GENERAL PURPOSE CONTOUR PLOTTING ROUTINES

by

P.J. Zsombor-Murray
DATAC Computer Laboratory
Department of Mechanical Engineering
McGill University

Contour mapping is an effective way of transforming a collection of discrete data into a family of equipotential curves. This form of graphical presentation finds wide application in disciplines concerned with potential distributions, natural or abstract. Two methods are discussed. One requires data to be on a uniform rectangular pitch and is described because it is representative of a number of earlier techniques. The other makes direct use of randomly distributed data points by forming them into a mesh of triangular cells. The relationship between neighbouring cells is easily described by their node numbering sequence. This makes the process of ordering level points into contour strings quite convenient.

MÉTHODES EN HYPSOMÉTRIE PAR L'ORDINATEUR

par

P.J. Zsombor-Murray
Laboratoire d'Ordinatrice DATAC
Département Génie Mécanique
L'Université McGill

L'Hypsométrie est une méthode servant à transformer un assemblage de données discontenues en une famille à caractéristiques équipotentiels. Cette procédure graphique se trouve des usages diverses dans les disciplines reliées aux fonctions potentielles. Deux méthodes pour construire les cartes hypsométriques sont presentées. La première méthode utilise des points données en espacement uniforme et rectangulaire et ressemble de près à certaines méthodes utilisés depuis déjà plusieurs années. Avec l'autre méthode les données prises au hasard forment une maille de cellules triangulaires. Les relations entre cellules adjacentes sont décrites par la séquence de numéro nodal du sommet de chaque cellule. On peut ainsi très facilement ordonner les points de contour.

illustrated in Fig. 6a for all horizontal segments and those in Fig. 6b for the remaining vertical segments. When the search routine is complete the vector pairs (X,Y) and (XB,YB) contain all points along grid segments at level ZC. These points may, like P(k',l) in Fig. 4, lie between nodes of a horizontal segment, may lie on a vertical segment like P(i,j') or may be at a node Z(I,J) = ZC like P(g,h). Note that each of these 3 distinct configurations is defined by the relationship between actual, (X,Y), and base, (XB,YB), coordinates. These relationships are necessary to establish the criteria by which any given point in (X,Y) can be legally connected to any other point in (X,Y). The search routine is constructed so as to eliminate any ambiguity as to what constitutes the lower left corner of a cell. Only those points which lie on the L which proceeds upwards and to the right of a node, but does not include the 2 nodes which terminate the arms of the L, will have as their base the node at the intersection of the arms.

The second major block in Fig. 2 is the sort routine. A preliminary step in generating ordered sequences of coordinates at the current potential value ZC is to determine whether the disordered set (X,Y) contains branches which are

1. Comprised of a single point,
2. Potentially open, 2 ended curves,
3. Potentially closed rings.

Possible forking, a conceivable contour condition, is not entertained by this algorithm. Singular points are easily identified as those which cannot be validly joined to any other point. A valid or legal link exists only among points sharing the perimeter of a cell. A point in (X,Y) which links to only one other point signifies a potentially open branch. The term -potentially-open is used to acknowledge that lorgnette-like structures such as that shown in Fig. 7a, along with a simple fork, may exist in some topographies. The algorithm, however, forms open curves, as shown in Fig. 7b, instead because the possibility of a closing arc is lost as points are deleted from the disordered set (X,Y). A potentially closed curve is characterized by a remaining set (X,Y) in which each member can be validly linked to 2 or more others. Although (X,Y) might be in a potentially closed curve state, a contour branch closing upon itself will not necessarily emerge. Links are formed to the closest eligible point remaining in (X,Y). Therefore an (X,Y) buffer in state 3. may well yield an open curve with some points left over. This remainder may in turn be potentially open or closed or even in a unique state 1. How any given buffer in state 3. is disposed of depends on the situation of X(IQLAST), Y(IQLAST), the last remaining point in (X,Y). A possible result is illustrated in Fig. 8. A snapshot of the unsorted buffer (X,Y), the sorted buffer (U,V), the vector (LS) and the counters JI and JJ may reveal a structure similar to that shown in Fig. 9. Points selected from (X,Y) and appended to the strings formed in (U,V) are deleted from (X,Y). Elements in (U,V) are counted as JI whilst the number of separate contour branches are counted as JJ. Each completed branch in (U,V) has its length recorded in the vector (LS). The subscript IXYS refers to the eligible point closest to the most recently selected point U(JI-1),V(JI-1). The sorting process, repeated for each branch, begins with an attempt to discover a type 1. or type 2. link. If this attempt fails a type 3. branch is assumed. This tentative linking, as well as the linking which follows to actually form the point strings in (U,V), relies on the relationship between every point in (X,Y) and its counterpart in (XB,YB) as shown in Fig. 4. Therefore all points which may legally connect with any given point are discovered by searching 6 grid segments as shown in Fig. 10a

Contour mapping is a means of interpolating between successive pairs of points of given potential value so as to construct a set of points of a desired, constant potential value; subdividing this set into ordered subsets and passing appropriate curves through these sequences; thus producing equipotential lines or contours. The interpolation and ordering procedures are facilitated if the original points represent values on the nodes of intersection of a uniform, rectangular grid and the values are presented as a matrix whose elements correspond to the nodes of the grid. The methods proposed by Cottafava and LeMoli (1) are based on this principle. Even commercial contour plotting packages such as (2), which accepts arbitrarily distributed potential values, construct a uniform rectangular grid with nodal values inferred from those arbitrarily distributed.

A particular method of drawing contours on a uniform rectangular grid is described in detail below, not because it differs markedly from earlier methods but because

1. These techniques and the inevitable difficulties which accompany them are best understood in the context of a specific algorithm,
2. In view of the many scientific and engineering applications of contour plotting, the literature is conspicuously devoid of program listings or algorithms,
3. The package described in (2) requires very large disc storage and, with an online plotter, is quoted as incurring run times of hours. The program described herein occupies less than 10000 core locations, including array storage, compiles in less than 3 minutes, including read-in, and produces annotated contour maps, typically, in 2 minutes. Useful contouring routines need not be huge, unwieldy and expensive.

The input to this program, punched on cards, is described in the table of Fig. 1. Before proceeding with the search for interpolated contour values, ZC, from a minimum of ZMN to a maximum of ZMX in increments of ZI, an exponential scale factor, IFACT, is calculated. This is used to compute a value which is, instead of ZC, annotated to contour lines. In this way all labels can be presented in the same format; 3 numerals, 2 after the decimal point, and a sign. This computed value is

$$Z_{label} = ZC*10.**IFACT$$

For example if ZMN = -15. and ZMX = 753. then IFACT = -2 and the first contour would be labelled -0.15, the last b̸7.53.

During the first iteration all contours for the current value ZC = ZMN are formed. ZC is then incremented, ZC = ZC + ZI, and the next contour family is formed. This is repeated to ZC = ZMX. The entire process is summarized in Fig. 2. The 3 major blocks thereon bear elaboration.

The first block is the search routine. This operates on the contents of the nodal value matrix, Z(I,J), shown in Fig. 3. For all horizontal and vertical grid segments, both nodes are checked to establish whether the value ZC exists at either node or along the internodal segment. If such exists, its coordinates are stored in (X,Y) along with the coordinates of the lower left corner of the rectangular cell containing it. These latter or base coordinates are stored in (XB,YB); see Fig. 5. The search routine performs the steps or 10b or by searching 8 segments as in Fig. 10c. The algorithm which determines whether (X,Y) is in state 1., 2. or 3. for each separate branch ZC and builds (U,V), (LS), JI and JJ is illustrated in Fig. 11.

After all branches of a contour level ZC have been inserted as point sequences into (U,V) -i.e. when (X,Y) has been exhausted with IQLAST decremented to 0- the contour lines are drawn, starting with the last string of length LS(JJ). Each subset of points in (U,V) containing more than 1 point is searched to establish the rectangular range which embraces the maximum and minimum x and y coordinate in that subset. This information is required to fit a smooth curve through the points. One or, if the number of points warrant it, a sequence of parametric cubic curves are fitted through the points. For a detailed description of this type of curve, known as an F-curve or T-conic, and its properties, refer to the excellent thesis by Forrest (3), especially pp. 40-42, and to other work by Ferguson (4) and by Rowin (5). Each contour line is labelled, at its beginning, with numerals in a 5/8x1/8 in. high rectangular field whose lower left corner is coincident with the first point on the contour line and whose length is sloped parallel to the line between the first and second points.

Before dealing with a contouring routine which forms a triangular mesh from a random distribution of data points and takes advantage of its connective topology to combine the search and sort procedures into one step, consider the alternative. This is illustrated by the following simple example. A collection of 13 randomly distributed points, $P_i(x_i,y_i)$, of level z_i is shown in Fig. 12a and a uniform rectangular grid is required such that the levels z_g associated with the nodes $N_g(x_g,y_g)$ convey a reasonable topographic equivalent to z_i. The suggested procedure begins by choosing a rectangle whose sides are parallel to x,y and contain the 4 extreme points in P_i; P_{x+}, P_{x-}, P_{y+}, P_{y-}; Fig. 12b. This rectangle is divided into n/2 cells, where n is the number of given points. If n = 13, n/2 is truncated to 6 or the smaller number nearest n/2 in the progression

$$((j + 1)/2)((j + 2)/2), \quad j = 1, 2, 3, \ldots$$

and the number of nodes is given by

$$((j + 1)/2)((j + 2)/2) + j,$$

a 12 node grid for n = 13. Node levels may be reasonably defined by the nodal level of the plane containing 3 points closest to that node. To be consistent with the rule that points on a grid segment can only influence contours within cells which share that perimeter, no node shall be defined if less than 3 points occupy the cells, including perimetric grid segments, which share that node. 3 points are required to establish the plane which defines the nodal level. The 13 point example cluster will have its SE and SW corner nodes thus eliminated as shown in Fig. 12d. Surviving grid points, P_g, will have levels given by

$$z_g = z_3 + (z_i - z_3)(((x_g - x_3)^2 + (y_g - y_3)^2)/ \\ ((x_i - x_3)^2 + (y_i - y_3)^2))^{1/2}$$

The relation between z_g and the 3 points P_1, P_2, P_3 which define it is illustrated in Fig. 12e.

