Automatic Contouring from Scattered Data Points

I. P. Schagen
Department of Computer Studies, University of Technology, Loughborough, Leicester LE11 3TU, UK

It is possible to interpolate in two dimensions from scattered data points using a stochastic process model, giving an interpolating function which is continuous in all derivatives, passes exactly through the points given and does not generate spurious features in regions of no data. Using this interpolating method, contours are produced directly from the data without an intermediate grid. Extensions of the basic model include a two-stage model which allows for a long-range trend.

INTRODUCTION

Relatively efficient methods exist for the automatic contouring of functions defined mathematically, or specified at the nodes of a grid system. However, when the surface to be contoured is specified only at a finite number of points arbitrarily positioned in two dimensions, we run into both philosophical and practical difficulties. The philosophical difficulty is that there are an infinite number of contour maps which will fit the given data points, and they can differ widely. Most existing schemes for interpolating the data to undefined points in the region of interest run into practical difficulties—either they fail to give the correct values at the known data points or they lead to interpolating functions which are not continuous in all derivatives. In addition, existing methods for scattered data points involve interpolating from the data points to the nodes of a grid, and then using an existing method for grid nodes to produce the contour map. This method can lead to loss of information in the final map unless the grid used is extremely fine.

The method described here uses a statistically-based interpolation technique, described in another paper,[1] which not only gives correct values at the data points but leads to an interpolating function which is continuous in all its derivatives. The contour map is produced directly from the given data points without an intermediate grid being necessary. This paper describes the practical aspects of generating such a contour map.

THE INTERPOLATING FUNCTION

Given n data points, values $z_1, z_2, \ldots, z_n$, at positions $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ and two parameters μ and ρ, the interpolating function at an unknown point is given by

$$f(x, y) = \gamma' \cdot \mathbf{c}(x, y) + \mu \tag{1}$$

where γ is a vector of n values obtained from the assumed correlation structure of the data (given by the parameter ρ) and the values at the n known points, and $\mathbf{c}(x, y)$ is a vector of correlation values between the n known data points and the unknown point (x, y). These correlations are specified by the distances between (x, y) and each known point (x_i, y_i) and by the parameter ρ.

Thus the values γ may be calculated once for all. This step involves inverting the $n \times n$ correlation matrix for the known points, and if n is large it may be more efficient to partition the points into groups, and calculate the γ-values separately for each group, taking into account neighbouring points. The values of γ may be considered to be 'uncorrelated' data values, wherein the fact that the data values are spatially correlated has been removed from the data.

Thus each interpolating function evaluation requires the calculation of n correlation values, and a multiplication with a constant vector. With little additional work, it is possible to calculate the first and second derivatives of the interpolating function, and these are used in the contour tracing. Details of the theory of this interpolating method are to be found in the other paper.[1]

The two parameters, μ and ρ, may be estimated from the known data points, as described in the earlier paper, or can be chosen to suit the user of the system. The 'correlation distance' ρ may be considered to be the distance over which spatial correlation between two points is appreciable. The 'grand mean' μ is the limiting value to which the interpolating function tends far away from any known data points.

CONTOUR TRACING

Most algorithms for contouring from scattered data points require that an intermediate system of grid nodes be produced, and the contouring is done from this.[2] Unless the data points all manage to coincide with grid nodes, it is possible for the contours so drawn to be inconsistent with the original data. This has been seen in practice with the program GPCP (General Purpose Contouring Program) produced by Calcomp Limited.[3] In general, it is going to be possible for the grid nodes and data points to be made to coincide if we use an irregular rectangular grid, with $n \times n$ nodes. In practice this may be impossible to realize if n is large.

The sole purpose for the intermediate grid is to enable the contouring algorithm to carry out the following functions: (1) ensure that all the contour segments appropriate to the set of data are drawn; (2) define a starting point for the drawing of each such contour segment; (3) decide when to terminate a contour segment, either because the starting point has been reached again, or because the area of interest has been left.

