

AAEC/E368



AAEC/E368

**AUSTRALIAN ATOMIC ENERGY COMMISSION
RESEARCH ESTABLISHMENT
LUCAS HEIGHTS**

**CONTOUR PLOTTING FROM ARBITRARILY SPACED DATA POINTS
WITHIN A REGION OF ARBITRARY SHAPE**

by

M. R. DAVIDSON

October 1975

ISBN 0 642 99714 4

AUSTRALIAN ATOMIC ENERGY COMMISSION

CONTOUR PLOTTING FROM ARBITRARILY SPACED DATA POINTS
WITHIN A REGION OF ARBITRARY SHAPE

by

M.R. DAVIDSON

ABSTRACT

A triangular mesh with data points as vertices is determined. Points on the contours are calculated by linear interpolation along triangle sides. Allowance for general boundary shapes gives the method applications in physics and engineering.

National Library of Australia card number and ISBN 0 642 99714 4

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

COMPUTER CALCULATIONS; DIAGRAMS; INTERPOLATION; PLOTTERS; TRIANGULAR CONFIGURATION

CONTENTS

	Page
1. INTRODUCTION	1
2. THE METHOD	1
3. THE TRIANGULATION	2
4. CURVE FITTING	3
5. PROGRAM DESCRIPTION	4
6. CONCLUSION	6
7. REFERENCES	7

Figure 1 A triangular mesh determined by subroutine TRIANG

Figure 2 Streamlines contoured from data at the 139 vertices of the triangular mesh in Figure 1

Figure 3 The region is contoured in two sections containing 109 and 108 data points respectively

Appendix A

1. INTRODUCTION

Most existing methods of automatic contouring involve the interpolation of the data onto an organised grid. A surface may be first fitted to the data or the values at grid points may be obtained directly. A review of contouring methods has been given by Crain [1970]. More recently, papers by Palmer [1970], Barry [1971], Hessing et al. [1972], McLain [1974] and Briggs [1974] have appeared.

In manual contouring, there is no interpolation onto an intermediate grid. At best it involves simple linear interpolation between data points to determine the points on the contours. A method which closely approximates manual contouring is called 'contouring by triangulation' [Crain 1970].

In this report we develop a process for machine contouring based on triangulation. The method makes allowance for general boundary shapes so that it is applicable to results in physics and engineering (e.g. fluid flow patterns, stress lines). Application to geophysical and geographical data is severely limited by the time required to determine the triangular mesh. This restriction can be removed, however, if the mesh is predetermined.

2. THE METHOD

Consider an arbitrary array of data points which lie within and on a single continuous closed boundary of arbitrary shape. The boundary may consist of any number of smooth segments. Although this work applies only to a region with one closed boundary, it may be extended to cope with any number of boundaries. The method consists of dividing the region into a series of triangles with a data point at each vertex. Linear interpolation along the sides of the triangles determines the points on the contours, except at a curved boundary where the interpolation is taken along the boundary curve.

To find all of the contour lines of a particular level(h , for example) it is first necessary to determine where h -level contours meet the boundary (if ever). Beginning at the boundary, we can follow a contour from triangle to triangle until it meets the boundary again. When all boundary intersections have been accounted for, we begin the process from an internal point. We continue until all triangle edges cut by a h -level contour are accounted for.

To follow a contour from one triangle to the next, it is convenient to ensure that the contour never passes through a vertex (*i.e.* the contour height and the height at a vertex are never identical). Should this case occur, it is removed by temporarily altering the height at the vertex by a

small amount. If h is the height at the vertex, then the new height is given by $h_1 = (1 + \epsilon)h$; ϵ is chosen here to be 10^{-5} .

3. THE TRIANGULATION

Points in a random array may be joined by straight lines in such a way that the space between the points is divided into a set of triangles. This process has been discussed by Saraga & Wavish [1972]. They first introduce the concept of neighbouring pairs of points; triangulation is obtained by joining all such pairs of points by straight lines. Data points are vertices of the resulting triangular net. Saraga & Wavish define neighbours as follows:

A pair of points are neighbours if there exists no other pair of points in the array which satisfies both the following two conditions:

Condition (1) The second pair of points are closer together than the first.

Condition (2) A straight line drawn between the second pair of points intersects one drawn between the first pair.