Two objections can be raised against preinterpolation of a randomly distributed collection of potential values onto a uniform grid.

1. As a linear or any other arbitrary interpolation technique is an approximation of the actual topography, applying it twice, once to establish the new nodes, again to establish the level points along the rectangular cell segments, may result in an unacceptable loss in fidelity between map and original data set.
2. The freedom to present a random distribution of data points represents an efficient way to vary the fineness of the mesh. A closely spaced network is requred in regions of highly convoluted contours. Where there is relatively constant gradient, a coarser net with wider spacing between points is sufficient. The uniform grid transformation described above cannot exploit the information in regions of high point density.

A better approach consists of a scheme whereby

1. A mesh composed of triangular cells, whose nodes include only and all of the given data points, is formed and
2. The cells are systematically searched so as to assemble all nodal and segmentally interpolated level points into individual contour branch sequences.

These two operations are equivalent to the search and sort routines pertaining to the uniform grid contouring method described before. However it is herein that the two methods differ most significantly. The only other difference that need be pointed out is that the uniform grid method sorted out all contour branches into (U,V) and kept track of the length of each point string in (LS) and of the number of branches in JJ. The triangular cell method sorts and plots one branch at a time. This is because a triangular mesh invariably contains many more cells, hence segments, than does a uniform rectangular grid composed of the same number of nodes, and storage is therefore more at a premium. Generally it is not possible to say how many more nodes or segments but if it is assumed that a given m x n uniform grid, Fig. 13a, is represented as an equivalent triangular mesh, Fig. 13b, the respective number of segments and cells in each grid can be seen to be

$2mn - (m + n)$ segments and

$mn - (m + n) + 1$ cells for the rectangular grid and

$3mn - 2(m + n) + 1$ segments and <u>not</u>

$2(mn - (m + n) + 1)$ cells but

$2(mn + (m + n) - 1)$ cells for the triangular mesh.

This latter, apparently incorrect, number of triangular cells includes as open cells the segments which form the perimeter of the mesh. The reason for this becomes clear upon examining how a cell is defined and how this definition is used to construct the mesh. A cell is named by any clockwise sequence -a consistent counterclockwise sequence would serve as well- of indices which identify the points at the cell vertices. The randomly distributed points are given in the triple vector (X,Y,Z). The cells are named in the triple vector (IT1,IT2,IT3), (IT) for short. Fig. 14 shows a 10 point mesh defined in this way. Note that any forward pairing, $IT1_i - IT2_i$, $IT2_i - IT3_i$ or $IT3_i - IT1_i$, is unique. Also, every reverse pair, $IT2_i - IT1_i$, $IT3_i - IT2_i$ or $IT1_i - IT3_i$, appears only once as a forward pair, $IT1_j - IT2_j$, $IT2_j - IT3_j$ or $IT3_j - IT1_j$, elsewhere in the table. With reference to Fig. 14 the network of triangles is constructed by the following steps:-

1. Set all triplets in (IT) to 0.
2. Pick any point, say -1-, and arbitrarily call it IT1(1).
3. Pick the point closest to -1-, imagine it is -2-, and enter it as IT2(1). If there are ties, arbitrarily select the first equally qualified candidate in hand.
4. Select a third point such that the vector cross-product

$$\begin{bmatrix}(X(2) - X(1))\bar{i} \\ (Y(2) - Y(1))\bar{j}\end{bmatrix} \times \begin{bmatrix}(X(3) - X(1))\bar{i} \\ (Y(3) - Y(1))\bar{j}\end{bmatrix} > 0$$

   and (X(3),Y(3)) is the eligible point closest to

$$\left(\frac{X(1) + X(2)}{2}, \frac{Y(1) + Y(2)}{2}\right)$$

5. If no such point is available leave IT3 blank.
6. Scan (IT) over the range of discovered triplets for the first forward pair sequence which does not yet appear as a reverse pair and enter it as the next IT1, IT2.
7. Continue the sequence of steps 4.-5.-6. until no more triplets, open or closed, can be found.

The cascade of arrows on the table in Fig. 14 shows the order in which triplets were generated. Note that only the first full triplet can have 3 offspring, one built upon each segment. No progeny proceed from open triplets except for a sole descendant when an IT3(1) cannot be found. A maximum of 2 but not less than 1 triplets issue from all other triindicial sequences.

Points at the current level, ZC, are found and assembled into strings by using (X,Y,Z) and (IT1,IT2,IT3). The segments and nodes of every cell are searched, cell by cell in (IT), until a cell which contains at least one point at level ZC is found. A closed cell might have 0, 1, 2 or 3 such points. There are 27 possible combinations of nodal values of the 3 states: Z < ZC, Z > ZC and Z = ZC. An open cell might have 0, 1 or 2 such points in 9 possible configurations. These are enumerated in Fig. 15 with the nodal states represented by +, - and o respectively. Whether the search of a given cell is successful or not, that cell is interchanged with the lowest members in the set (IT) still within the search range. Then the length of the upper region in (IT) is decremented. When the first cell containing any points at level ZC is found, the coordinates of 1 or 2 such points are placed as the first entries in (U,V). The node indices of the segments upon which level points were found are recorded in a 4 element table, (IR). If a contour level occurred at a node, only one number is placed in one or the other element pair of (IR). Links to points in (U,V) are usually made by examining the cell containing the node pair IR(1)-IR(2) in reverse order and then overwriting the numbers in IR(1) and IR(2) with the node sequence of the segment in the new cell which contains a linking point. If no link is found, IR(1) and IR(2) are zeroed out. The linking point corresponding to the segment indicated by the new IR(1) and IR(2) is added to the end of the list in (U,V). When no more points are found using IR(1) and IR(2), the other end of the contour line, terminating on segment IR(3)-IR(4) or on node IR(4) if IR(3) = 0, is built in a similar manner until it too terminates. Additions to (U,V) found using IR(3) and IR(4) are added to the beginning of the list after pushing all previous elements in (U,V) down 1 step. JI contains the current length of (U,V). When (IR) is blanked out and JI > 0, the contour branch in (U,V) is plotted. As long as level points are discovered only upon segments between nodes there is never an alternative link; if it exists it is unique. Such cells are entered on one segment and, if possible, left via another. Alternate links occur only when a cell is entered on a node at level ZC. When a link to a node is made, all remaining cells containing that node are searched and the closest-level-point-in-hand criterion is used to select the path and to break ties. All these searched cells which radiate from the nodal hub at level ZC are discarded. Any nonselected points in these discarded cells have been or will be picked up in other branches or even further along in the current branch. No eligible point at level ZC can ever be overlooked. It sooner or later appears in an appropriate contour string in (U,V). Another attractive feature of the triangular mesh structure is that strings which define closed contours are built easily and naturally. There is no need to establish whether the remaining mesh is potentially closed and to store the coordinates of the starting point. The cell which shares the segment, containing the starting point, with the first cell will eventually be entered. Then the starting point is rediscovered and appended to (U,V) thus closing the ring. This is shown in the simple example of Fig. 16. Say the string is begun with cell 1-3-2 and points B and A are placed in (U,V) first. (I,R) receives the pairs 1-3 and 3-2. Building onto 1-3 with the reverse order 3-1 reveals the cell 3-1-4 which adds C after A in (U,V). 1-3-2 is eliminated by placing it at the bottom of (IT). The sequence 4-3 replaces 1-3 in IR(1) and IR(2). Continuing in this way, cells 3-4-5, 3-5-6 then 3-6-2 are examined and IR(1) and IR(2) successively become 5-3, 6-3 then 2-3, adding D, E and finally B again to (U,V). This end of the contour sequence in (U,V) now terminates and IR(1) and IR(2) are blanked out. Turning to IR(3) and IR(4) which still contain 3-2, the attempt to build onto the top of the list in (U,V) above B fails immediately since the cell containing 2-3 has already surrendered B to the other end of the list and has submerged beneath the limit of the search range at the end of (IT). Once (IR) has been blanked out, the string in (U,V), if it exists, is plotted using the same routine as that used for the uniform grid. The triplets remaining in (IT) are all external segments. Each will produce a separate null set in (U,V) and each, in turn, will sink beneath the rising search range level. In the end (IT) will be in a different order than when the building process began. However this is immaterial with respect to the formation of subsequent contours, i.e. those at higher levels. A typical contour search of a somewhat larger, 10 node mesh is summarized in Fig. 17.

References

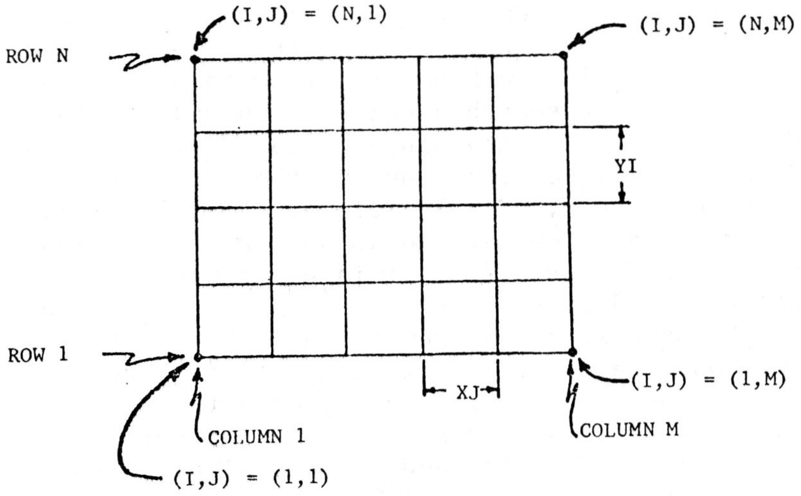
1. Cottafava, G. and LeMoli, G. "Automatic Contour Map", Comm. ACM 12, 7 (July, 1969), 386-391.