CCC-0010-4620/82/0025-0007 $02.50

All these functions can in fact be carried out without the use of an intermediate grid. This is done as follows—as well as the known data points, a rectangular boundary is defined, within which the contours are to be drawn. At each vertex of the bounding rectangle, the value of the interpolating function is calculated, giving us four 'dummy' data points. The system for keeping track of the contours works by means of a set of 'reference points'. Such a set is defined for each contour level, and consists of a number of points where the interpolating function value exactly equals the contour level. The set is chosen so that at least one such reference point lies on each 'definable' contour segment within the area of interest.

A definable contour segment is one which divides the area of interest into two parts, each containing at least one data point. It is possible for undefinable contour segments to exist, which cannot be detected by the present algorithm, except by accident. Figure 1 shows such a segment. Such contour segments are products of the interpolating function, and do not relate directly to the given data. It is debatable whether or not they should actually be drawn at all. If they need to be drawn, it should be possible to include extra heuristics in the algorithm to discover them.

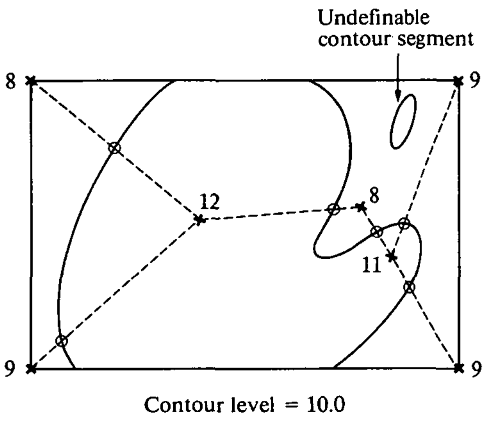

Figure 1. Example of reference points. X^9—Data point with value. ·—Reference point.

Reference points are defined by means of a set of straight lines joining data points. Each such line joins a data point with value greater than the contour level to a point with value less than the contour level. Border points are all connected to internal data points. A search is carried out along each line until a point is found with interpolating function value equal to the contour level, and this becomes a new reference point. Figure 1 illustrates this process. The number of reference points is less than or equal to $n + 3$.

Using these reference points, the algorithm for drawing all the definable contour segments appropriate to a particular contour level works as follows:

(1) The first reference point on the list is taken as the starting point of the new contour segment.
(2) From the present point on the contour, a new point is computed. This is repeated until

either (3) If the contour segment has reached its starting point again, then the tracing of this segment is ended and it is drawn.

or (4) If the new point falls outside the boundary, this arm of the segment is ended. If the other arm has also been ended, then the contour segment is drawn. Otherwise, tracing the other arm of the segment is begun from the starting point, in the opposite direction.

(5) After a contour segment has been drawn, all the reference points which lie on that segment are deleted from the list.
(6) If any reference points still exist on the list, a new segment is started from step 1.

A flowchart to illustrate the algorithm is shown in Fig. 2.