To avoid the ambiguity produced when the two intersecting straight lines have the same length, they assume that the first pair of points are neighbours and that the second pair are not.

Now while this definition is adequate in many cases, it does not always yield a triangular net (see Appendix A). Also, in the above process, the region (D) occupied by the points is not clearly specified before triangulation. In our case, the data points lie within and on a prescribed boundary which encloses D.

To aid in a modified definition of neighbours, we define sets B and G as follows:

- . A pair of points is an element of B if both points lie on the boundary and they are adjacent.
- . A pair of points is an element of G if (i) the pair is not an element of B; and (ii) the straight line drawn between the two points lies entirely within D.

We determine the set of neighbouring pairs (P) as follows:

Step 1 Initialisation. Number the data points and let points 1 and 2 be neighbours. Set $I = 3$.

Step 2 Choose the I th point and consider the pairs consisting of point I and each of the previous $I - 1$ points. In the spirit of Saraga and Wavish's definition, we let such pairs be neighbours if (i) the pair is an element of B;

or (ii) the pair is an element of G such that there exists no element of P satisfying Conditions 1 and 2.

Step 3 Any element of P which satisfies Condition 2 but not Condition 1 is removed from P. However, for every element of P removed there must be a replacement pair which includes the point I, even if that pair has been previously disqualified from membership of P. Experience indicates that it is this step which ensures complete triangulation.

Step 4 If any points remain to be considered, set $I = I + 1$ and go to Step 2.

It is necessary, in dimensioning the work space in the contouring program, to determine the number of triangles and hence the number of neighbouring pairs for a given number of points. For any triangulation,

$$e = v + t - 1, \quad \dots(1)$$

where v is the number of vertices, t is the number of triangles, and e is the number of edges (neighbouring pairs). This result is a special case of Euler's formula for a polygonal net in a plane [Ore 1963]. If b is the number of boundary vertices, then

$$3t + b = 2e \quad \dots(2)$$

since adding three sides for each triangle counts each side twice except for those on the boundary. Equations (1) and (2) yield $t = 2v - b - 2$ and $e = 3v - b - 3$. For convenience, work space is allocated on the basis of $2v$ triangles and $3v$ edges.

4. CURVE FITTING

As mentioned previously, points on the contour lines are obtained by linear interpolation along the sides of the triangles. Those contour points may be joined either by a series of straight lines or by a smooth curve. However, the latter method is not recommended and is discussed shortly. The curve fitting procedure used here (primarily for curved sections of the boundary) is that of Akima [1970]. This is a new method which attempts to incorporate the features of manual curve fitting. The slope of the curve at each of the given points is determined by the slopes of the straight line segments joining nearby points. Each interval between successive points is fitted with a third order polynomial which is determined by the co-ordinates and the slopes at those points.

It should be noted that the above procedure takes no account of second derivatives and so should not be used without sufficient data to describe contour curvature. A smooth curve threaded through the contour points will attempt to round out sharp corners without heed to adjacent contour lines. This may result in contours of different heights crossing when sufficient data are lacking. In these cases such problems may be avoided by using the straight line representation. Also, if the mesh is coarse, errors in the calculated contour may result in unnatural wiggles in the fitted curve; again, the straight line representation is suggested.

5. PROGRAM DESCRIPTION

The program which produces the triangular mesh (TRIANG) and the contouring program (RCONT) may be invoked by using the following Fortran statements:

```
CALL TRIANG (X,Y,N,NB,OX,OY,XSCALE,YSCALE,WORK,NWORK)
CALL RCONT (X,Y,H,C,N,NB,NC,OX,OY,XSCALE,YSCALE,WORK,NWORK)
```

Additional statements which provide for curved boundary segments and smooth contours will be described later.

- (i) N is the total number of data points.
- (ii) NB is the number of data points on the boundary.
- (iii) NC is the number of contour levels to be plotted.
- (iv) X, Y, and H are each arrays of dimension N. The Ith data point has height H(I) with rectangular co-ordinates (X(I),Y(I)). The first NB data points must be boundary points taken in order. The remaining data points may then be inserted in any order.
- (v) C is an array of dimension NC and contains the contour heights to be plotted. Each height can be represented on the boundary at most 50 times. Non-existent contour heights are ignored and a message is produced to that effect.
- (vi) (OX,OY) are the co-ordinates in inches of a new origin.
- (vii) XSCALE and YSCALE are the plotting scales in data units per inch for the X and Y co-ordinates respectively.
- (viii) NWORK is an integer working array of dimension at least 18N and WORK is a real working array of dimension at least 8N.