2. IBM Application Program H20-0357-0, 1130 Numerical Surface Techniques and Contour Map Plotting (1130-CX-11X) Programmers' Manual, 1971.

3. Forrest, A.R. "Curves and Surfaces for Computer-Aided Design", PhD Thesis, University of Cambridge, 1968.

4. Ferguson, J.C. "Multivariable Curve Interpolation", J. ACM 11, 2 (April, 1964), 221-228.

5. Rowin, M.S. "Conic, Cubic and T-Conic Segments", The Boeing Company, Document D2-23252, (April, 1964).



M .... NUMBER OF POINTS PER ROW OR NUMBER OF COLUMN ELEMENTS IN POTENTIAL VALUE MATRIX (≤ 20)

N .... NUMBER OF POINTS PER COLUMN OR NUMBER OF ROW ELEMENTS IN POTENTIAL VALUE MATRIX (≤ 20)

ZMN .... MINIMUM VALUE OR LOWEST CONTOUR TO BE PLOTTED

ZMX .... MAXIMUM VALUE OR HIGHEST CONTOUR TO BE PLOTTED

ZI .... INCREMENT OF CONTOUR VALUE, A CONSTANT

XJ .... INTERCOLUMN GRID SPACING

YI .... INTERROW GRID SPACING

Z(I,J) ..... POTENTIAL VALUE AT EACH GRID NODE, READ IN ROW BY ROW STARTING WITH THE LOWER LEFTMOST AS Z(1,1) AND PROCEEDING TO THE RIGHT OVER THE RANGE J = 1,M THE LAST OR UPPERMOST ROW IS THE N'th (≤ 20x20)

FIG. 1

Z(N,1)	Z(N,2)	...	Z(N,M)
⋮	⋮		⋮
Z(2,1)	Z(2,2)	...	Z(2,M)
Z(1,1)	Z(1,2)	...	Z(1,M)

FIG. 3

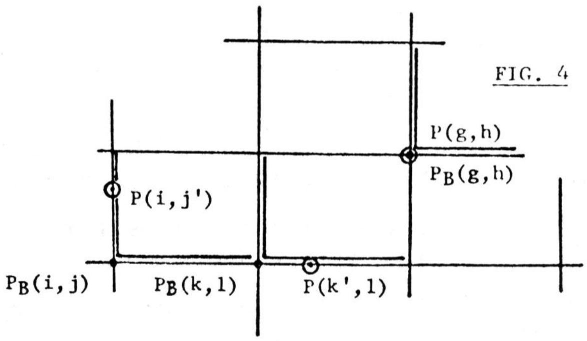
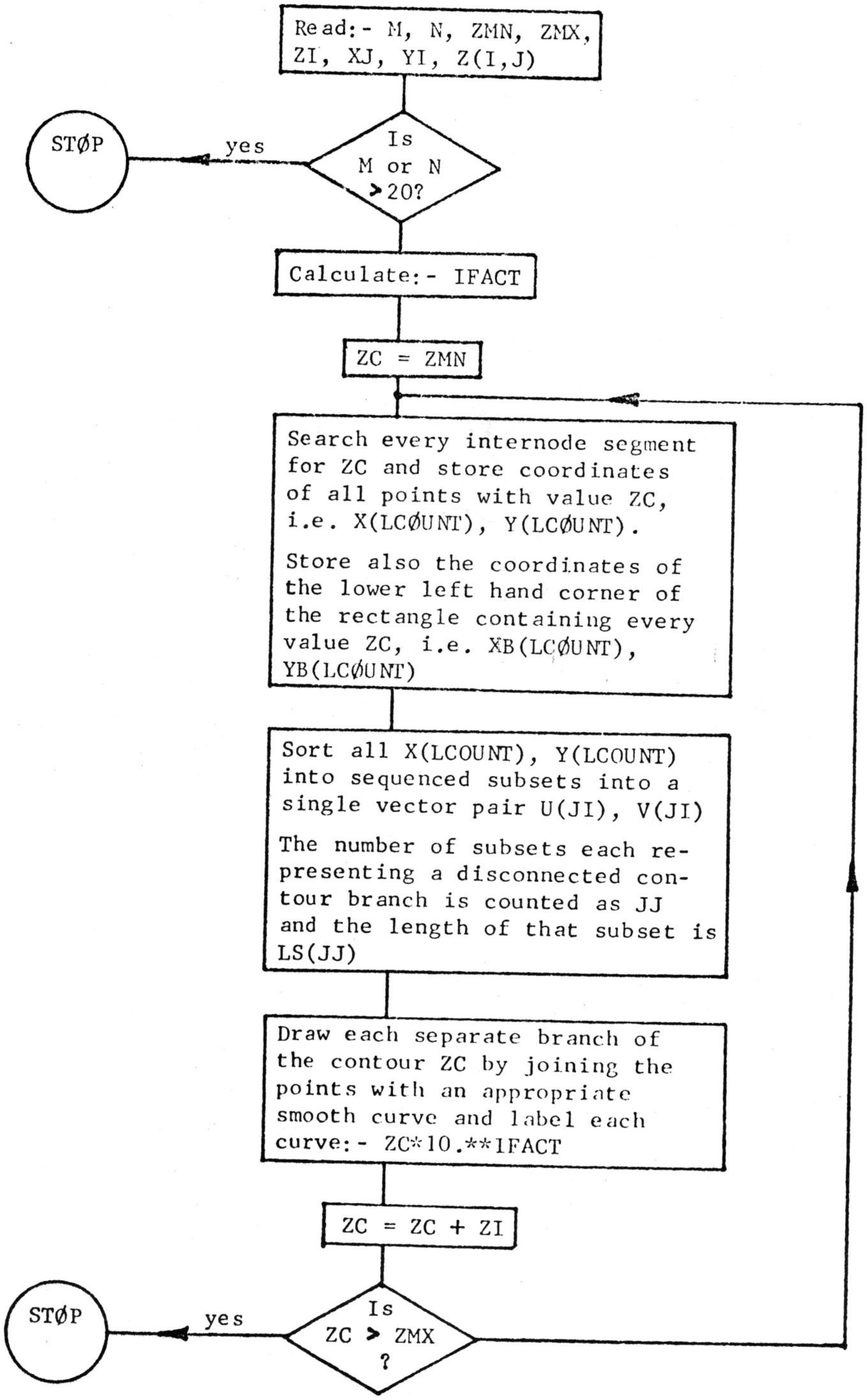
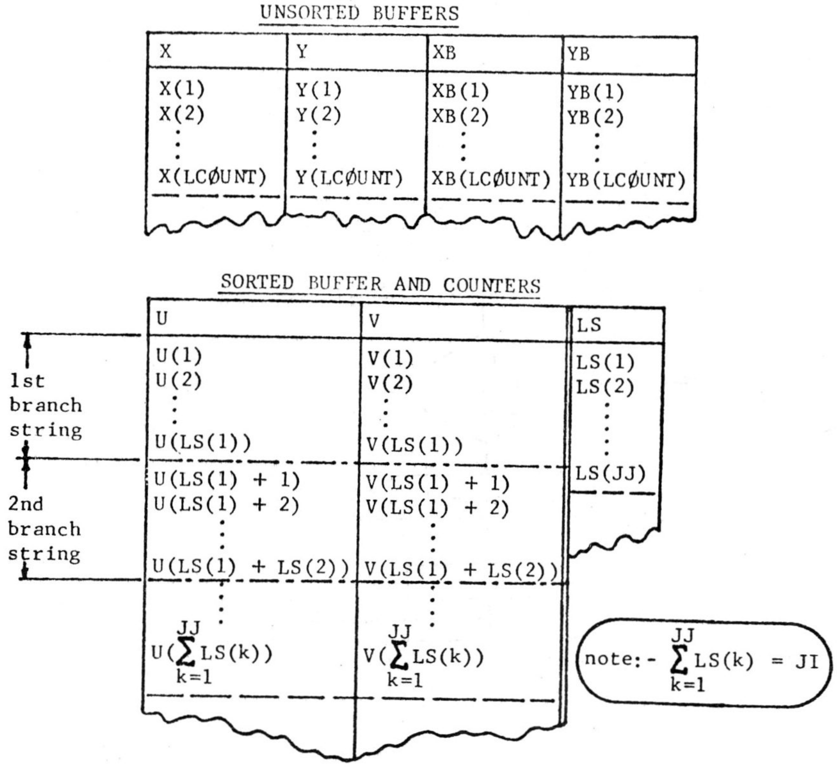


FIG. 4

FIG. 2


Read:- M, N, ZMN, ZMX, ZI, XJ, YI, Z(I,J)
STØP
yes
Is M or N >20?
Calculate:- IFACT
ZC = ZMN
Search every internode segment for ZC and store coordinates of all points with value ZC, i.e. X(LCØUNT), Y(LCØUNT).
Store also the coordinates of the lower left hand corner of the rectangle containing every value ZC, i.e. XB(LCØUNT), YB(LCØUNT)
Sort all X(LCOUNT), Y(LCOUNT) into sequenced subsets into a single vector pair U(JI), V(JI)
The number of subsets each representing a disconnected contour branch is counted as JJ and the length of that subset is LS(JJ)
Draw each separate branch of the contour ZC by joining the points with an appropriate smooth curve and label each curve:- ZC*10.**IFACT
ZC = ZC + ZI
STØP
yes
Is ZC > ZMX ?