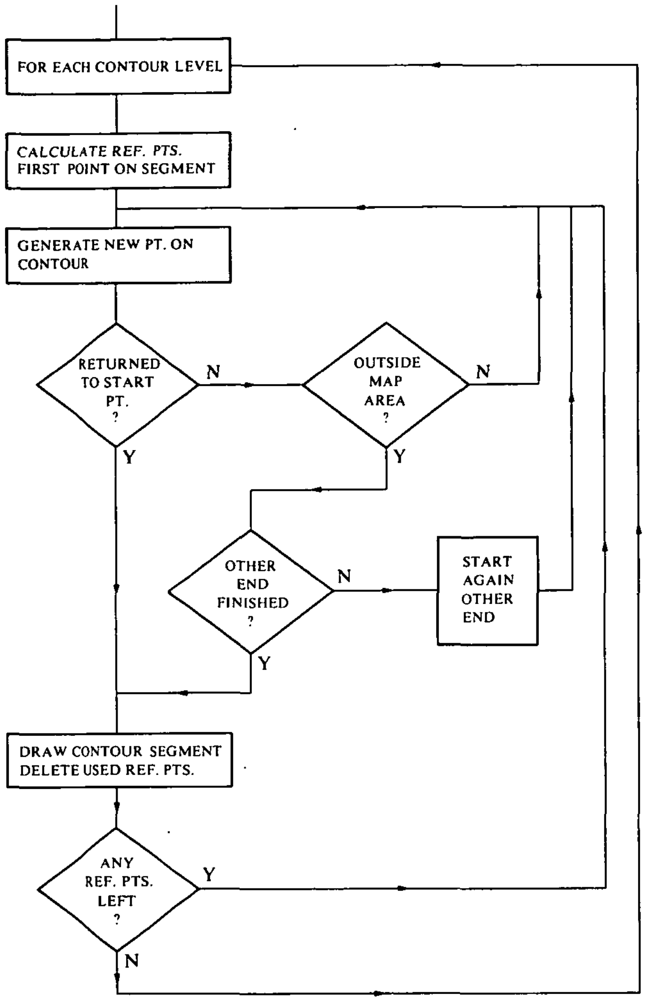

Figure 2. Outline of contouring algorithm using reference points.

DEFINING A NEW CONTOUR POINT

The algorithm for generating a new point on the contour from the previous point operates in two stages:

(1) A tangent is drawn to the contour at the current point, at an angle of θ to the horizontal, where

$$\theta = \tan^{-1}\left(\frac{-\partial f/\partial x}{\partial f/\partial y}\right) \tag{2}$$

A point a distance Δr along the tangent is chosen, where Δr is known as the 'step length' for generating the new point.

(2) From this point a perpendicular is drawn to the tangent, and a search is carried out along this line until a function value $f(x, y)$ is found which is within a specified tolerance of the desired contour level. If no such point is found within a reasonable distance (due for example to the contour forming a sharp bend in the area), then the value of Δr is halved and the process repeated.

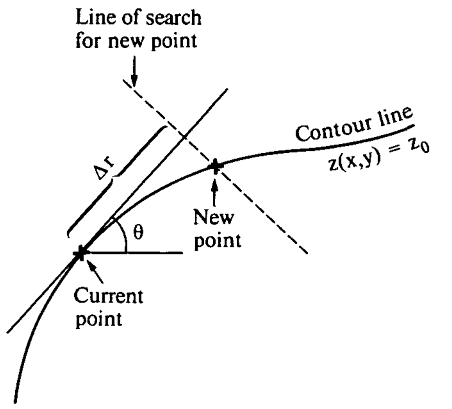

Figure 3. Defining new point on contour.

Figure 3 illustrates this procedure. An interesting refinement is to attempt to select an optimal value of Δr, the step length, so as to use the fewest points to define a reasonably smooth contour. In areas where the contour is almost straight, Δr can be large. Where the contour is sharply curved, Δr should be much smaller. Let $d\theta/dr$ be the rate of change of the contour tangent angle with distance along the contour. Then

$$\frac{d\theta}{dr} = -\beta^3\left[\left(\frac{\partial f}{\partial y}\right)^2\frac{\partial^2 f}{\partial x^2} - 2\frac{\partial f}{\partial x}\cdot\frac{\partial f}{\partial y}\cdot\frac{\partial^2 f}{\partial xy} + \left(\frac{\partial f}{\partial x}\right)^2\frac{\partial^2 f}{\partial y^2}\right] \quad (3)$$

where $\beta = \left[\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2\right]^{-1/2}$

If we specify a desired change in direction from point to point on the contour of $\Delta\theta$, then we may get a reasonable value for the step length by setting

$$\Delta r = \frac{\Delta\theta}{|d\theta/dr|} \quad (4)$$

It is obviously necessary to put reasonable maximum and minimum limits on Δr in the contouring program.