Program TRIANG calculates the mesh data for the program RCONT and produces a plot of the mesh (on one plotting page only). If we view NWORK as a 2-dimensional array whose first and second dimensions are exactly 6N and 3 respectively then, after the triangulation, array elements NWORK (4N + I,1), NWORK (4N + I,2), NWORK (4N + I,3) always contain the

numbers of the data points which form the vertices of the Ith triangle.

The triangulation step can be very time consuming (see later); it is therefore desirable to minimise the number of calls to TRIANG. For a given set of co-ordinate data, the program TRIANG need only be invoked once; the mesh data may be stored separately to be read back when required.

In the absence of any of the optional statements (described below), RCONT produces a plot in which

- (a) the boundary, which is assumed to consist of straight line segments, is plotted; and
- (b) the interpolated points representing a particular contour line are connected by straight lines.

The printout provides information about the number of interpolated points on each contour, the co-ordinates of the end points of the contours (useful for identification purposes), and whether the contours are open or closed.

Any of the following optional statements may be made, before calling RCONT, in order to modify the output.

- (i) CALL CURVE (NCNR, NCVE)

NCVE is the number of curved boundary segments and NCNR is a 2-dimensional array whose first and second dimensions are 2 and NCVE respectively. If $N1 = NCNR(1,I)$ and $N2 = NCNR(2,I)$, then the data points numbered $N1$ and $N2$ (both are boundary points) mark the ends of the Ith curved segment. $N1$ and $N2$ must be chosen so that the curve beginning at $N1$ and ending at $N2$ passes through at least one other point and moves in the direction of the boundary point numbering, i.e. we require

$$\begin{aligned} N2 + NB - N1 - 1 > 0 & \quad \text{if } N2 < N1, \\ N2 - N1 - 1 > 0 & \quad \text{if } N2 > N1. \end{aligned}$$

In the first case, either the first or the last boundary point lies on the curve between points $N1$ and $N2$.

If the boundary is a single closed smooth curve then $NCVE = 1$ and $N1 = N2$.

- (ii) CALL NOLINE (NDEL, NK)

NK is the number of straight or curved segments between consecutive boundary points which we wish to delete from the output. The array NDEL of dimension NK contains the segment numbers which we wish to omit. The boundary segments are numbered from the first data point, i.e. the line joining point 1 to point 2 is line 1, and the line joining point 2 to point 3 is line 2, etc.

(iii) CALL SMOOT

This statement causes the calculated contour points to be connected by smooth curves. Note that this statement does not affect the boundary points which may also lie on smooth curves.

(iv) CALL CWRITE

This causes the following statement to be executed for each interpolated contour point:

```
WRITE (8) XC, YC
```

where (XC, YC) are the co-ordinates of the point.

When calling program TRIANG, it is not necessary for the user to begin and end the resulting plot; this task is performed by the program. However, immediately before the first call to program RCONT, a plot must be started by the user with CALL GPSEND (1, M), where M is the number of plotting pages to be allocated. After RCONT has been called, the user is required to terminate the plot with CALL GPSEND(2), after additional plotting (if any) has been performed. This includes further calls to RCONT and its associated options (which must be reset otherwise the default options apply).

Details of the plotter and associated software used here are described in unpublished work by Cox (undated).

6. CONCLUSION

As indicated earlier, the method is limited by the time required to determine the triangular mesh. Typical computation times on the 360/65 are shown in the following table. These are the times taken to triangulate random data within a square possessing 16 boundary points.

The tabulated results show an enormous increase in computation time with an increasing quantity of data. However, a considerable saving in time results if the region is contoured in sections (all in the one plot) using repeated calls to RCONT. For example, if a region containing 200 data points is contoured in two sections each containing 100 points, a time saving of more than 50 per cent results.