UNSORTED BUFFERS

X	Y	XB	YB
X(1)	Y(1)	XB(1)	YB(1)
X(2)	Y(2)	XB(2)	YB(2)
⋮	⋮	⋮	⋮
X(LCØUNT)	Y(LCØUNT)	XB(LCØUNT)	YB(LCØUNT)

SORTED BUFFER AND COUNTERS

	U	V	LS
1st branch string	U(1)	V(1)	LS(1)
	U(2)	V(2)	LS(2)
	⋮	⋮	⋮
	U(LS(1))	V(LS(1))	
2nd branch string	U(LS(1) + 1)	V(LS(1) + 1)	LS(JJ)
	U(LS(1) + 2)	V(LS(1) + 2)	
	⋮	⋮	
	U(LS(1) + LS(2))	V(LS(1) + LS(2))	
	⋮	⋮	
	U($\sum_{k=1}^{JJ}$LS(k))	V($\sum_{k=1}^{JJ}$LS(k))	

note:- $\sum_{k=1}^{JJ}$ LS(k) = JI

FIG. 5

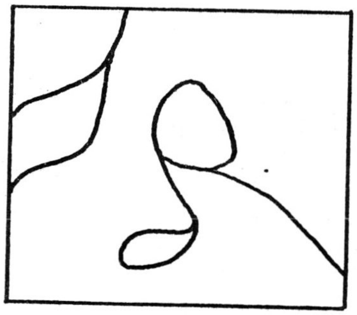


FIG.7a

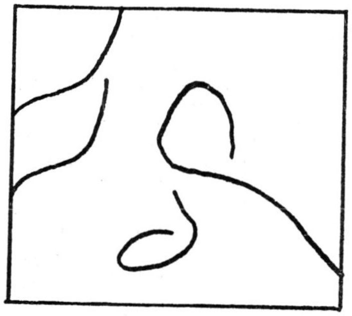


FIG. 7b

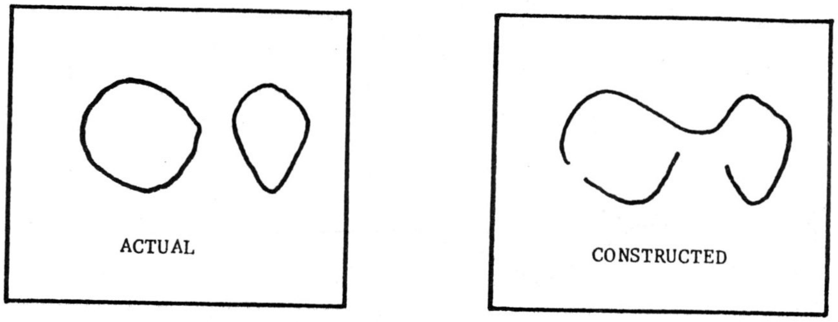


FIG. 8

HORIZONTAL SEGMENT SEARCH

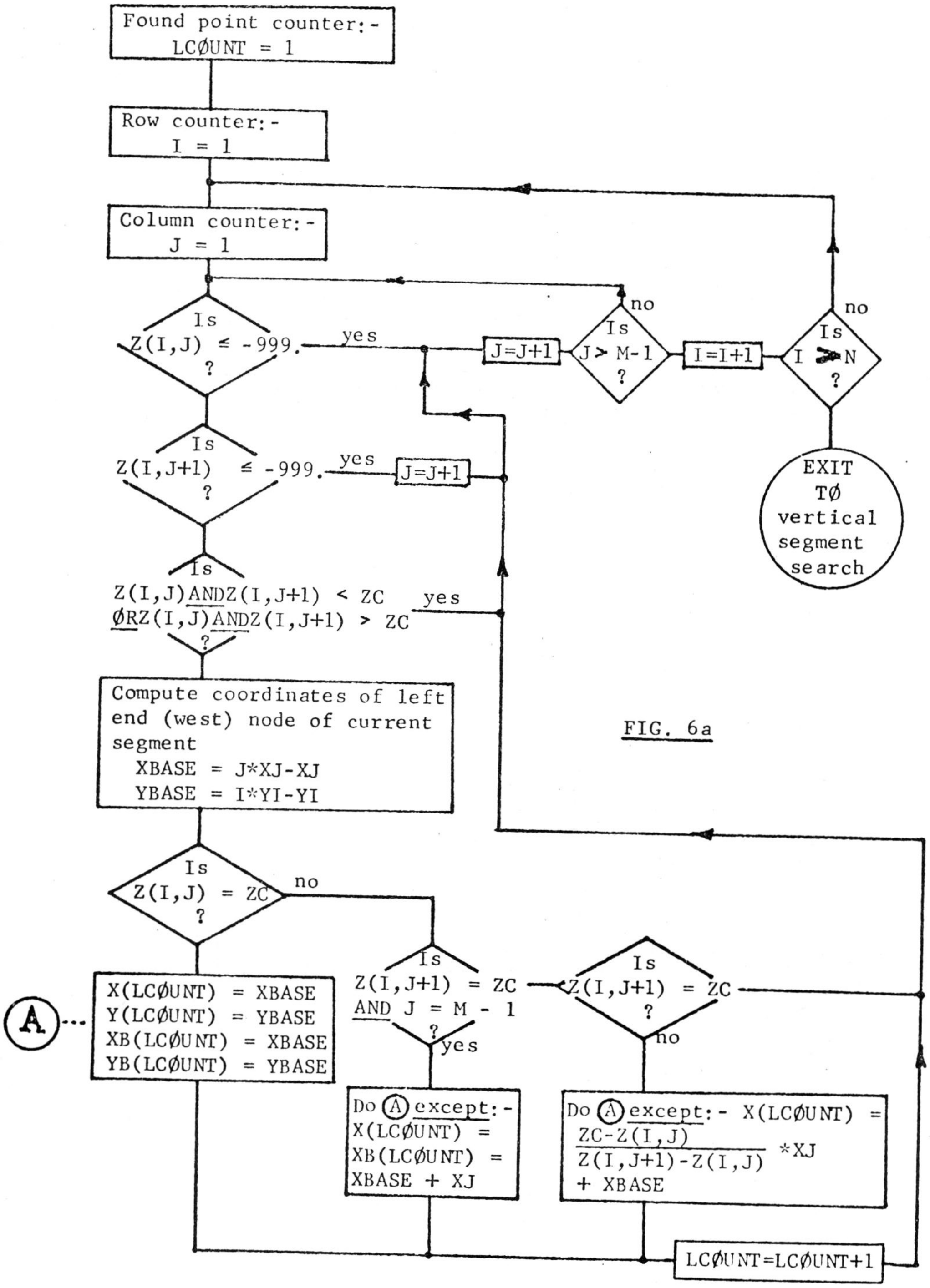


FIG. 6a

VERTICAL SEGMENT SEARCH

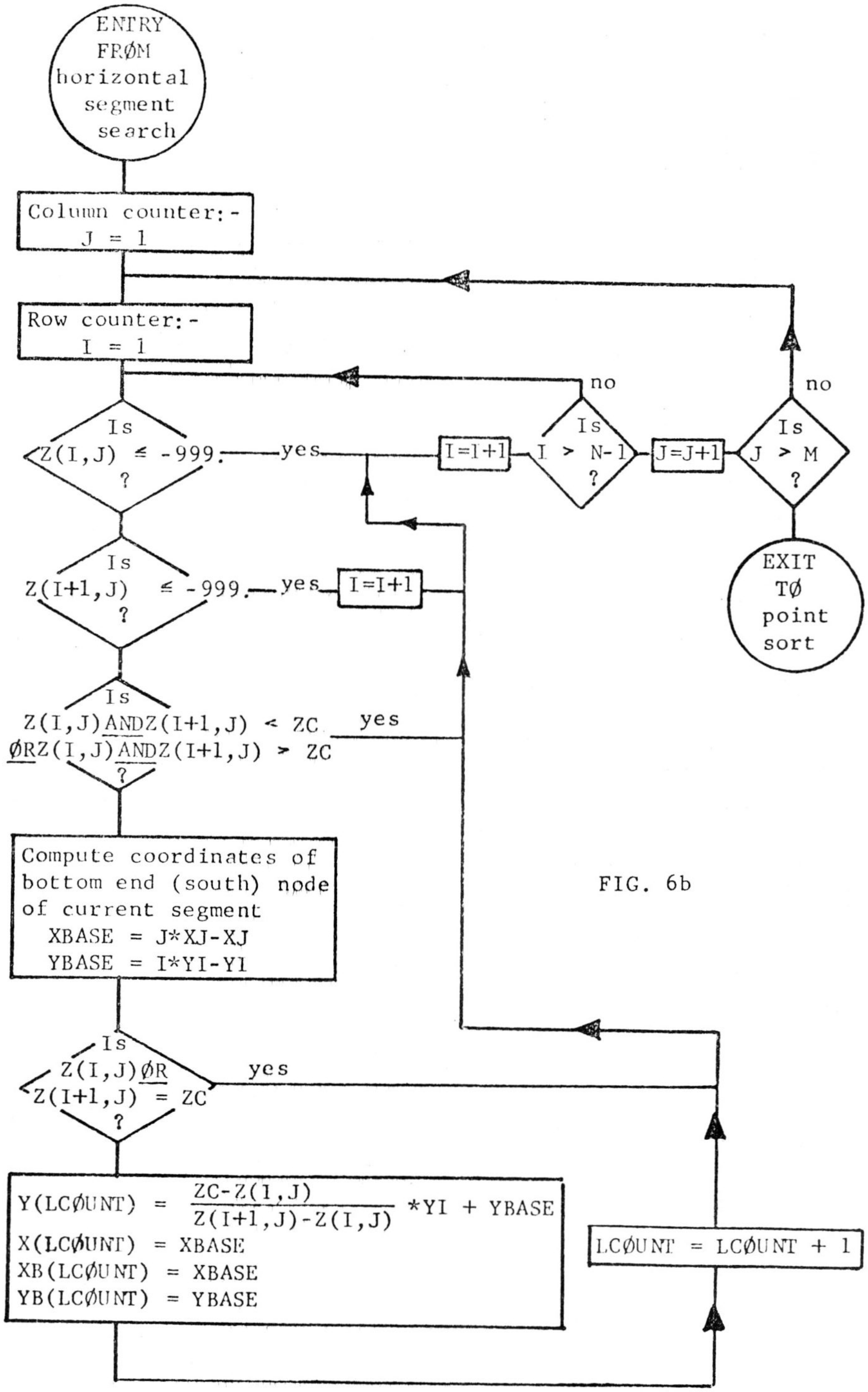


FIG. 6b