THE TWO STAGE MODEL

The main criticism which could be levelled at the simple stochastic process model used to produce the interpolating function is that it assumes the process is stationary and makes no allowance for a trend or large-scale drift of the mean. In a lot of cases this simple model gives good results, but it is obvious that in many cases something more sophisticated is required. On the other hand, the fitting of functions such as polynomials to represent the trend of the data, while common, has nothing to recommend it but computational convenience. Very few physical variables extrapolate to infinity or minus infinity as polynomials do, so that such a model of a large-scale trend is of value only over a limited area. The approach used in 'universal kriging'[4] of modelling the data with a polynomial or similar functional trend plus a correlated stochastic residual, is a rather unnatural hybrid.

The simple stochastic model can be extended quite naturally, by assuming that the trend is also a stationary stochastic process with a large correlation distance, so that it varies more slowly than the stochastic process representing the residuals. The stochastic model to be fitted to the data is therefore

$$Z(x, y) = Z_L(x, y) + Z_S(x, y) \quad (5)$$

where $Z_L(x, y)$ is a Gaussian random process over the plane area of interest, with mean μ and correlation function

$$g_L(r) = \exp[-r^2/2\rho_L^2] \quad (6)$$

and $Z_S(x, y)$ is a similar process with mean 0 and correlation function

$$g_S(r) = \exp[-r^2/2\rho_S^2] \quad (7)$$

and $\rho_L \gg \rho_S$. Figure 4 illustrates this model.

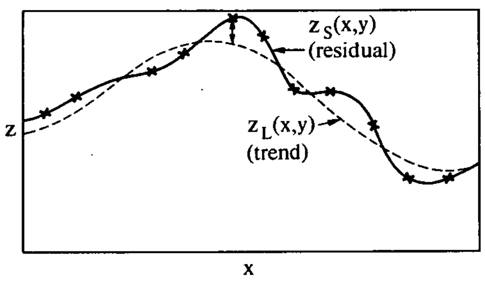

Figure 4. Illustration of two-stage model.

Three parameters (μ, ρ_L and ρ_S) are needed to fit the model. This could obviously be fitted by writing down an expression for the likelihood, and finding three values which maximized this, but such an approach would be rather inefficient computationally. Therefore a simpler, though rather *ad hoc*, approach has been developed, which will be described briefly.

First, the data points are grouped together into clusters of some specified size. This could be done in various ways: the method selected is to add a new point to a cluster only if it lies within some preselected distance, l say, of all the existing points in the cluster. This gives rise to a set of clusters which are reasonably compact. The centroid of each such cluster is found and given a value equal to the average of the points in the cluster. In this way a set of m 'average points' is found, with $m \ll n$, the original number of data points.

The parameters μ and ρ_L are fitted to these m points using the maximum likelihood methods developed in the case of the simple stochastic model. The trend part of the model has now been defined. All that remains is to estimate ρ_S, which is done using the residual errors at the n data points

$$z_i^E = z_i - \hat{z}_i \quad (8)$$

where z_i^E is the residual error, z_i is the original data value and $\hat{z}_i$ is the trend value at the point, estimated from the m average points.

