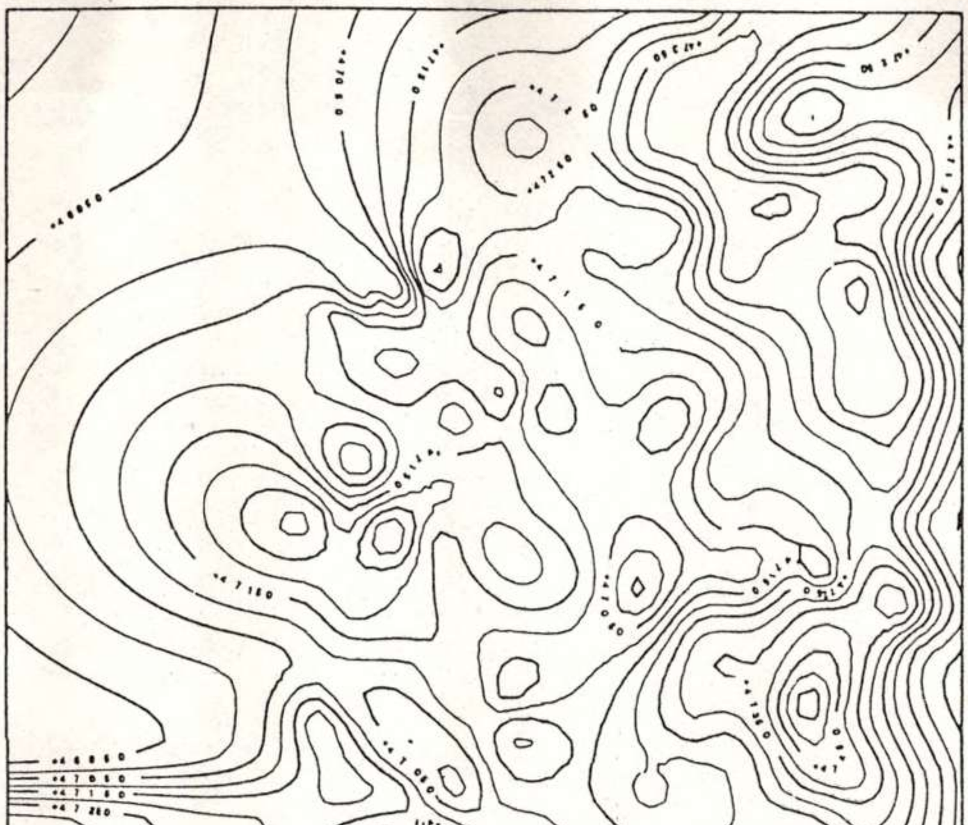


Fig. 7. Interpolation of the data set in Figure 5 on a regular grid of 65×103 points with the thin plate approach, plus linear interpolations between the regular grid points for final contouring yields a smoother aspect to the map. The same set of coefficients as those in Figure 6 were used for the analytical computations.