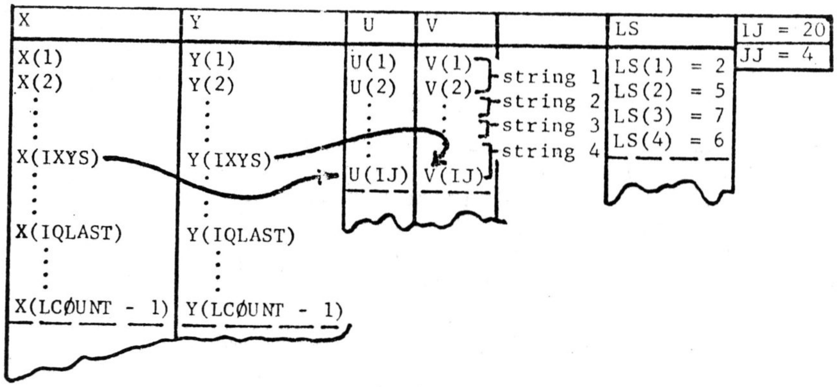
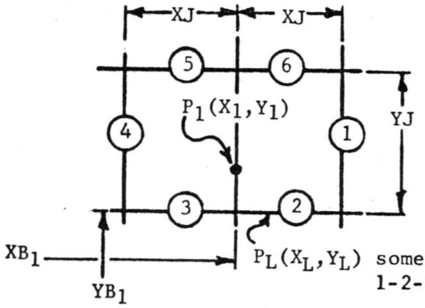


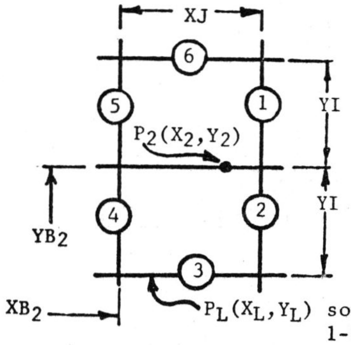
FIG. 9



0 - 6 points X_L
could exist

FIG. 10a

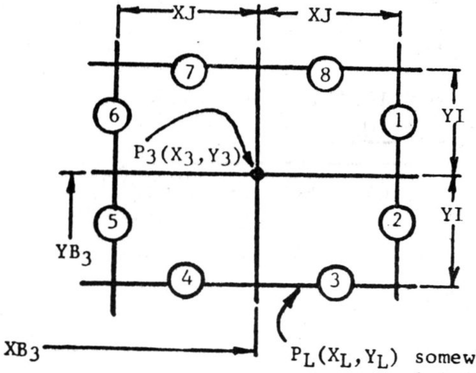
$P_L(X_L,Y_L)$ somewhere on perimeter
1-2-3-4-5-6 $\left\{\begin{array}{r} XB_1 - XJ \le X_L \le XB_1 + XJ \\ YB_1 \le Y_L \le YB_1 + YI \end{array}\right.$



0 - 6 points X_L
could exist

FIG. 10b

$P_L(X_L,Y_L)$ somewhere on perimeter
1-2-3-4-5-6 $\left\{\begin{array}{r} XB_2 \le X_L \le XB_2 + XJ \\ YB_2 - YI \le Y_L \le YB_2 + YI \end{array}\right.$



0 - 8 points X_L
could exist

FIG. 10c

$P_L(X_L,Y_L)$ somewhere on perimeter
1-2-3-4-5-6-7-8 $\left\{\begin{array}{r} XB_3 - XJ \le X_L \le XB_3 + XJ \\ YB_3 - YI \le Y_L \le YB_3 + YI \end{array}\right.$