<u>Number of data points</u>	<u>Approx. time for triangulation</u>
50	18 s
100	1 min 23 s
150	3 min 22 s
200	6 min 32 s

Figures 2 and 3 are demonstration contour plots showing the streamlines for Stokes flow entering an expanding sphere [Davidson & Fitz-Gerald

1972]. Figure 1 is the triangulation associated with Figure 2. Figure 2 is contoured from 139 data points, and Figure 3 is contoured in two sections containing 109 and 108 data points respectively.

7. REFERENCES

- Akima, H. [1970] - A new method of interpolation and smooth curve fitting based on local procedures. *J. ACM (Assoc. Comput. Mach.)* 17:589.
- Barry, J.M. [1971] - A method of producing contour lines from rectangularly spaced data. AAEC/TM582.
- Briggs, I.C. [1974] - Machine contouring using minimum curvature. *Geophysics*, 39:39.
- Cox, G.W. (undated) - Software requirements for the Calcomp 565 plotter. AAEC unpublished report.
- Crain, I.K. [1970] - Computer interpolation and contouring of two-dimensional data: A review. *Geoexploration*, 8:71.
- Davidson, M.R. & Fitz-Gerald, J.M. [1972] - Flow patterns in models of small airway units of the lung. *J. Fluid Mech.*, 52:161.
- Hessing, R.C., Lee, H.K., Pierce, A. & Powers, E.N. [1972] - Automatic contouring using bicubic functions. *Geophysics*, 37:669.
- McLain, D.H. [1974] - Drawing contours from arbitrary data points. *Comput. J.*, 17:318.
- Ore, O. [1963] - Graphs and their uses. Random House, NY.
- Palmer, J.A.B. [1970] - An economical method of plotting contours. *Aust. Comput. J.*, 2:27.
- Saraga, P. & Wavish, P.R. [1972] - Edge tracing in binary arrays. In 'Machine Perception of Patterns and Pictures.' Conf. Series No. 13. Institute of Physics, London, p294.