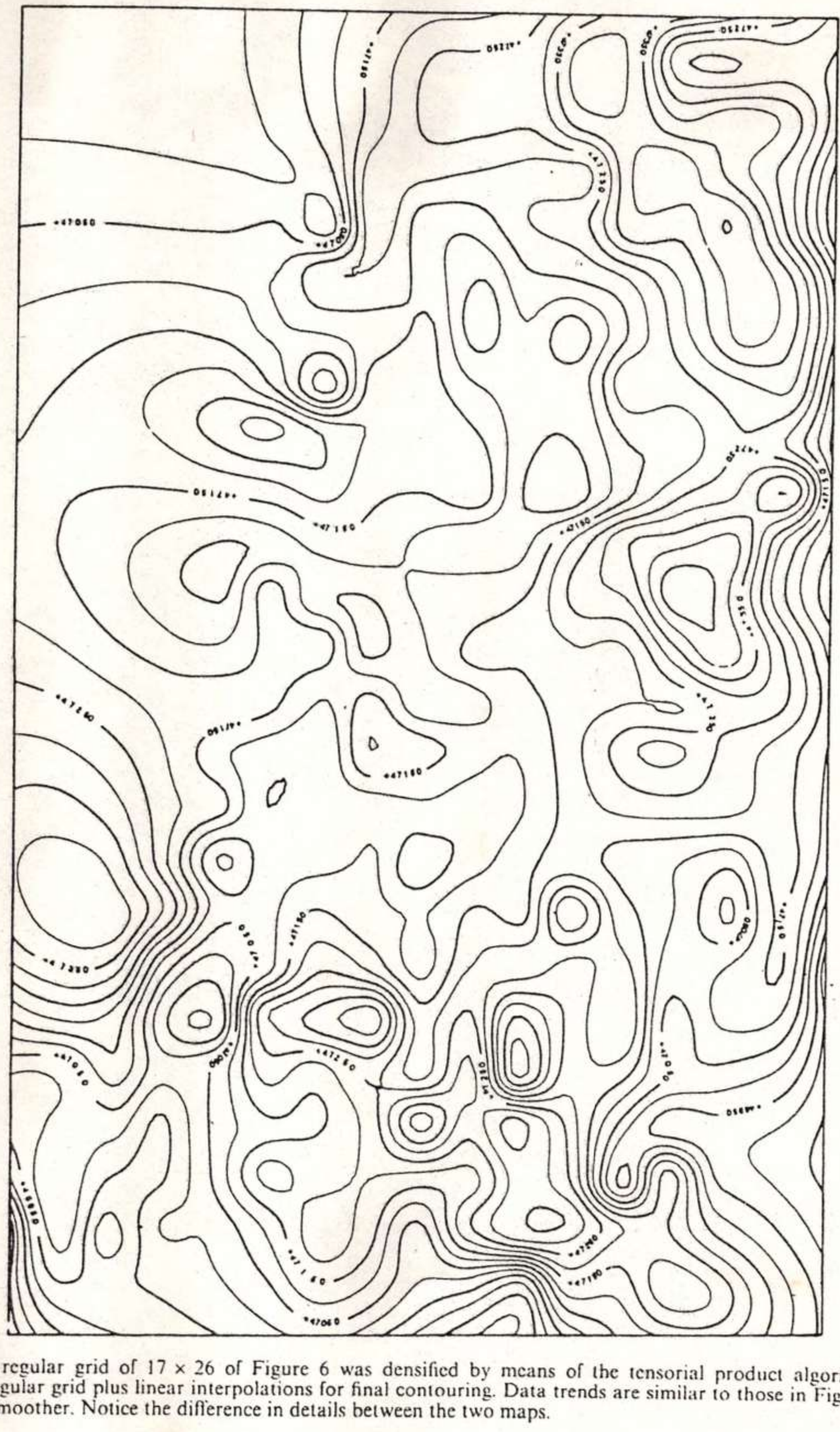


FIG. 8. The regular grid of 17 × 26 of Figure 6 was densified by means of the tensorial product algorithm to a 113 × 276 regular grid plus linear interpolations for final contouring. Data trends are similar to those in Figure 7, but this map is smoother. Notice the difference in details between the two maps.