POINT SORT

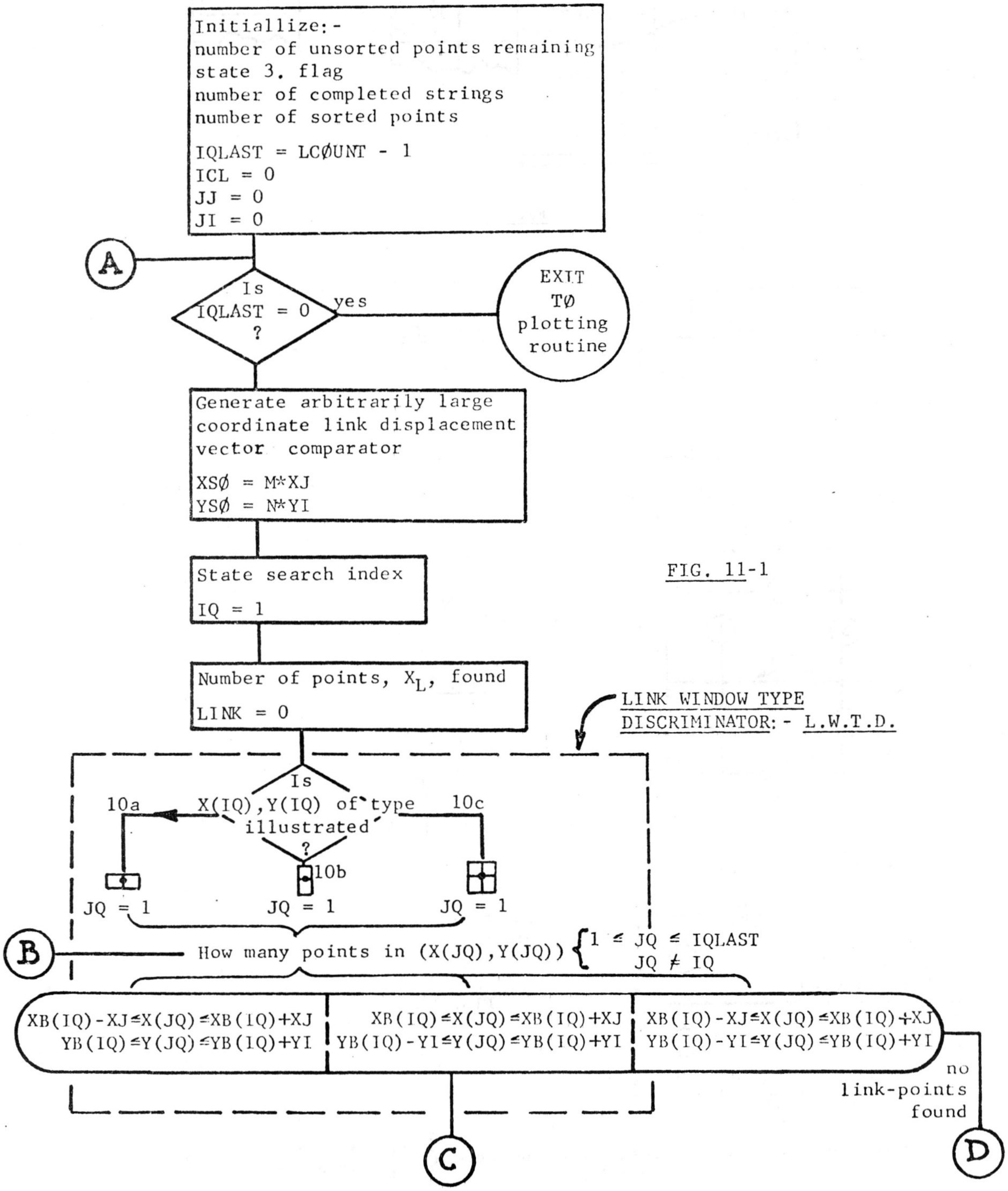


FIG. 11-1

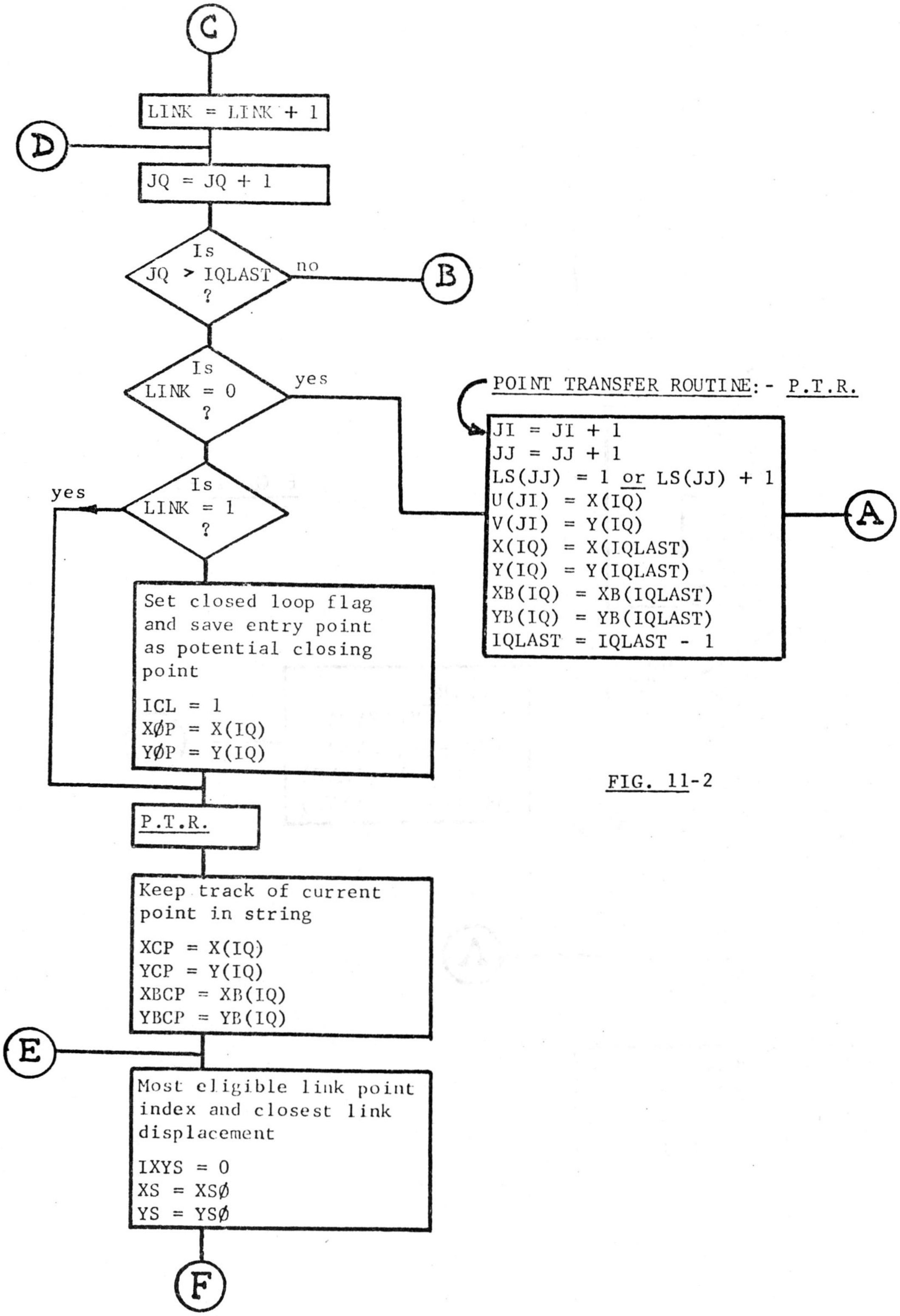

C
LINK = LINK + 1
D
JQ = JQ + 1
Is JQ > IQLAST ?
no
B
Is LINK = 0 ?
yes
Is LINK = 1 ?
yes
Set closed loop flag and save entry point as potential closing point
ICL = 1
XØP = X(IQ)
YØP = Y(IQ)
P.T.R.
Keep track of current point in string
XCP = X(IQ)
YCP = Y(IQ)
XBCP = XB(IQ)
YBCP = YB(IQ)
E
Most eligible link point index and closest link displacement
IXYS = 0
XS = XSØ
YS = YSØ
F
POINT TRANSFER ROUTINE:- P.T.R.
JI = JI + 1
JJ = JJ + 1
LS(JJ) = 1 or LS(JJ) + 1
U(JI) = X(IQ)
V(JI) = Y(IQ)
X(IQ) = X(IQLAST)
Y(IQ) = Y(IQLAST)
XB(IQ) = XB(IQLAST)
YB(IQ) = YB(IQLAST)
IQLAST = IQLAST - 1
A

FIG. 11-2

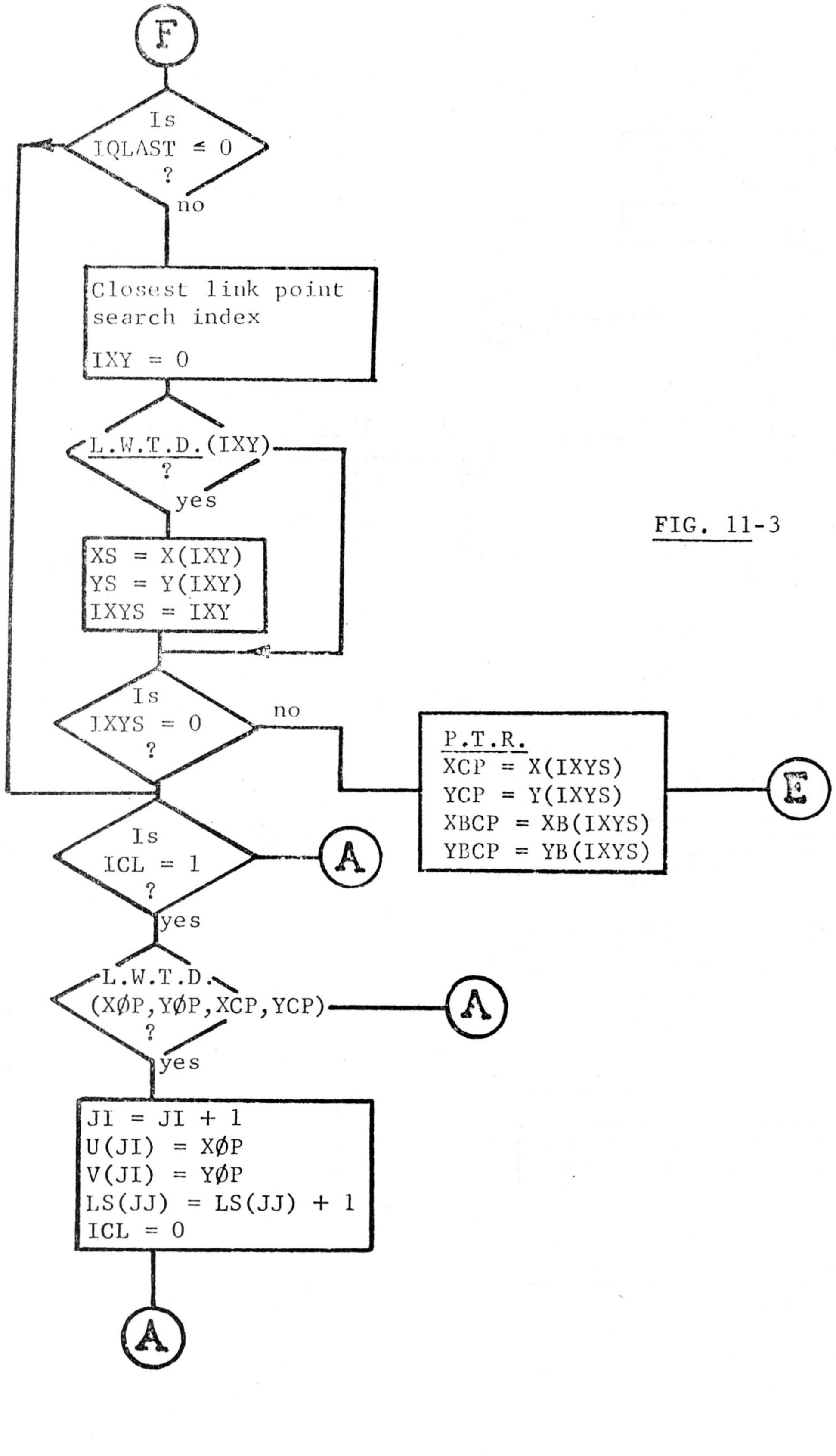

F
Is
IQLAST ≤ 0
?
no
Closest link point
search index
IXY = 0
L.W.T.D.(IXY)
?
yes
XS = X(IXY)
YS = Y(IXY)
IXYS = IXY
Is
IXYS = 0
?
no
P.T.R.
XCP = X(IXYS)
YCP = Y(IXYS)
XBCP = XB(IXYS)
YBCP = YB(IXYS)
E
Is
ICL = 1
?
A
yes
L.W.T.D.
(XØP,YØP,XCP,YCP)
?
A
yes
JI = JI + 1
U(JI) = XØP
V(JI) = YØP
LS(JJ) = LS(JJ) + 1
ICL = 0
A