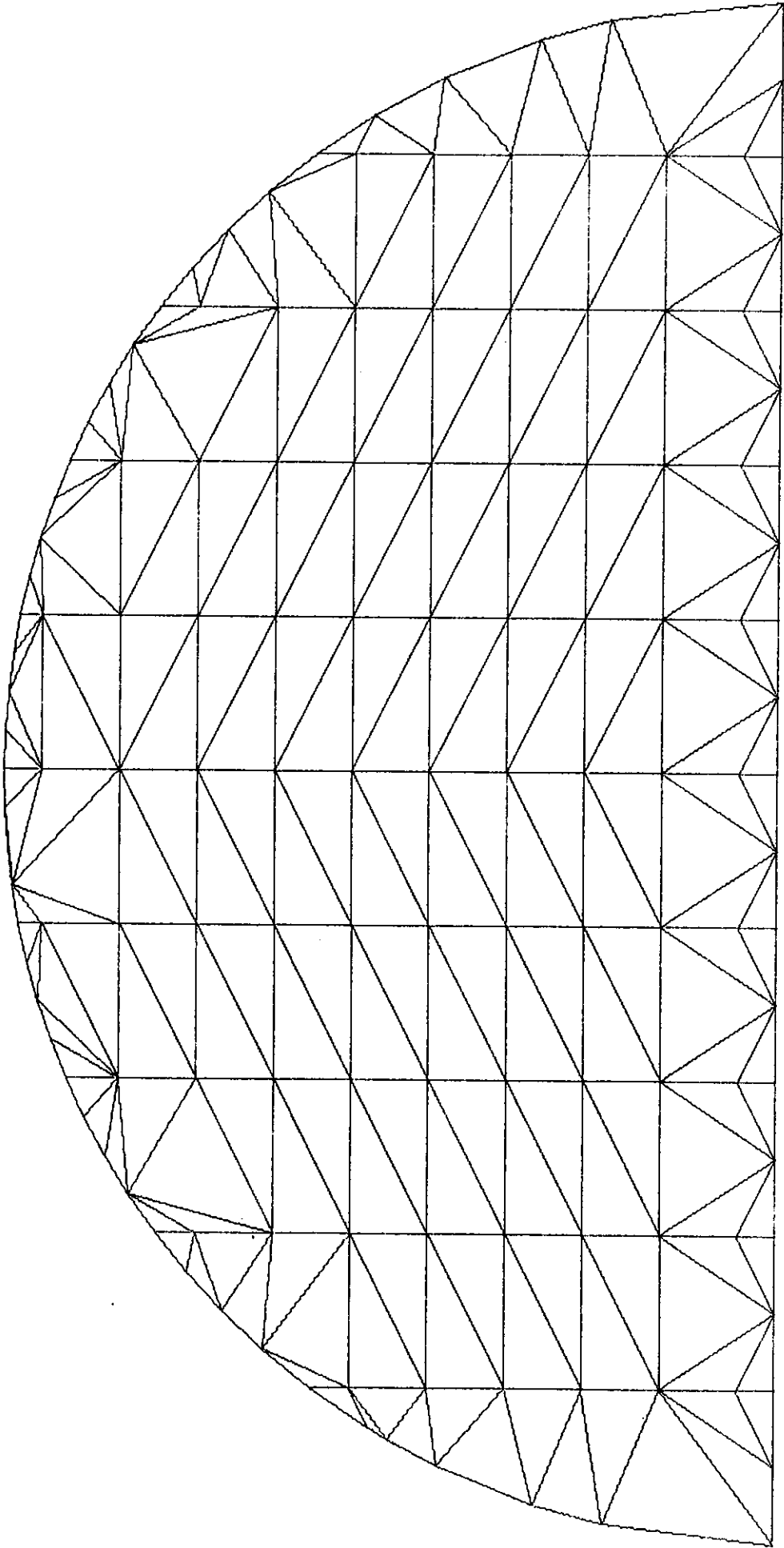


FIGURE 1. A TRIANGULAR MESH DETERMINED BY SUBROUTINE TRIANG

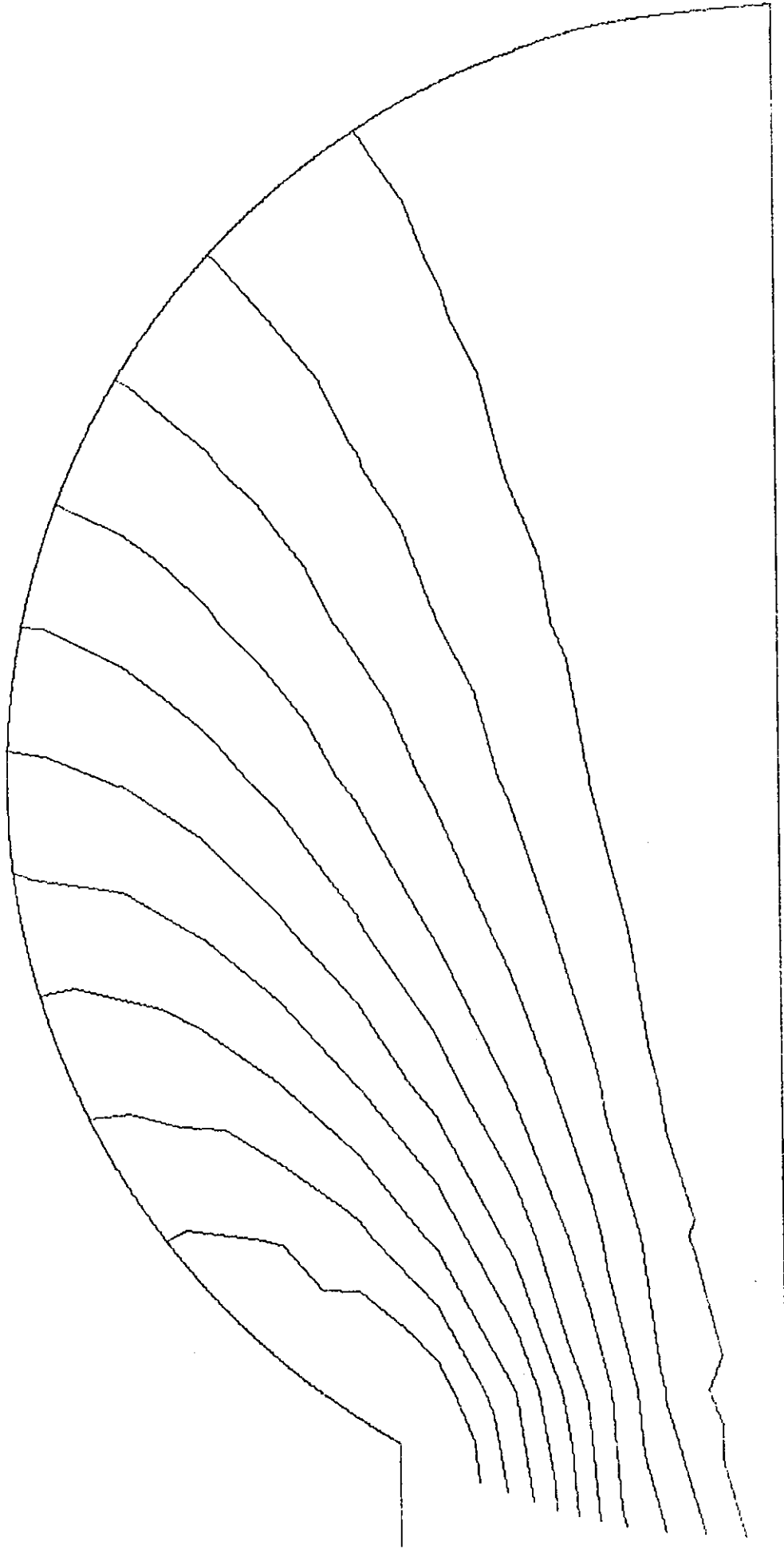


FIGURE 2. STREAMLINES CONTOURED FROM DATA AT THE 139 VERTICES OF THE TRIANGULAR MESH IN FIGURE 1

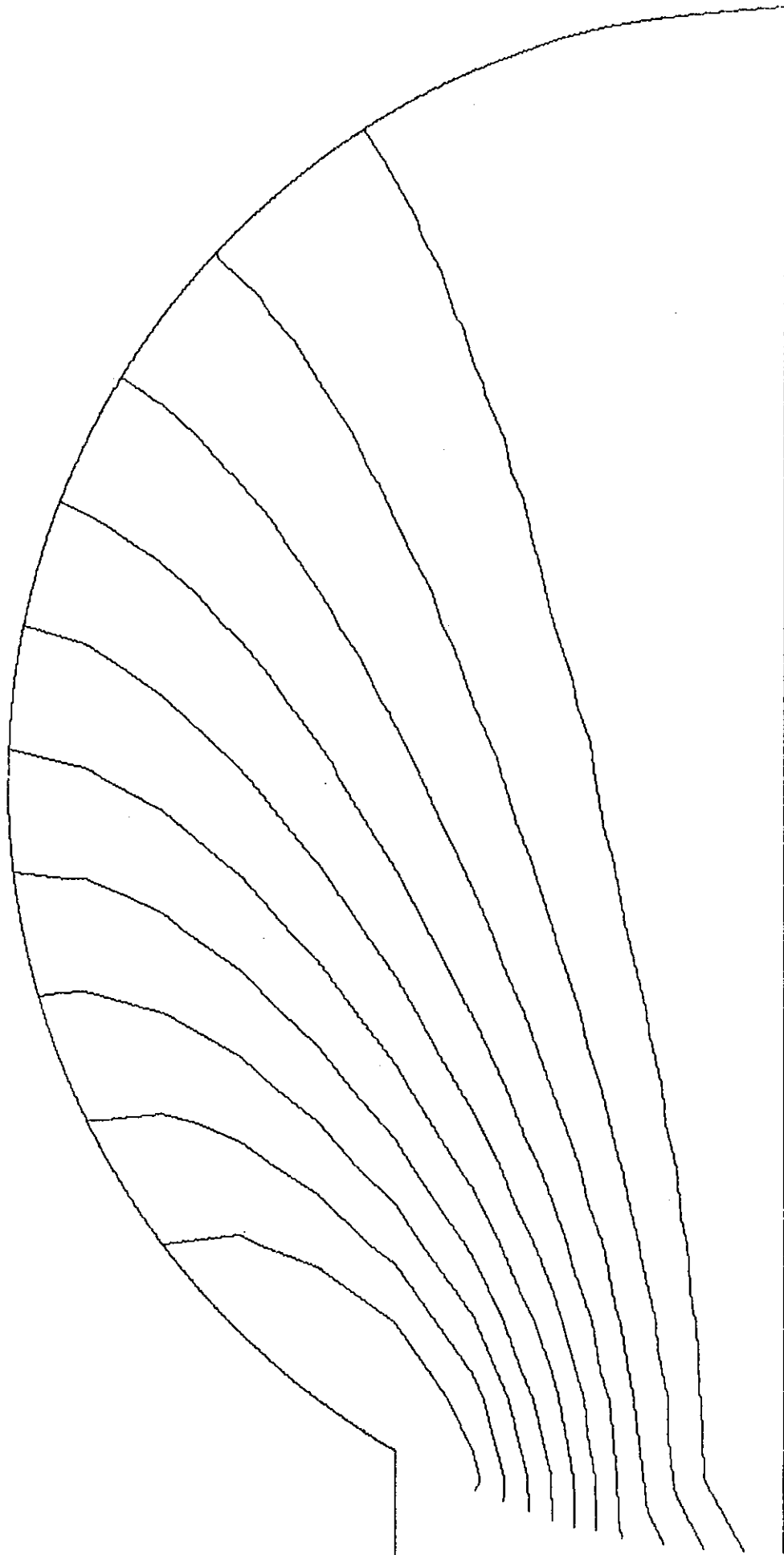


FIGURE 3. THE REGION IS CONTOURED IN TWO SECTIONS CONTAINING 109 AND 108 DATA POINTS RESPECTIVELY

APPENDIX A