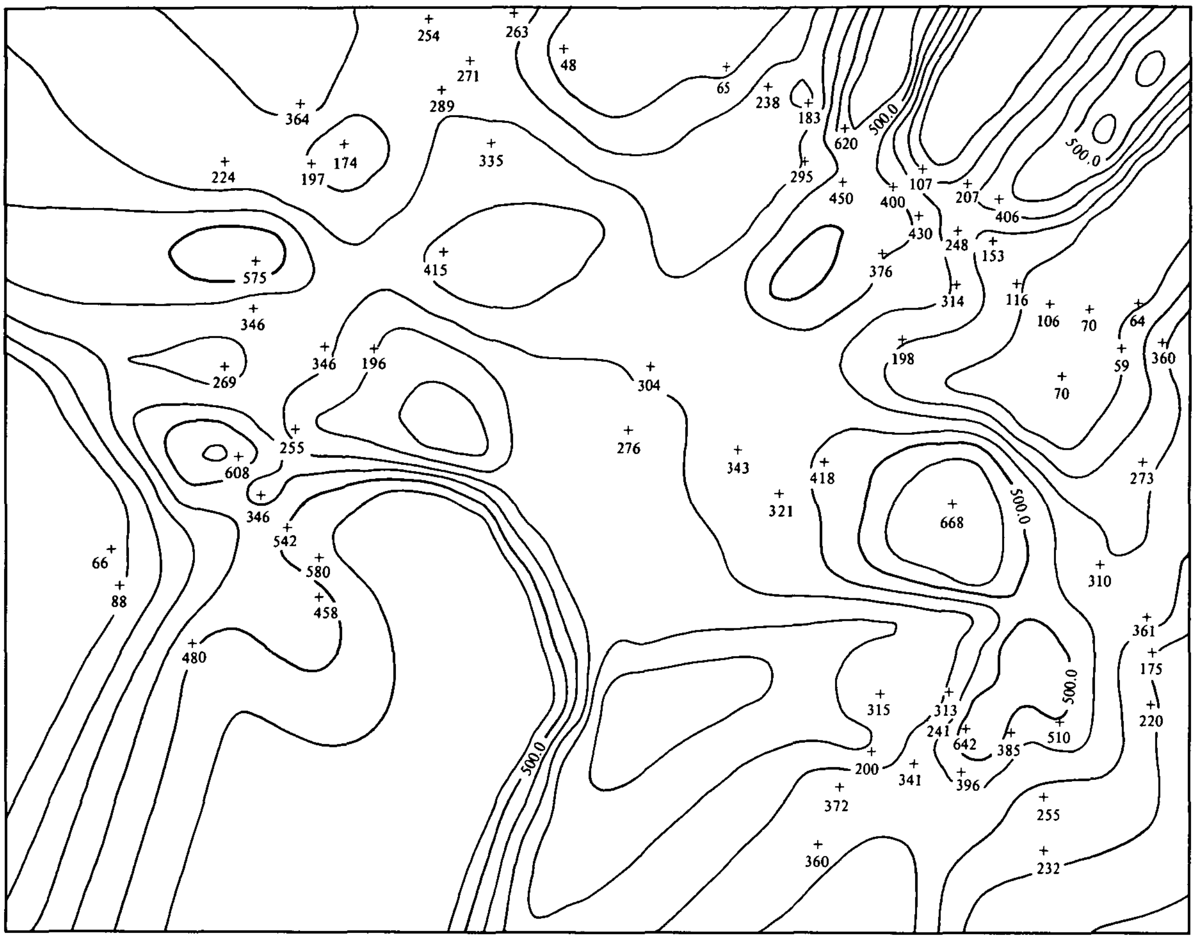

Figure 5. Shkapovskii data contoured by GPCP.