FIG. 11-3

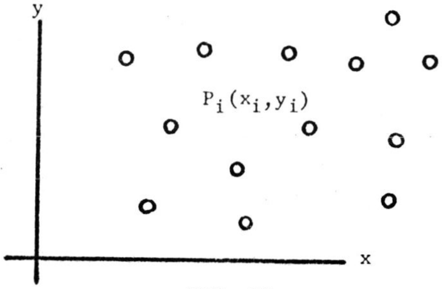


FIG. 12a

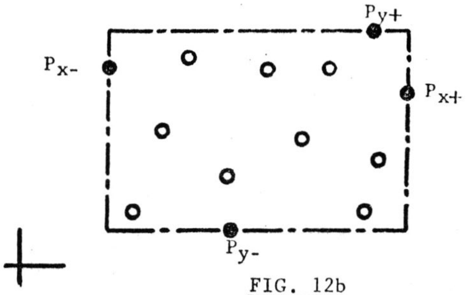


FIG. 12b

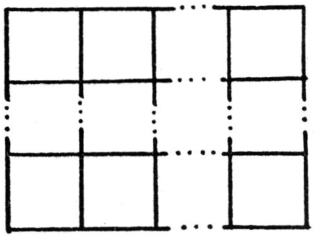


FIG. 13a

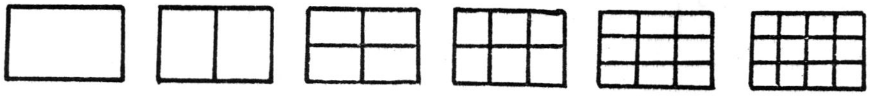


FIG. 12c

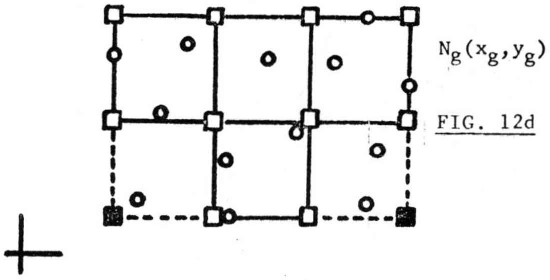


FIG. 12d

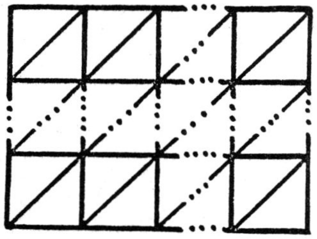


FIG. 13b

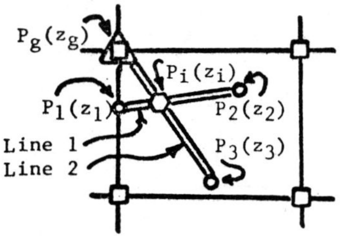


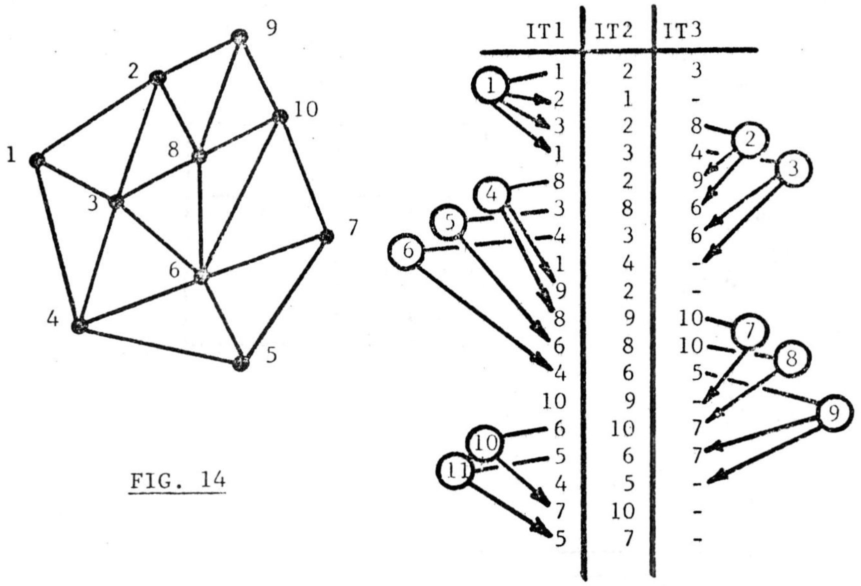
FIG. 12e

Line 1 $\quad y=a_1x+b_1,\ a_1=\frac{y_2-y_1}{x_2-x_1}$ and $b_1=y_1-\frac{(y_2-y_1)x_1}{x_2-x_1}=y_1-a_1x_1$

Line 2 $\quad y=a_2x+b_2,\ a_2=\frac{y_g-y_3}{x_g-x_3}$ and $b_2=y_3-a_2x_3$

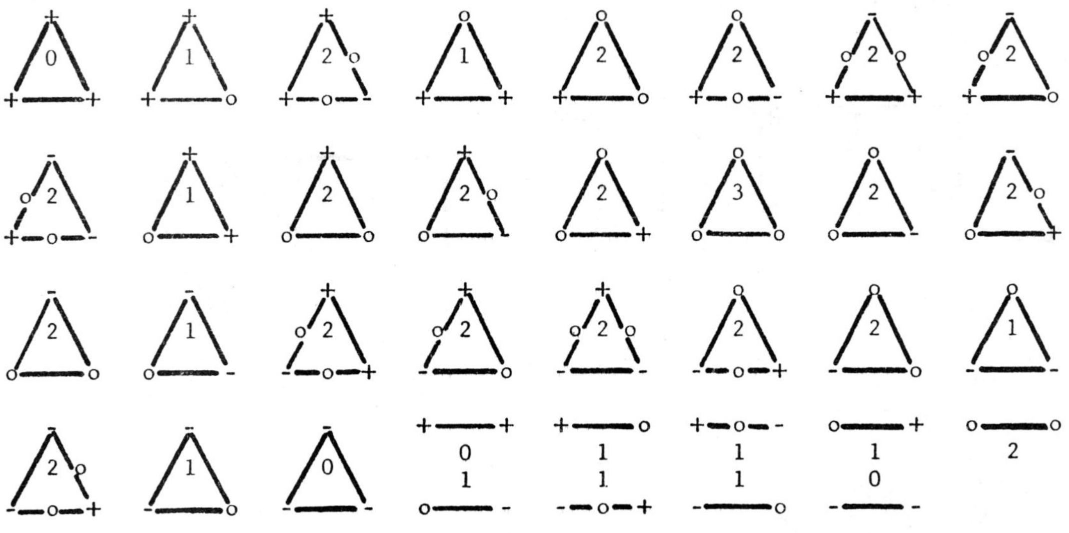
Intersection, 1 and 2 $\quad (a_2-a_1)x_i+(b_2-b_1)=0,\ x_i=\frac{b_1-b_2}{a_2-a_1}$ and $y_i=a_1\frac{(b_1-b_2)}{(a_2-a_1)}+b_1$

$$z_i=z_1+(z_2-z_1)\sqrt{\frac{(x_i-x_1)^2+(y_i-y_1)^2}{(x_2-x_1)^2+(y_2-y_1)^2}} \quad \text{and} \quad z_g=z_3+(z_i-z_3)\sqrt{\frac{(x_g-x_3)^2+(y_g-y_3)^2}{(x_i-x_3)^2+(y_2-y_1)^2}}$$



IT1	IT2	IT3
1	2	3
2	1	-
3	2	8
1	3	4
8	2	9
3	8	6
4	3	6
1	4	-
9	2	-
8	9	10
6	8	10
4	6	5
10	9	-
6	10	7
5	6	7
4	5	-
7	10	-
5	7	-

FIG. 14



o ... Indicates a point, on segment or node, at level ZC

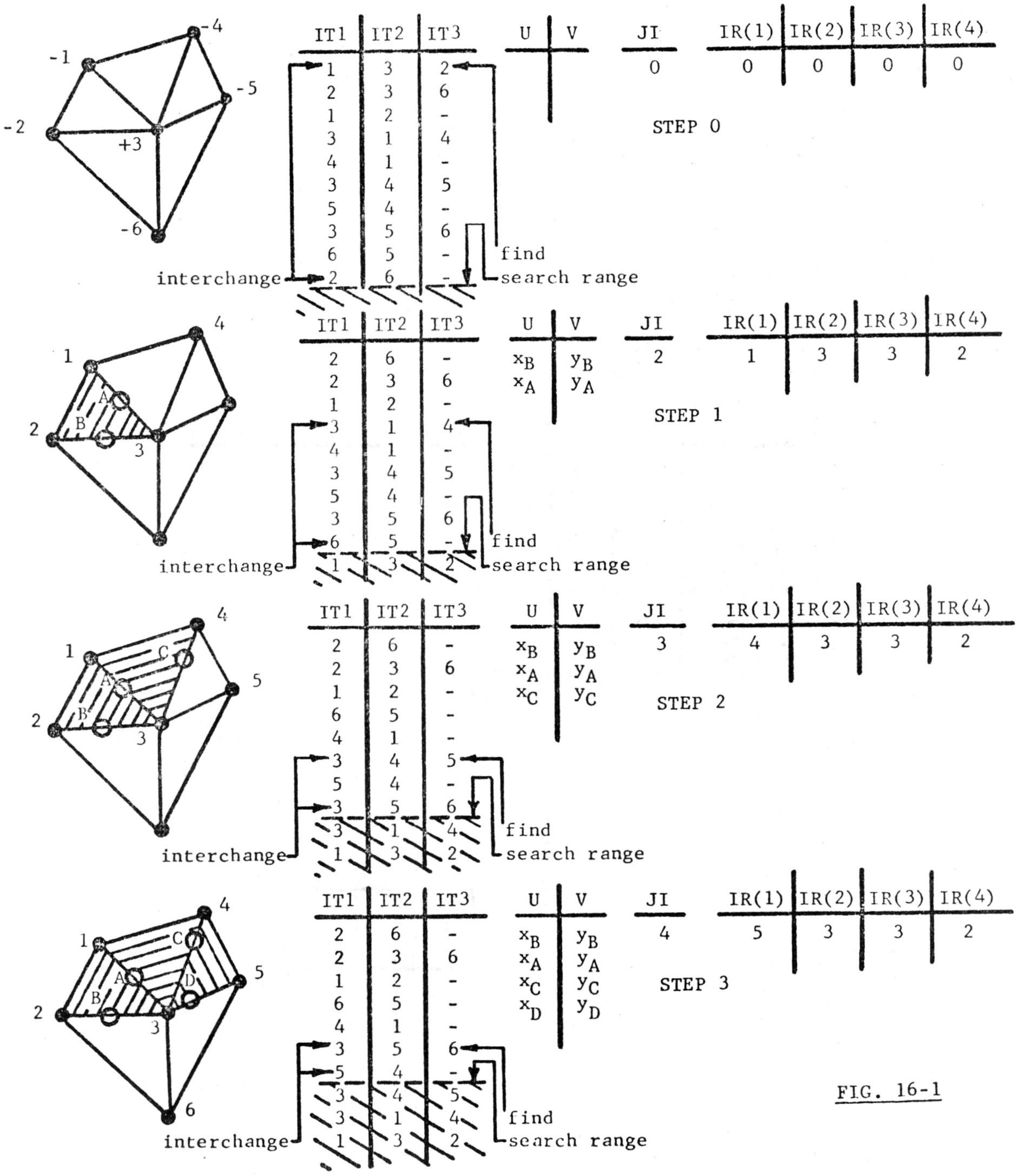
n ... Indicates number of points on cell boundary at level ZC