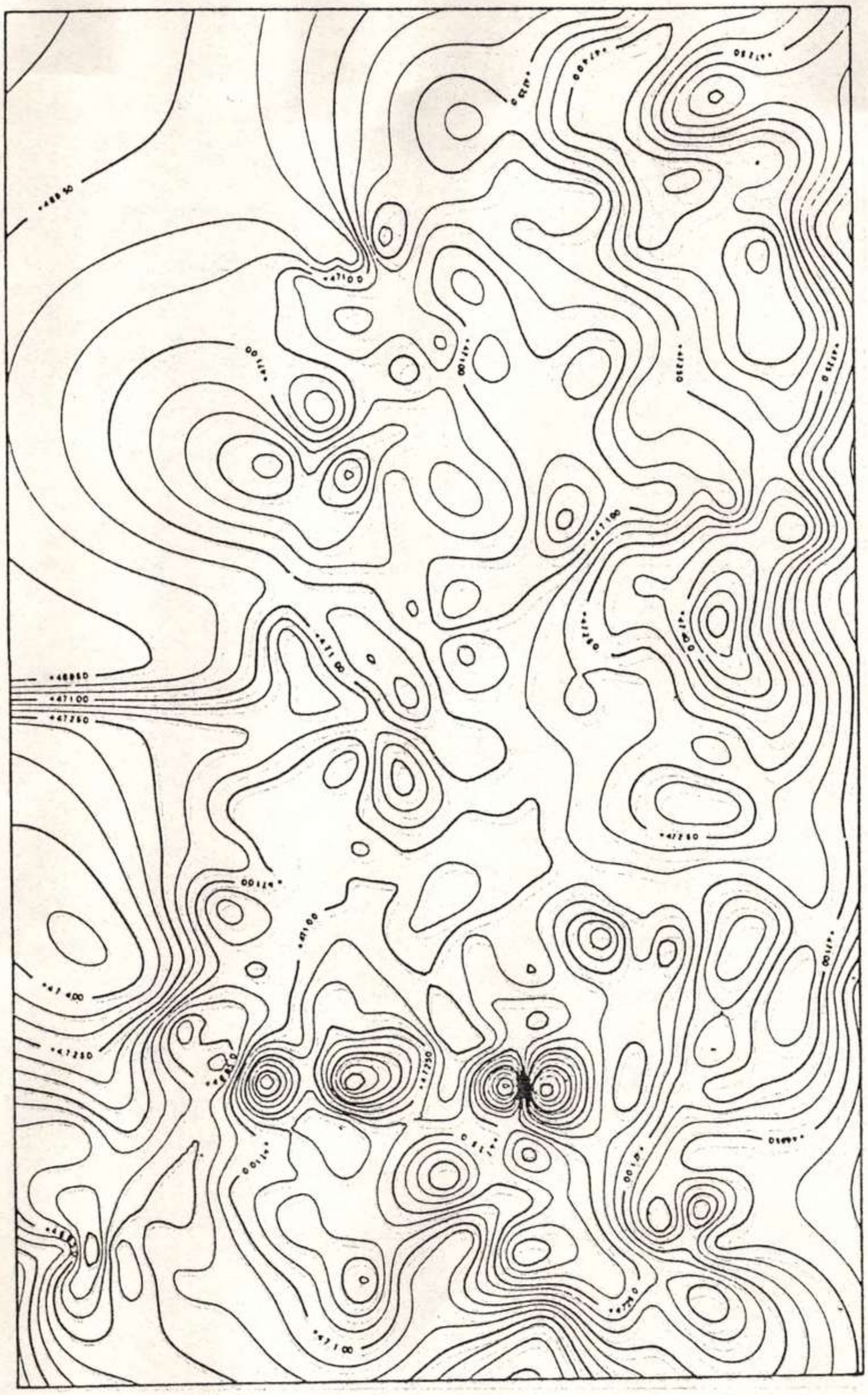


FIG. 9. The regular data grid of 65 × 103 points in Figure 7 is densified to a regular grid of 129 × 205 points by means of the tensorial product algorithm. This map is approximately 15 times more expensive than the map in Figure 8; notice, however, the enhancement of small anomalies.

lytical solution are the most accurate methods of interpolating random data. However, they are also the most expensive methods.

A quantitative analysis of interpolation errors suggests that a combination of methods may work satisfactorily and economically. Starting from a set of random data a regular grid is produced with a strictly bidimensional algorithm; the data density of the interpolated data is approximately the same value as that of the random data. The regular grid can next be used as input of a quasi-bidimensional algorithm in order to further densify the data. Actual contouring is performed last by means of a fast, linear algorithm operating on a dense grid. An actual example involving 327 randomly distributed points within an irregular area and bounded by a rectangle is used to illustrate some of the problems involved in automatic contouring of geophysical data sets.

ACKNOWLEDGMENTS

We acknowledge the enthusiastic help of Angeles Figueroa in preparing the typed version of this manuscript. The referees' suggestions greatly improved the original text.

REFERENCES

Ahlberg, J. H., Nilson, E. N., and Walsh, J. L., 1967, The theory of splines and their applications: Academic Press, Inc.

Aronszajn, N. A., 1950, Theory of reproducing kernels: Trans. Amer. Math. Soc., **68**, 337–404.

Bhattacharyya, B. K., 1969, Bicubic spline interpolation as a method for treatment of potential field data: Geophysics, **34**, 402–423.

Briggs, J. C., 1974, Machine contouring using minimum curvature: Geophysics, **39**, 39–48.

Campos Enriquez, J. O., Thomann, J., and Goupillot, M., 1983, Applications of bidimensional spline functions to geophysics: Geophysics, **48**, 1269–1273.

Crain, I. K., 1970, Computer interpolation and contouring of two-dimensional data: A review: Geoexplor., **8**, 71–86.

Crain, I. K., and Bhattacharyya, B. K., 1967, Treatment of non-equispaced two-dimensional data with a digital computer: Geoexplor., **5**, 173–194.

De Boor, C., 1962, Bicubic spline interpolation: J. Math. and Phys.: **41**, 212–218.

Dooley, J. C., 1976, Two-dimensional interpolation of irregularly spaced data using polynomial splines: Phys. Earth and Plan. Int., **12**, 180–187.

Duchon, J., 1975, Fonctions spline du type plaque mince en dimension 2: Sèminaire d'analyse numérique, n. 231, Université Scientifique et Médicale de Grenoble, France.

Ferguson, J., 1964, Multivariate curve interpolation: J. Assn. Comp. Mach., **11**, 221–228.

Gel'fand, I. M. and G. E. Shilov, 1964, Generalized functions: Vol. 1. Properties and operations: Academic Press, Inc.

Heissing, R. C., Lee, H. K., Price, A., and Powers, E. N., 1972, Automatic contouring using bicubic functions: Geophysics, **37**, 669–674.

Landau, L. D. and Lifschitz, E. M., 1970, Theory of elasticity: Pergamon Press, Inc.

Marcus, M., 1960, Basic theorems in matrix theory: Nat. Bur. Stand. Appl. Math. Series 57.

Nering, E. D., 1963, Linear algebra and matrix theory: John Wiley and Sons, Inc.

Paihua, L. and Utrera Diaz, F., 1976, Un ensamble de programmes pour l'interpolation des fonctions spline du type plaque mince: Séminaire d'analyse numérique n. 273, Université Scientifique Médicale de Grenoble, France.

Rasmussen, K. L. and Sharma, P. V., 1979, Bicubic spline interpolation: A quantitative test of accuracy and efficiency: Geophys. Prosp., **27**, 394–408.

Reinsch, C. H., 1967, Smoothing by spline functions: Numer. Math., **10**, 177–183.

Stiefel, E. L., 1963, An introduction to numerical mathematics: Academic Press, Inc.

Thomson, R. F., 1970, Spline interpolation on a digital computer: Preprint X-692-70-261, Goddard Space Flight Center, Greenbelt, Maryland.

Wahba, G., and Wendelberger, J., 1980, Some new mathematical methods for variational objective analysis using splines and cross-validation: Monthly Weather Rev., **108**, 1122–1143.