Let ABCD be a unit square as shown in Figure A1. Choose points P, Q, R to be on the perpendicular bisectors of sides AD, AB, BC respectively such that $PL = QN = RM = h$. Let $PQ = x$. We can easily show that for $0 < h < \frac{1}{2}$, we have $\frac{1}{\sqrt{2}} < x < \sqrt{2}$, i.e. there exists an $h < \frac{1}{2}$ for which $x > 1$ (since $\sqrt{2} > 1$). In this circumstance, PQ intersects AB and AD but is the longer line; and so by the definition of Saraga & Wavish [1972], P and Q cannot be neighbours. Similarly AC is longer than PQ, so A and C cannot be neighbours. The result of joining the neighbouring pairs is shown by the solid lines in Figure A1; complete triangulation does not occur.

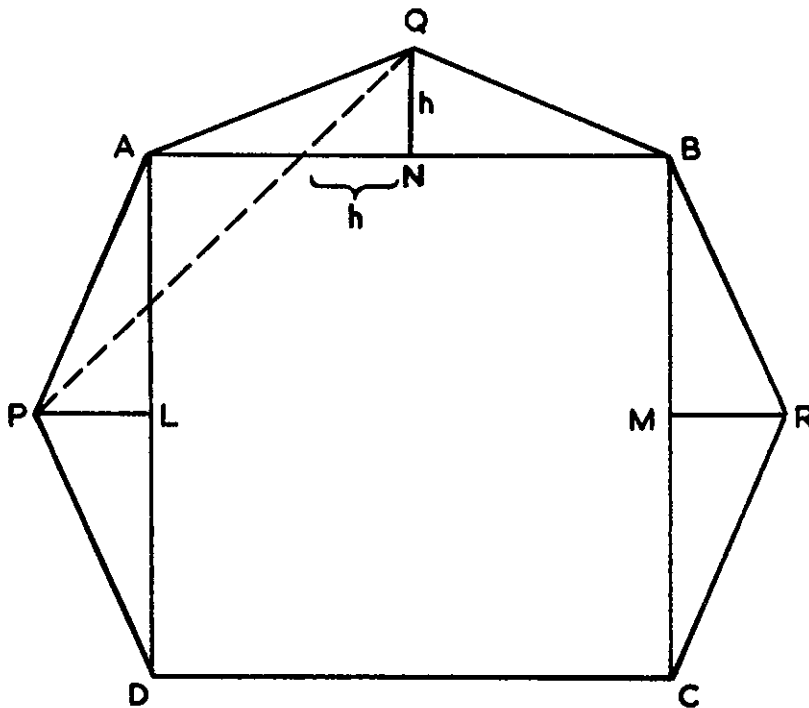


FIGURE A1. AN EXAMPLE OF A SET OF POINTS A,B,C,D,P,Q,R WHICH DO NOT BECOME THE VERTICES OF A TRIANGULAR MESH, USING SARAGA AND WAVISH'S DEFINITION OF NEIGHBOURS