\+ ... Indicates a node, Z ZC

o ... Indicates a node, Z = ZC, if applicable

\- ... Indicates a node, Z ZC

FIG. 15



STEP 0

IT1	IT2	IT3
1	3	2
2	3	6
1	2	-
3	1	4
4	1	-
3	4	5
5	4	-
3	5	6
6	5	-
2	6	-

U	V

JI
0

IR(1)	IR(2)	IR(3)	IR(4)
0	0	0	0

STEP 1

IT1	IT2	IT3
2	6	-
2	3	6
1	2	-
3	1	4
4	1	-
3	4	5
5	4	-
3	5	6
6	5	-
1	3	2

U	V
x_B	y_B
x_A	y_A

JI
2

IR(1)	IR(2)	IR(3)	IR(4)
1	3	3	2

STEP 2

IT1	IT2	IT3
2	6	-
2	3	6
1	2	-
6	5	-
4	1	-
3	4	5
5	4	-
3	5	6
3	1	4
1	3	2

U	V
x_B	y_B
x_A	y_A
x_C	y_C

JI
3

IR(1)	IR(2)	IR(3)	IR(4)
4	3	3	2

STEP 3

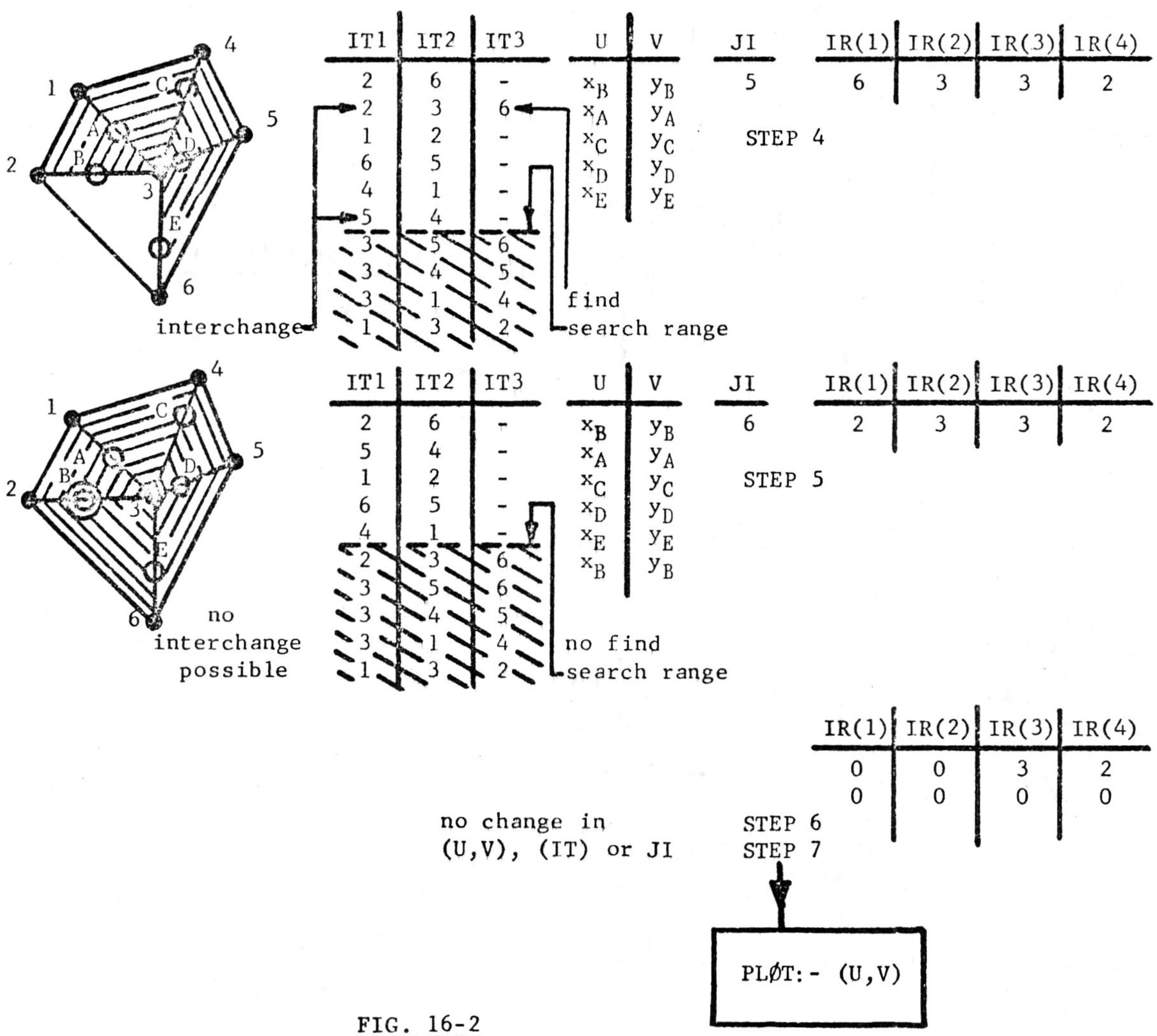
IT1	IT2	IT3
2	6	-
2	3	6
1	2	-
6	5	-
4	1	-
3	5	6
5	4	-
3	4	5
3	1	4
1	3	2

U	V
x_B	y_B
x_A	y_A
x_C	y_C
x_D	y_D

JI
4

IR(1)	IR(2)	IR(3)	IR(4)
5	3	3	2

FIG. 16-1



IT1	IT2	IT3
2	6	-
2	3	6
1	2	-
6	5	-
4	1	-
5	4	-
3	5	6
3	4	5
3	1	4
1	3	2

U	V
x_B	y_B
x_A	y_A
x_C	y_C
x_D	y_D
x_E	y_E

JI
5

STEP 4

IR(1)	IR(2)	IR(3)	IR(4)
6	3	3	2

IT1	IT2	IT3
2	6	-
5	4	-
1	2	-
6	5	-
4	1	-
2	3	6
3	5	6
3	4	5
3	1	4
1	3	2

U	V
x_B	y_B
x_A	y_A
x_C	y_C
x_D	y_D
x_E	y_E
x_B	y_B

JI
6

STEP 5

IR(1)	IR(2)	IR(3)	IR(4)
2	3	3	2

IR(1)	IR(2)	IR(3)	IR(4)
0	0	3	2
0	0	0	0

no change in
(U,V), (IT) or JI

STEP 6
STEP 7

PLØT:- (U,V)

FIG. 16-2

Contour	Step	Rejected Cell(s)			Points Added					
		IT1	IT2	IT3	(U,V)	JI	IR(1)	IR(2)	IR(3)	IR(4)
1	1	1	2	3	A				2	3
					B↓	2	3	1		
	2	1	3	4	C↓	3	3	4	2	3
	3	4	3	6	D↓	4	6	4	2	3
	4	4	6	5	E↓	5	5	4	2	3
	5	4	5	-	-	5	0	0	2	3
	6	3	2	8	F↑	6	0	0	0	8
	7	3	8	6			0	0		
		6	8	10						
		8	2	9 *	G↑	7			2	9
		8	9	10						
	8	9	2	-	-	7	0	0	0	0
					JI = 7, PLØT A-B-C-D-E-F-G**					
2	1	2	1	-	-	0	0	0	0	0
					JI = 0, NØ PLØT					
3	1	1	4	-	-	0	0	0	0	0
					JI = 0, NØ PLØT					
4	1	10	9	-	-	0	0	0	0	0
					JI = 0, NØ PLØT					
5	1	6	10	7	H				10	7
					I↓	2	7	6		
	2	5	6	7	J↓	3	7	5	10	7
	3	5	7	-	-	3	0	0	10	7
	4	7	10	-	-	3	0	0	0	0
					JI = 3, PLØT H-I-J					
					(IT) EXHAUSTED, ZC = ZC + ZI and CØNTINUE					

* Indicates selected cell
**Actual plotting sequence is G-F-A-B-C-D-E

↓Indicates point entered at bottom of list (U,V), ↑ at top

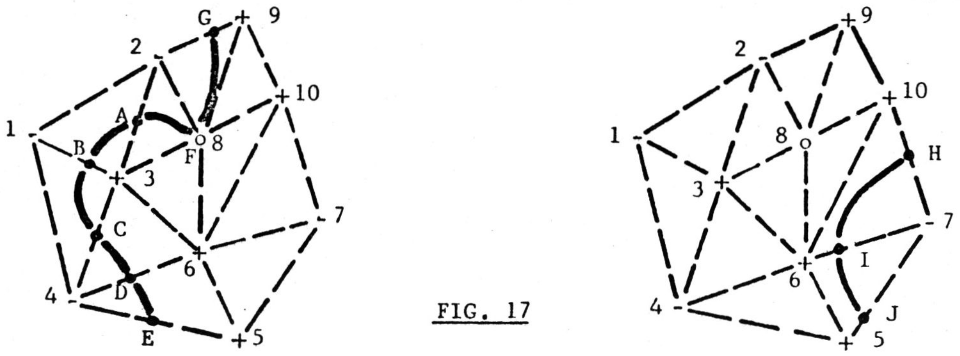


FIG. 17