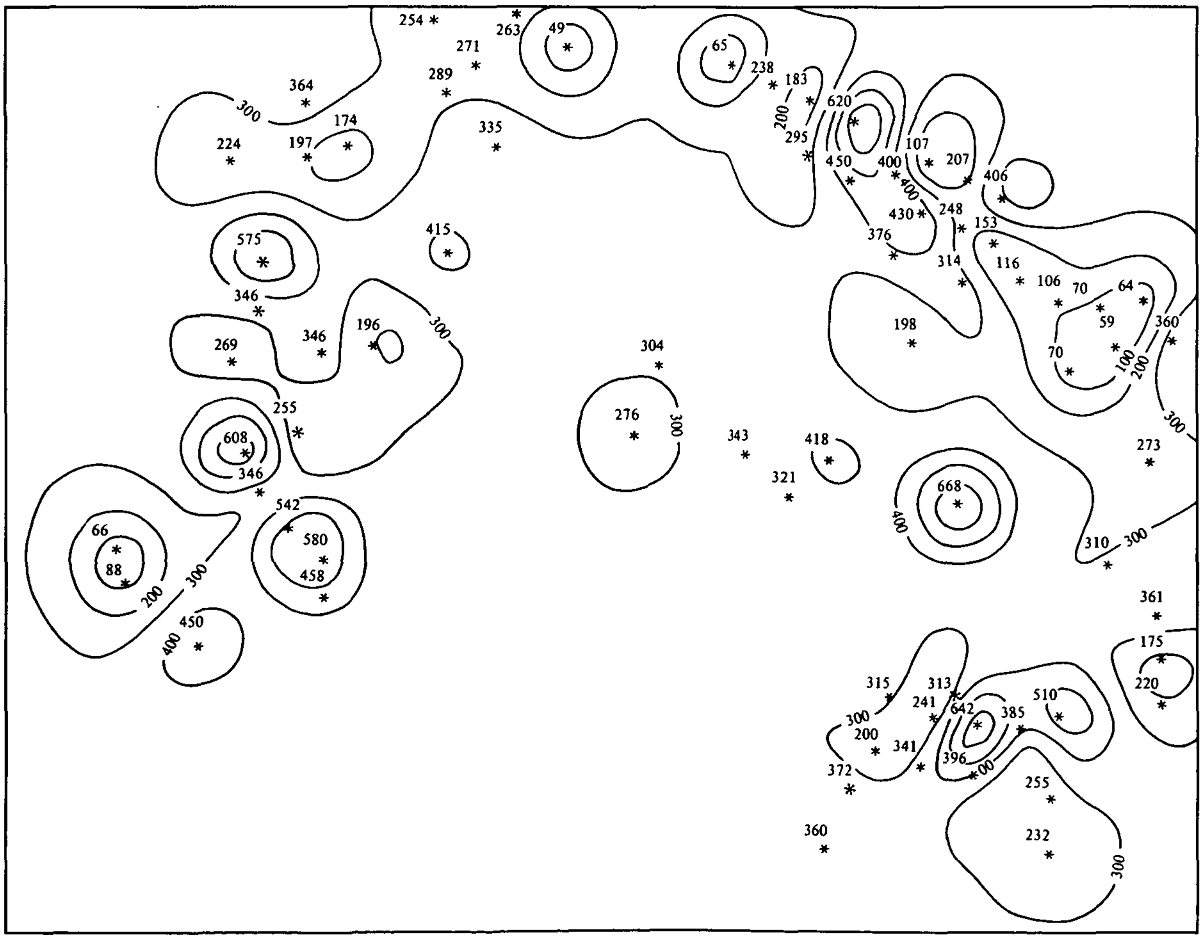

Figure 6. Shkapovskii data contoured by LUCAS.

ρ_S is fitted to the n values $\{z_i^E\}$, not using the maximum likelihood method but a faster method which gives a good approximation if n is large (greater than about 20 say). Basically, the method considers each point and its nearest neighbour (values z_i and z_j) and estimates the correlation between them, and hence the value of ρ_S. Repeating the procedure for each point and averaging gives an estimate of ρ_S which appears to be close enough to the value obtained by maximum likelihood for most practical purposes.

Thus, the interpolating function at an unknown point (x, y) becomes

$f(x, y)$ = Estimated trend from m average points using μ and ρ_L
+ Estimated residual from n data points using ρ_S.

RESULTS FROM TEST DATA

The ideas outlined in this paper have been incorporated in a program LUCAS, written in ALGOL-68R and running on the Loughborough University of Technology 1904 computer. Real data has been used to test the program, rather than artificially conceived data. The first set of test data to be considered is to be found in Ref. 5, and is a set of 72 measured permeability values from oil wells in a Russian oilfield, the Shkapovskii oil deposit. As a comparison the data has been contoured using a conventional package, GPCP[3] and the results are shown in Fig. 5. Although the contours presented are attractive and smooth, there are some obvious problems. The contours are not entirely consistent with the original data at some points, due presumably to the intermediate grid adopted by GPCP not being sufficiently fine. Furthermore, in the Southern part of the oilfield where there is actually no data, the program manages to produce some detailed contours, including an extremely steep cliff.

In comparison, LUCAS produced Fig. 6, a rather more boring picture. It reflects the fact that the data is actually not very highly correlated and therefore the contour map has not much structure away from the data points. Large blank areas appear, as much as to say 'unknown territory'. Thus, the contrasting philosophies of the two approaches result in radically different contour maps.

In order to test the two stage stochastic model, a different set of test data was used. This was provided by a colleague, Dr Roger Smith of the Mathematics Department at Loughborough, and relates to erosion of an iridium projection.

One hundred and nineteen data points were provided, and a trend was fitted which is shown in the contour map Fig. 7. In many ways this map may be of more interest than the map including the residuals and fitting all the data exactly, which is shown in Fig. 8.

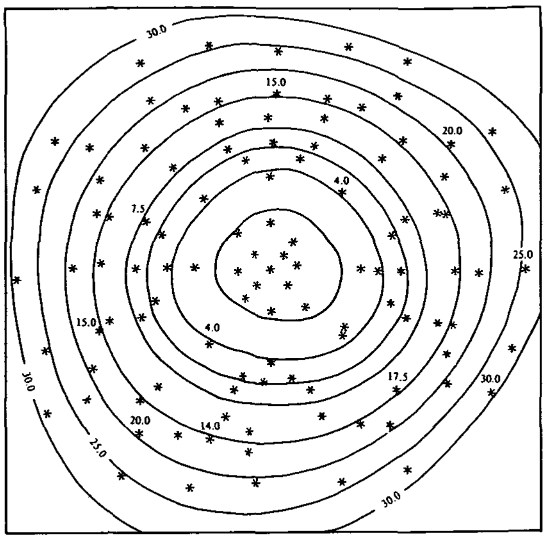

Figure 7. Trend contours. Symmetric iridium tip.

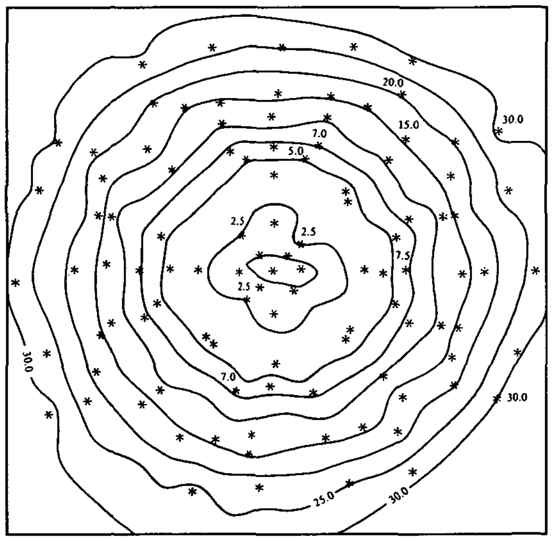

Figure 8. Full contours. Symmetric iridium tip.

CONCLUSIONS

A contouring algorithm has been derived for scattered data points which does not rely on the generation of an intermediate grid of nodal values. The interpolating function fits the data points parsimoniously, without producing 'imaginary' structures in regions not controlled by data points. Although it is difficult to devise an objective test for a contouring algorithm, the performance against examples of real data seems satisfactory. A two-stage stochastic model incorporating a trend as well as correlated residual has been developed, and further generalizations of the basic model are always possible.

REFERENCES

1. I. P. Schagen, *Interpolation in two dimensions—a new technique. J. Inst. Maths. Applics.* (1979).
2. D. H. McLain, Drawing contours from arbitrary data points. *The Computer Journal* **17**, 318–324 (1974).
3. CALCOMP Inc. *GPCP User's Manual* (1971).
4. C. Huijbregts and G. Matheron, Universal kriging. *Decision Making in the Mineral Industry. (Special Volume)* **12**, 159–169 (1971).
5. M. I. Shvidler, *Filtration flows in Heterogeneous Media.* Consultants Bureau, New York (translated from Russian). (1964).

Received March 1979